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

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.*

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., [59] 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 concern 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 [47]. 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, 36], 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 when given 3 in-context examples. Our 4B model surpassses 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*.

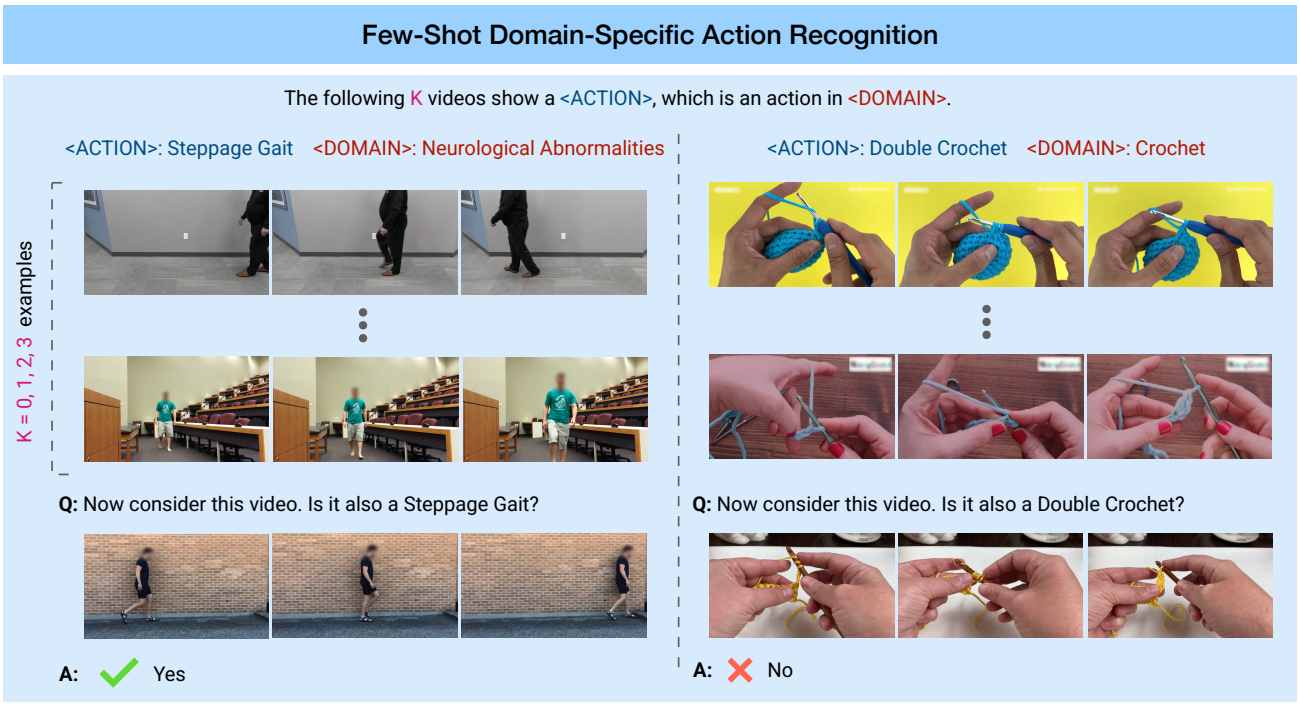


Figure 1. **Examples from the VideoNet benchmark.** Our benchmark includes over 1000 actions across 38 domains, ranging from Crochet to Neurological Abnormalities. It features 5 well-trimmed clips per action. (The prompts shown above are simplified for succinctness.)

- 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.[1]

2. Related Work

Action recognition has been has been extensively explored and fall broadly into three categories. The first set [10, 18, 22, 28, 37, 46, 49] predominantly contain *coarse-grained* labels (e.g., [18] has a single class for "rock climbing", whereas VideoNet contains 23 distinct bouldering actions.) Unsurprisingly, foundation models exceed at recognizing such coarse-grained labels. For instance, InternVideo2 [54] attains 92.1% on Kinetics-400 and 95.9% on ActivityNet. The second set [23, 29, 31, 32, 43, 45, 57, 58] focus on a limited set of sports, rendering them unable to test the generalization promise of foundation models. The third set [7, 8, 26, 47] 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 [17], covers only 8 domains, compared to VideoNet's 38. Our benchmark rivals the size of [17]'s entire dataset, while our training data contains 30x more videos. Perhaps most critically, [17] 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 [21], collects fine-grained actions for videos. However, it is restricted to egocentric videos. The third, ActionAtlas [45], 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.

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), re-

[1] 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.

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

quire 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 [45], 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 [45] to generate definitions. However, specialized domains pose challenges, as LLMs occasionally encode incorrect or outdated domain knowledge [51]. 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, 25, 30]. 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 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

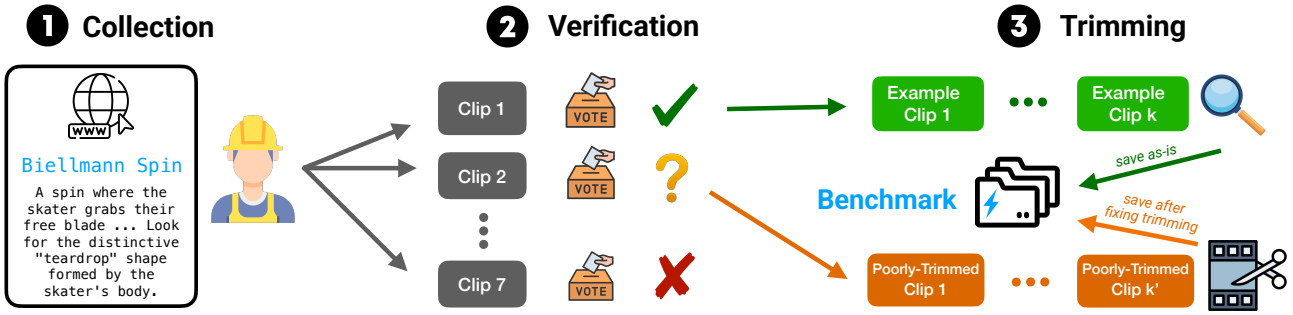


Figure 3. **Benchmark data collection pipeline**, as described in Section 3.2. Given an action name and a definition, humans (1) find clips on the web, (2) remove outliers among these clips, and (3) fix the clip trimmings. This pipeline yields five well-trimmed clips per action.

annotators are *not* experts in these domains. We solve this dilemma by reducing the problem from k-way classification to 3-way classification. Where [58] and [32] 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).

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

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 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, 45]. 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 [44, 60]. When possible, we find experts in our local communities and ask them to verify the data labels, akin to [21, 47, 55]. 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 [44]. As shown in Table 2, we see 97% accuracy in our data, exceeding MMLU-Pro's [55] expert accuracy of 85.4% and ImageNet's [44] 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 [15, 24]. 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) [16] 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.

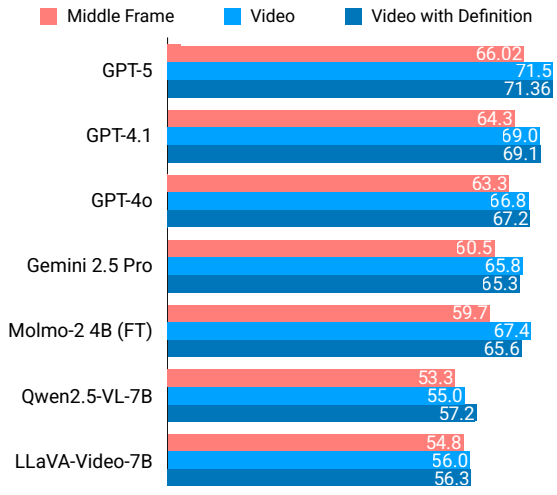
For open models, we use InternVL3-8B [64], Qwen2.5-VL-7B [2], and LLaVA-Video-7B [62]. For proprietary models, we use Gemini 2.5 Flash, Gemini 2.5 Pro, GPT-4o, GPT-4.1, and GPT-5.[2] A discussion of their context lengths–which may impact few-shot performance–is in the Appendix. Also in the Appendix are results for Qwen2.5-VL-72B, CLIP models [38, 52, 53, 61], 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 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.

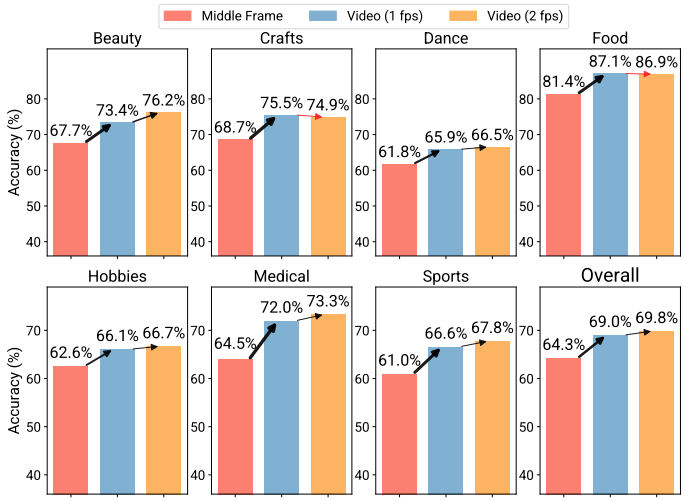
[2] Snapshots/versions of proprietary models available in Appendix.

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
Closed	Gemini 2.5 Flash	70.18	72.69	59.90	86.05	63.85	69.15	60.72	65.14
	Gemini 2.5 Pro	74.19	73.51	62.26	87.37	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	75.00	64.18	87.37	65.57	74.00	67.59	69.02
	GPT-5	**75.00**	**77.53**	**70.07**	**88.40**	**67.61**	**79.50**	**68.32**	**71.51**
Open Weight	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
Fully Open	Molmo2-4B (base)	50.00	51.20	50.00	72.16	55.58	58.00	53.26	55.01
	Molmo2-4B (FT)	**75.00**	69.66	66.84	76.03	66.51	71.00	66.33	67.36



(a) Accuracy of open and proprietary video models with different video input settings. Molmo-2 4B (FT) denotes our fine-tuned model using SINGLEACTION.



(b) GPT-4.1 accuracy across categories for single frame vs. video at varying frame rates. Black arrows are gains from previous setting with width proportional to relative improvement; red arrows are decreases in performance.

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.

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, 36]. 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 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

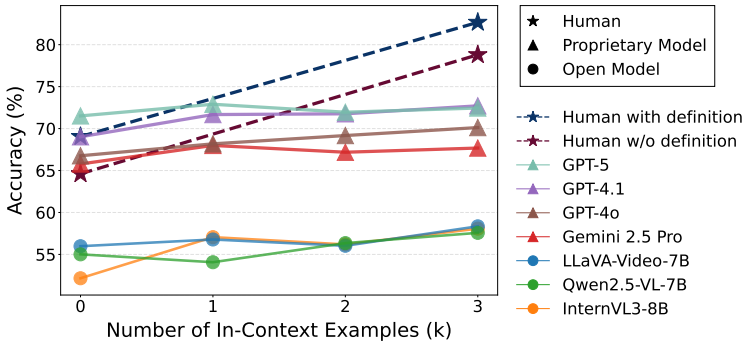


Figure 5. **Few-shot accuracy** of video models and humans with k in-context video demonstrations. Humans benefit significantly more than models do.

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

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-short learning capabilities.

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

Supplementary Material

This Appendix contains the following sections:

- § A - **Benchmark statistics**; discusses VideoNet's inter-domain breadth and intra-domain depth, the latter in comparison to existing works.
- § B - **Benchmark collection**; prints LLM prompts and UIs used during benchmark construction.
- § C - **Model evaluation**; details on how we evaluated existing models on the VideoNet benchmark (prompts, video sampling, model versions, etc.).
- § D - **Zero-shot ablations**; detailed results for the ablations shown in Figure 4.
- § E - **Few-shot results**; detailed results for models in the few-shot setting. Additional results for 72B models, CLIP models, and optical flow models. Discussion of prompt-sensitivity in Gemini and the impact of few-shot examples on yes/no bias.
- § F - **Benchmark qualitative analysis**; examination of the types of failures VLMs suffer on the domain-specific action recognition task.
- § G - **Human evaluation**; details on the human evaluation setup. In-depth human evaluation results.
- § H - **Additional training details**; construction of VQA pairs from labeled video clips. Listing of learning rates, image pooling, etc.
- § I - **Data filtering strategies**; description of and motivation behind filtering strategies. Analysis of differences in downstream performance on VideoNet benchmark when different filters are applied.

Table 6. **Depth of VideoNet.** The last two columns report, for a given domain, the # of actions in other benchmarks and the # of actions in VideoNet respectively. When compared to domain-specific benchmarks that focus on fewer domains, it is clear that VideoNet maintains sufficient depth in the domains it covers. (Many values in the second-to-last column sourced from Table 1 in [58].)

Domain	Paper Name	Paper Venue	Theirs	Ours
Figure Skating	MCFS [34]	AAAI 2021	130	40
	MMFS [35]	arXiv	46	
	FSBench [19]	CVPR 2025	20	
	Fis-V [56]	TCSVT 2020	13	
	FSD-10 [33]	Neurocomputing	10	
Basketball	FineSports [58]	CVPR 2024	52	46
	Basket [42]	CVPR 2025	20	
	Basketball [23]	ICASSP 2020	27	
	MultiSports [32]	ICCV 2021	18	
Soccer	MultiSports [32]	ICCV 2021	21	43
	SoccerNet [20]	CVPR 2018	3	

A. Benchmark Statistics

Given that previous domain-specific benchmarks (*e.g.* [23, 35, 48, 57, 58], see Section 2) have chosen to sacrifice breadth for depth, it is natural to ask whether VideoNet inevitably sacrifices depth for breadth. As shown in Table 6, VideoNet achieves greater depth in many of the domains it covers when compared to previous one-domain works.

For the VideoNet benchmark, we release 7,036 clips spanning 38 domains within 7 categories. Table 7 provides a breakdown of each domain's category, number of actions, number of clips, and the length of these clips.

Basic benchmark-wide statistics on video duration are provided in Table 1; in particular, the average clip is 12.8 seconds long and the typical clip is 5.0 seconds long. Here we emphasize the long-tail nature of video lengths in VideoNet. This is caused by a handful of domains having much lengthier clips than most. For instance, the median length of a knots clip and a suturing clip are 36 seconds and 32.5 seconds respectively (see Table 7). Concretely, the kurtosis of video durations in VideoNet is 40.26, indicating a heavy tail.[3] The long tail is made evident by Figure 6.

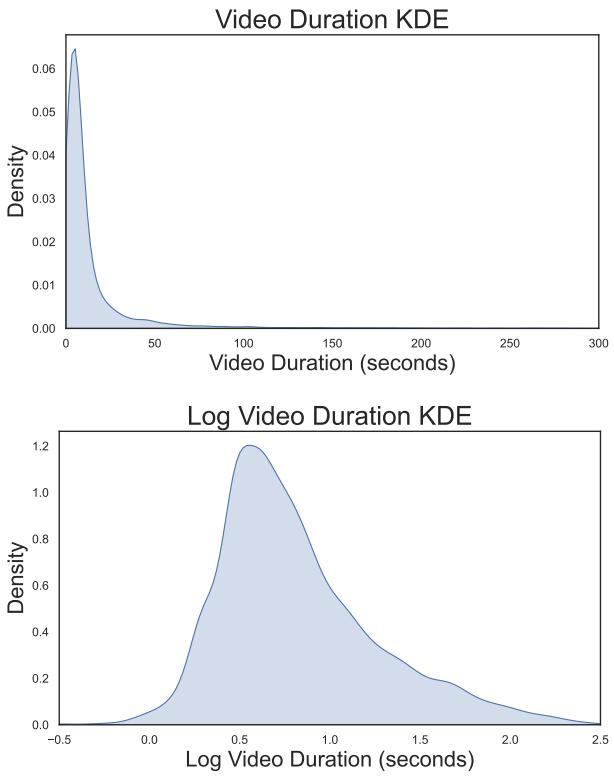


Figure 6. **Kernel Density Estimation for VideoNet Clip Durations.** The top graph clearly shows a long tail, but it is difficult to analyze due to the right tail's sheer length and the concentration of near-0 durations. For closer inspection, the bottom graph uses log duration. The smoothing bandwidth is determined by Scott's Rule.

[3]We report the Pearson kurtosis, not the Fisher/excess kurtosis. For reference, the Pearson kurtosis of the normal distribution is 3.

Table 7. **Actions & Clips per Domain.** We report the number of actions and number of clips in our benchmark for each domain below. Also reported are clip length statistics in seconds. While our evaluation setup only requires 5 clips for each action, we release between 5 and 7 clips for each action for future works to build on. Entries ordered alphabetically, left-to-right.

Category Name	**Domain Name**	**# Actions**	**# Clips**	**Clip Duration (s)**	
				Mean	**Median**
Beauty & Self Care	Hairstyling	14	91	12.34	4.7
	Spa Massage	11	72	21.40	11.0
	Tattooing	6	39	7.98	4.8
Crafts & Art	Calligraphy	8	53	5.86	4.6
	Crochet	38	252	40.23	22.0
	Hand Sewing / Embroidery	41	265	41.47	25.0
	Knots	55	379	44.56	36.0
	Painting	8	49	25.71	11.0
	Pottery	10	67	25.84	12.0
	Woodworking / Whittling	4	25	7.57	5.0
Dance	Ballet	39	248	5.28	4.0
	Bharatanatyam	24	161	13.95	9.0
	Break Dance	34	220	6.63	5.0
	Salsa	21	143	7.26	6.0
	Tap Dance	29	186	6.69	4.0
Food & Beverage	Bartending	30	197	9.50	5.0
	Coffee	16	100	12.98	9.0
	Cooking	51	331	17.89	10.0
Hobbies	Bouldering	23	146	7.58	6.0
	Gardening	20	121	21.24	10.7
	Gym	22	145	5.28	4.0
	Juggling	26	176	6.52	4.0
	Parkour	40	261	4.69	3.4
	Pen Spinning	33	220	4.04	3.0
	Skateboarding	49	324	4.70	4.0
	Yo-yo	55	361	8.10	6.0
Medical	Neurological Abnormalities	21	136	12.02	7.6
	Neurological Assessments	15	101	9.15	7.0
	Suturing	14	94	48.00	32.5
Sports	American Football	54	336	5.79	5.0
	Basketball	46	284	4.01	4.0
	Cheerleading	23	155	5.71	4.4
	Cricket	46	288	3.95	3.1
	Fencing	20	123	4.99	3.0
	Figure Skating	40	258	6.33	5.0
	Ice Hockey	39	244	4.35	4.0
	Soccer	43	268	3.95	3.3
	Tennis	19	117	4.40	3.0
All	*All*	1,087	7,037	12.80	5.0

B. Benchmark Collection

B.1. LLM Augmentation of Action Lists

After collecting initial action lists from expert online sources, we expand them with Claude as specified in Figure 7.

B.2. LLM Deduplication of Action Lists

We then de-duplicate the action lists. Note that the LLM's response is only taken as a suggestion – the authors manually review duplicate actions identified by the LLM to decide if they are true duplicates or not. To preserve the integrity of our negatives and improve the fine-grained nature of our benchmark, if the action list has a general action (e.g., dunk) and many varieties of that action (e.g., tomahwak dunk, windmill dunk, alley-oop dunk), we remove the former and keep the latter. Refer to Figure 8 for the prompt.

B.3. LLM Generation of Action Definitions with Web-Search

We walk through our action definition generation pipeline as discussed earlier in § 3.1.

Initially, our pilot annotation study revealed that annotators had trouble correctly identifying actions when provided only with action labels, mainly due to their lack of domain-specific knowledge; based on their feedback, they struggled to ground the performed action in video and distinguish the accurate actions from incorrect ones. This initial setup resulted in numerous inaccurately labeled video clips.

To address this knowledge gap, we provide explicit action definitions describing the visual characteristics of each action using layman's terms. We design these definitions to be a stand-alone resource, thereby removing the need for annotators to locate external references. We use an LLM, Claude-3.7, with web-search capabilities to generate accurate action definitions informed by expert online communities. For each domain, we provide all actions at once and ensure the definitions satisfy the following conditions: they avoid overlap and do not reference other actions' definitions; they clearly elaborate on basic, atomic actions to minimize jargon, particularly for actions involving combinations of simpler actions; and they mention key differences from similar actions in the same list to prevent confusion.

We observe that providing action definitions during the annotation stage significantly helps non-expert humans in understanding the action. These improvements are further supported by the human evaluation results presented in Figure 5. We provide our exact prompt in Figure 9.

B.4. LLM Generated Hard Negatives

Figures 10-14 present the prompts and LLM generation parameters used to create the hard negatives described in § 3.3. In the first stage, we use `gpt-4.5-preview` to create an initial balanced set of hard negative candidates (Figure 11). In later stages, we use `o3-2025-04-16` to iteratively refine the negatives by 1) correcting false negatives that may co-occur with the positive actions, 2) diversifying the selection patterns by incorporating negatives with varying types of visual similarity, and 3) ensuring each action appears as a hard negative with balanced frequency (Figures 12-14).

B.5. Human Annotator UIs

Figures 15, 16, and 17 contain the user interfaces shown to human annotators during the collection, verification, and trimming stages respectively. For full reproducibility, the HTML/CSS will be made available on our GitHub repository. Annotators were paid $15-$17 per hour for their efforts.

B.6. Sourcing Human Annotators

We begin with two pools of approximately 1000 and 50 human annotators. The annotators in these pools have done "good" and "exemplary" jobs, respectively, in previous Prolific studies hosted by the authors.[4]

(It may be helpful to review the annotation stages shown in Figure 3.) All annotators from the first pool are invited to complete Stage 1 (clip collection) on a small subset of domains (we later re-collected the data for this subset after we had filtered a set of "great" annotators). We then asked the second pool, in whom we had high confidence, to complete Stage 2 (clip verification). We kept the top one-fifth of annotators, as determined by the percentage of "yes" votes the clips they collected in Stage 1 received during the verification process in Stage 2. This newly-derived pool of approximately 200 annotators was used to collect clips for the VideoNet benchmark.

[4] Prolific is a crowd-sourcing platform.

```
I have the following list of <DOMAIN> actions:

<INITIAL ACTION LIST>

Provide me with suggestions of <DOMAIN> actions that are well-defined and
highly-discernible. Your suggestions should not overlap with each other, nor
should they overlap with any of the <DOMAIN> actions on the list I provided.
```

Figure 7. **Action list augmentation prompt.**

```
Are there any duplicates or near-duplicates in this list of <DOMAIN> actions?
```

Figure 8. **Action deduplication prompt.**

```
Generate detailed definitions for the following <DOMAIN> actions from
<CATEGORY> category. Each definition should:

Be completely self-contained and understandable without referencing other
actions. Explain any specialized terminology within the definition (using
phrases like
"which is..." or "meaning...")
Include visual identification cues (what to look for to recognize the action)
Describe how this action differs from similar actions when applicable.
Be written for a general audience with no prior knowledge of the domain.

Format each definition as:
[ACTION NAME]: [Complete definition with all elements above]
Use web search to gather accurate information about these actions, but DO NOT
include source links or citations in your final output. The goal is to create
clean, comprehensive definitions that can be easily copied into a spreadsheet
or database.

Here are the actions to define:
<ACTION LIST>
Remember that each definition must stand alone since readers may only see
one definition at a time.
```

Figure 9. **Action definition prompt.**

System Prompt

```
You are creating challenging "hard negative" options for multimodal
action classification datasets across various domains
(sports, arts, crafts, cooking, etc.).

Each action requires 3 hard negative options that are genuinely difficult
for a machine learning model to distinguish from the positive action.

A truly "hard" negative:
- Shares visual/motion similarities with the positive action that would
  be difficult to distinguish in brief clips
- Is fundamentally different in purpose or technique despite visual
  similarities
- Cannot reasonably co-occur with the positive action in the same short
  video
- Avoids obvious selection patterns that would make classification too
  easy

Note:
  Negatives should only come from the action list provided (not
  definitions or other sources)
    - Check that EXACT positive and negative action names are used
      in the actions list when generating csv.
```

Figure 10. **System prompt for hard negative generation**

User Prompt

```
Below is my list of "<ACTION>" actions, along with their definitions:
<ACTION DEFINITION>
===
Your task is to create genuinely challenging "hard negative" options
for each action that would confuse a computer vision model. Format
your output as a clean CSV:

action,negative_1,negative_2,negative_3
(action_1),(hard negative_1),(hard negative_2),(hard negative_3)
...
CRITICAL REQUIREMENTS FOR TRULY HARD NEGATIVES:

1. MAXIMIZE VISUAL CONFUSION WITHOUT OBVIOUS PATTERNS:
   - Select actions that share visual features, body positions, or
     motion qualities with the positive action
   - Avoid predictable selection patterns (e.g., don't always choose
     the "next level up/down" or "same family" actions)
   - Mix selection criteria unpredictably to prevent the model from
     learning simple heuristics
2. STRATEGIC AMBIGUITY:
   - Include some negatives that differ in subtle ways (small
     variations in technique/position)
   - Include some negatives that differ in more significant ways but
     still maintain visual similarity
   - Vary the type of similarity (sometimes motion-based, sometimes
     position-based, sometimes tool/environment-based)
3. AVOID FUNCTIONALLY RELATED ACTIONS FOR NEGATIVES:
   - Never select actions that typically occur together with the
     positive action
   - Avoid actions that are commonly performed in sequence or as
     part of the same technique
   - Don't pair actions that would naturally appear in the same
     short video clip
   - Don't pair action categories that are too similar or the same
     as the positive action
4. REASONABLE DISTRIBUTION:
   - Each action should appear as a negative approximately 2-5 times
     across the dataset
   - Avoid extreme over-representation or under-representation
   - The overall pattern of selections should appear random and
     unpredictable

Please provide your hard negative choices for these actions in the
same order as provided:
<ACTION LIST>
Negatives should only EXACTLY come from the action list provided
(not definitions or made-up sources)
- Check that EXACT positive and negative action names are used in
  the actions list when generating csv.
```

API Details

```
model: gpt-4.5-preview-2025-02-27
temperature: 0.5
max_tokens: 4096
```

Figure 11. **First user prompt for hard-negative generation** (1/4)

User Prompt

```
Please provide your analysis of negative selections for
their effectiveness as genuinely "hard" negatives:

First, check for selection patterns that could make classification too
easy:
- Are there predictable patterns in how negatives were selected?
- Is there too much consistency in how negatives relate to positives?
- Would these patterns potentially provide shortcuts for a
classification model?

Second, examine the visual confusion potential:
- How visually similar are the negatives to their positive actions?
- Is there sufficient variety in the types of visual similarity?
- Are the differences appropriately subtle to create genuine challenges?

Third, check for functional relationships:
- Are there any positive-negative pairs that typically occur together?
- Are there pairs that represent sequential or component actions?
- Would any pairs likely appear together in a short video clip?

Finally, review the overall distribution:
- Is any action severely over-represented or under-represented as a
  negative?
- Does the selection appear sufficiently unpredictable and varied?
- Are there imbalances that should be addressed?

For any issues identified, suggest specific improvements to create more
genuinely challenging hard negatives.
Provide a summary of the analysis and suggestions for improvement.
```

API Details

```
model: o3-2025-04-16
reasoning effort: high
```

Figure 12. **Second user prompt for hard-negative generation** (2/4)

User Prompt

```
Based on the analysis, provide a revised CSV with improved hard negatives.

Focus on fixing:
  1. The most problematic selection patterns identified
  2. Any actions with co-occurring negatives
  3. Distribution imbalances

Briefly explain the changes made to each action's negatives,
ensuring that the new selections are genuinely challenging
and visually confusing.
Then, provide the revised CSV with fixed negative selections,
without detailed explanations for each change.
```

API Details

```
model: o3-2025-04-16
reasoning effort: high
```

Figure 13. **Third user prompt for hard-negative generation** (3/4)

User Prompt

```
Based on the comprehensive analysis and specific suggestions,
synthesize a final CSV with truly challenging hard negatives
for each action.

Incorporate all the suggested improvements while ensuring:
  1. The final list follows the exact same order as the original
     action list
  2. Each action has 3 negatives that create genuine visual confusion
  3. The selection patterns remain unpredictable and varied
  4. No functionally related actions are paired
  5. The distribution is reasonably balanced
     (each action appears 2-5 times as a negative)

Provide the final clean CSV with optimized hard negatives:
```

API Details

```
model: gpt-4.5-preview-2025-02-27
temperature: 0.5
max_tokens: 4096
```

Figure 14. **Final user prompt for hard-negative generation** (4/4)

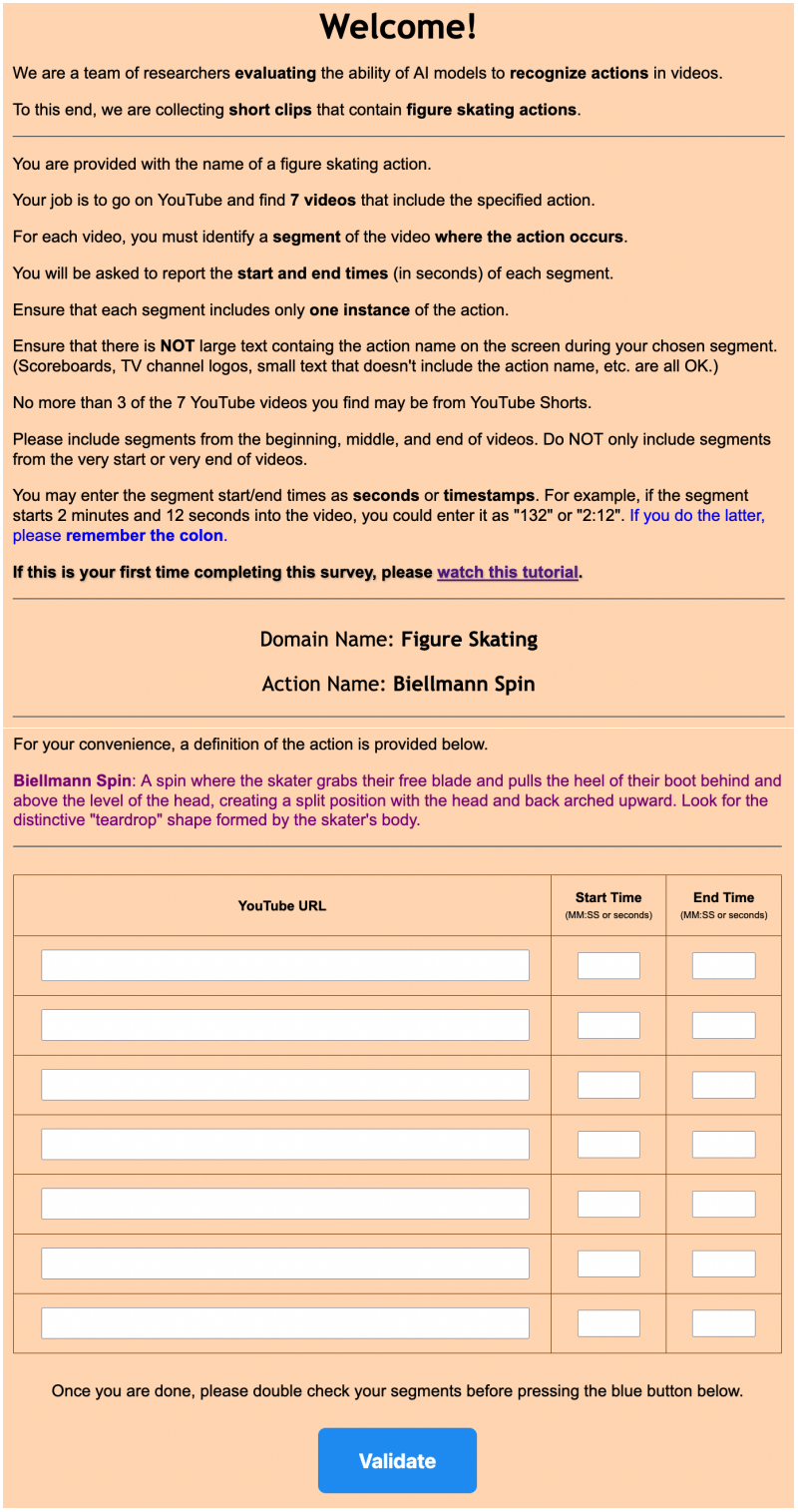


Figure 15. **Benchmark Clip Collection UI**. All of our UIs were refined based on annotator feedback. The annotators found this interface to be easy-to-use and appreciated the video tutorial. (Since the video tutorial was filmed by one of the authors, it will be linked in the final version of the paper once the double-blind process is over.)

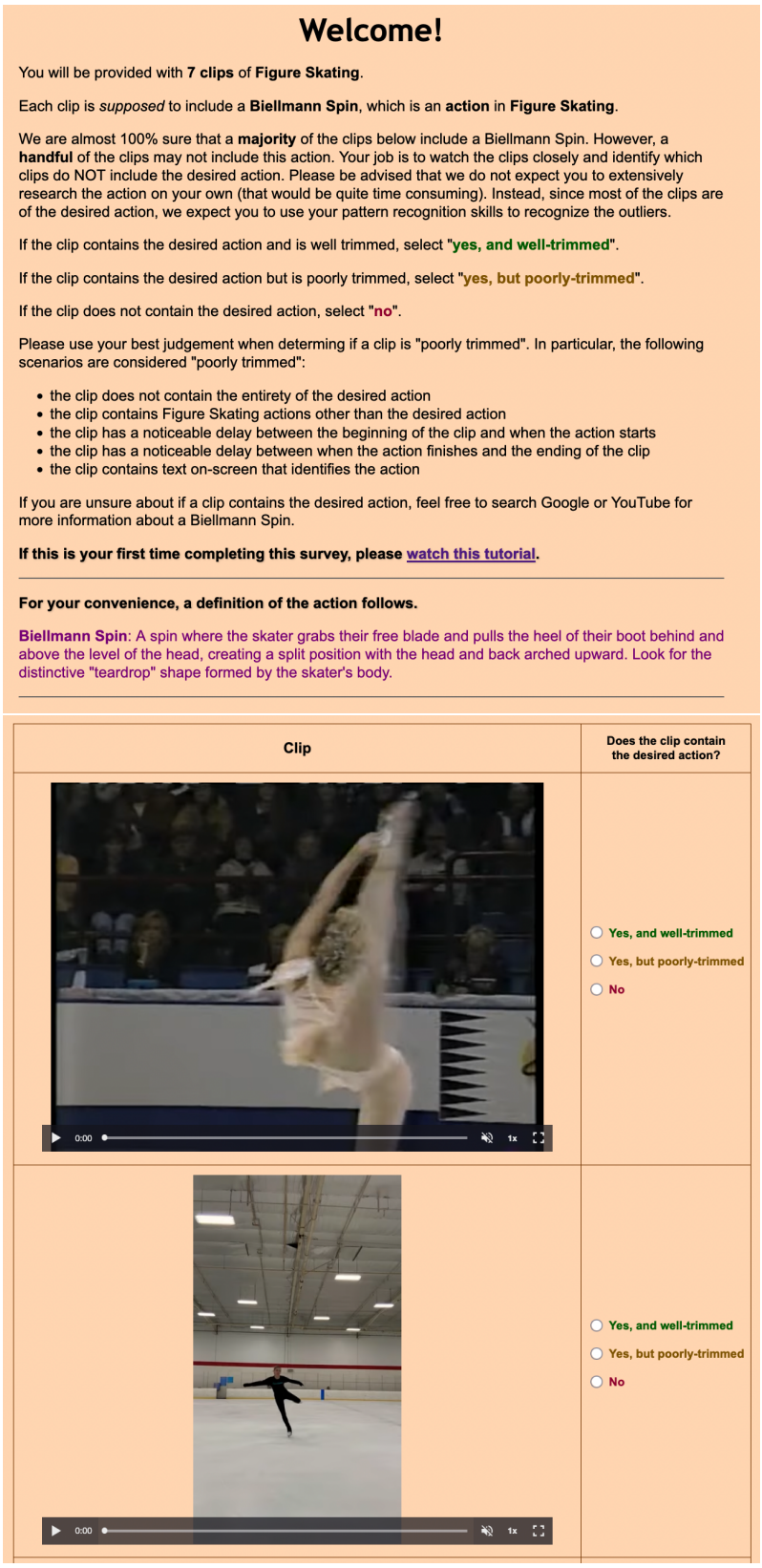


Figure 16. **Benchmark Clip Verification UI.** For brevity, only two of seven clips are displayed in the screenshot above. Likewise, a green submit button follows these clips, but is omitted above.

Welcome!

You are provided with **7** clips of Figure Skating.

Each clip includes a **Biellmann Spin**, which is an an **action** in **Figure Skating**.

You will help us ensure that these clips are well-trimmed.

Clips are either "well-trimmed" or "poorly-trimmed".

We say that a clip is well-trimmed if the clip contains the entirety of the action and not much else.

On the other hand, a clip is poorly-trimmed if **at least one** of the following conditions are met:

- it does not contain the entirety of the action,
- it contains Figure Skating actions other than the desired action,
- there is a noticeable delay between the beginning of the clip and when the action starts,
- there is a noticeable delay between when the action finishes and the ending of the clip.

Of the **7** total clips, another Prolific annotator determined that **5** of them were well-trimmed, while the remaining **2** clips were poorly-trimmed.

Your job is two-fold: first, you will watch the 5 well-trimmed clips to get a sense of what a Biellmann Spin in Figure Skating looks like; then, you will watch the 2 poorly-trimmed clips and adjust their trimmings so that they become well-trimmed.

Lastly, we want to ensure that none of the clips contain any text on-screen that writes out "Biellmann Spin". If such text is in the original poorly-trimmed clip, please try to trim it out. If your updated trimming still includes the text, please indicate so using the provided checkbox.

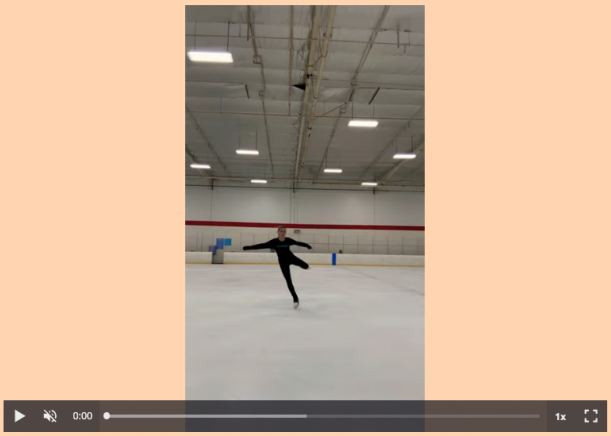
For your convenience, a definition of the action is provided below.

Biellmann Spin: A spin where the skater grabs their free blade and pulls the heel of their boot behind and above the level of the head, creating a split position with the head and back arched upward. Look for the distinctive "teardrop" shape formed by the skater's body.

The following videos are examples of **well-trimmed** clips of the **Biellmann Spin** action in **figure skating**.

Note that some examples may be **bad examples** (e.g., they may not contain the desired action).
This should be a rare occurence, but if it happens please select the "Bad Example" checkbox.

Also note that some examples may have **on-screen text containing "Biellmann Spin"**.
This should also be a rare occurence, but if it happens please select the relevant checkbox.



☐ Bad Example

☐ Onscreen Text has Action Name

☐ Bad Example

☐ Onscreen Text has Action Name

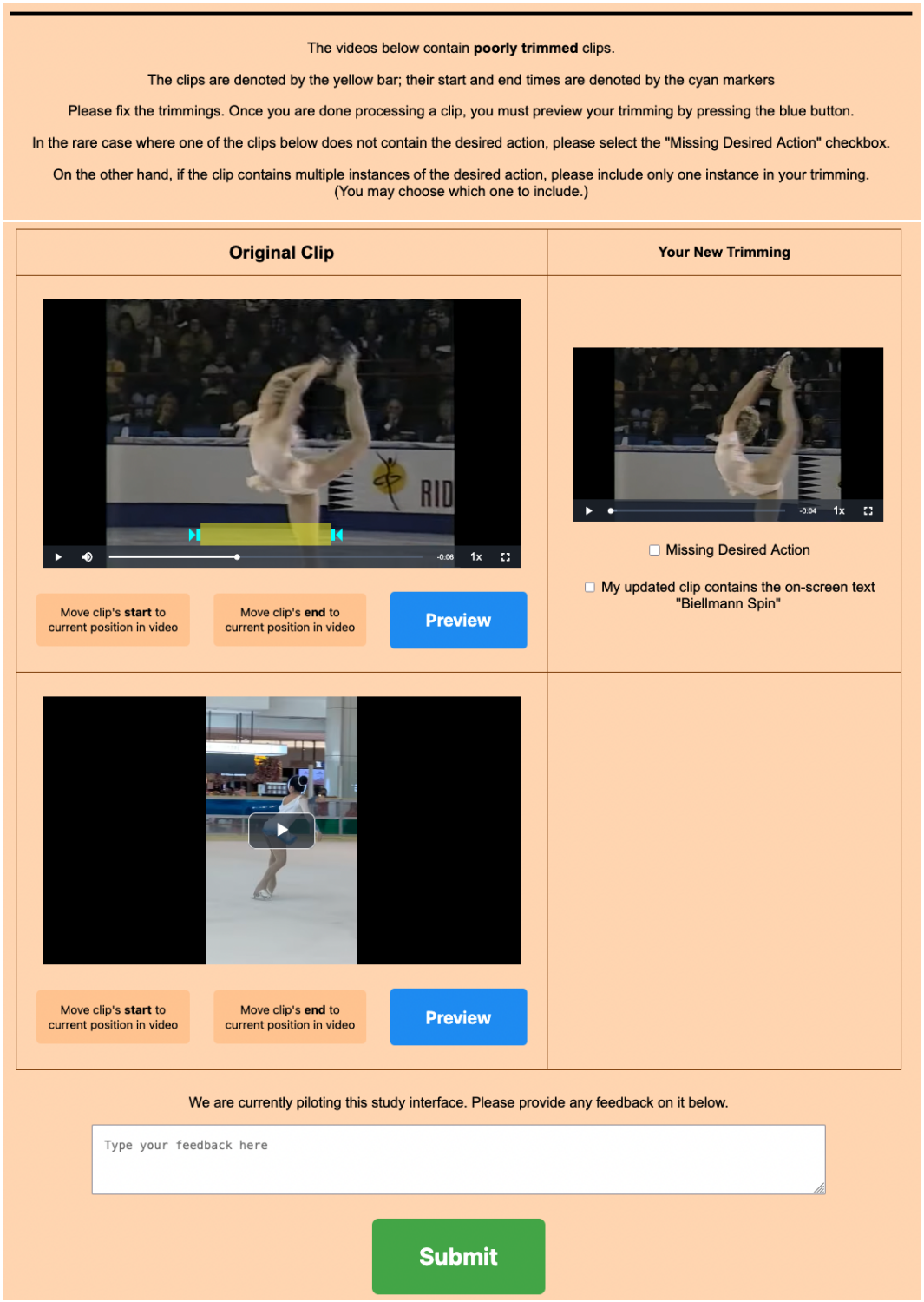


Figure 17. **Benchmark Clip Trimming UI.** The number of well-trimmed examples varies; for the action above, the true number is 5, but only 2 are shown for brevity. Similarly, the number of poorly-trimmed clips also varies.

C. Model Evaluation

Our entire evaluation code will be made available on our GitHub repository for reproducibility. In this section, we highlight some of the important decisions we make in our evaluation setup.

C.1. Evaluation Prompts

While we often tailor prompts to fit the expected input for each model (details on GitHub), they all closely resemble the following prompts.

—

0-shot Prompt

```
Recall that <a OR an> <ACTION> is <a
OR an> <SUBDOMAIN> in <DOMAIN>.  Does
the following video show <a OR an>
<ACTION>?  Please reason through your
answer.  It is critical that you output
'yes' or 'no' on the final line of your
answer.

<VIDEO>
```

—

3-shot Prompt

```
  The following 3 videos show <a
OR an> <ACTION>, which is <a OR an>
<SUBDOMAIN> in <DOMAIN>.

<VIDEO EXAMPLES>

Now consider the following video.  Is
it also <a OR an> <ACTION>?  Please
reason through your answer.  It is
critical that you output 'yes' or 'no'
on the final line of your answer.

<VIDEO>
```

—

The `<SUBDOMAIN>` field defaults to the string `"action"`, but we sometimes provide a more descriptive word in its place (e.g., some American Football actions are classified under the subdomain of `"run"`).

The `<a OR an>` field is either the string `"a"` or the string `"an"` depending on if the word it precedes begins with a vowel.

The 1-shot and 2-shot prompts are nearly identical to the 3-shot prompt above and can be found on our GitHub repository. They are omitted here for brevity.

C.2. Video Sampling

We generally use the video sampling techniques recommended by the authors of each model. In certain cases, we place an upper bound on frame sampling due to compute constraints.

- InternVL3-8B [64]: uniformly sample, max 64 frames.
- Qwen2.5-VL [2]: one frame per second (fps).
- LLaVA-Video-7B [62]: uniformly sample, max 110 frames.
- Molmo2-4B: four fps, max 64 frames.
- Gemini 2.5 Flash & Gemini 2.5 Pro [50]: one fps.
- GPT-4o, GPT-4.1, GPT-5 [39–41]: one fps, max 110 frames.

C.3. Context Lengths

For the open models, these numbers reflect a shared maximum on the number of tokens in both the input and output. For closed models, we have separate maximums for the input tokens and output tokens.

- InternVL3-8B: 8,192 tokens total
- Qwen2.5-VL: 128,000 tokens total
- LLaVA-Video-7B: 32,768 tokens total
- Molmo2-4B: 6,656 tokens total
- Gemini 2.5 Flash & Gemini 2.5 Pro: 1,048,576 input tokens; 65,536 output tokens
- GPT-4o: 128,000 input tokens; 16,384 output tokens
- GPT-4.1: 1,047,576 input tokens; 32,768 output tokens
- GPT-5: 400,000 input tokens; 128,000 output tokens

C.4. Proprietary Model Versions

We used the following versions of proprietary models.

- `gemini-2.5-flash-preview-04-17`
- `gemini-2.5-pro-preview-03-25`
- `gpt4o-2024-11-20`
- `gpt-4.1-2025-04-14`
- `gpt-5-2025-08-07`

D. Zero-shot Ablations

Tables 8 and 9 contains category-level results for GPT-4o and GPT-4.1 with 1 frame per second (fps) sampling and 2 fps sampling. We also provide results for GPT-4.1 in a 4 fps setting. (As noted in Appendix C, we feed no more than 110 frames to the GPT models.)

GPT-4.1 sees little difference in its performance upon varying the FPS, suggesting that a lack of frames is *not* the primary roadblock to achieving better performance on our benchmark. Additional analysis can be found in § 5.1. NB we chose GPT-4.1 over GPT-5 for the 4fps ablation due to its longer context length (1M vs. 400k) and better performance in the 1fps 3-shot setting (72.71% vs. 72.45%).

Table 10 contains category-level results for all models in the typical zero-shot setup of providing an input video, as well as two ablations: one where only the frame located at the (temporal) middle of the video is provided, and one where a definition of the action (as described in § 3.1) is provided alongside the video. In general, performance is best when a definition is provided alongside the video, and worst when only the middle frame is provided. Additional analysis can be found in Section 5.

Table 8. **Impact of higher FPS sampling in 0-shot.** Performance gain from the previous setup is highlighted in blue. If the model is sufficiently strong, like GPT-4.1, the additional frames do not seem to significantly boost performance on the domain-specific action recognition task.

Model	FPS	Beauty	Crafts	Dance	Food	Hobbies	Medical	Sports	Overall
GPT-4o	1	71.90	73.25	61.10	86.49	63.79	66.15	65.72	66.76
	2	76.92	73.87	63.80	84.49	64.68	67.60	67.64	68.21 (+1.45)
GPT-4.1	1	73.39	75.00	64.18	87.37	65.57	74.00	67.59	69.02
	2	76.23	74.89	66.54	86.91	66.69	73.30	67.78	69.85 (+0.83)
	4	75.70	75.14	66.53	87.23	66.11	73.78	69.19	69.93 (+0.08)

Table 9. **Impact of higher FPS sampling in 3-shot.** Performance gain from the previous setup is highlighted in blue and loss in red. The decrease in performance for GPT-4.1 suggests that the model struggles to handle the increase in visual tokens caused by the 3-shot, multi-fps setting. Interestingly, GPT-4o handles the additional tokens well, despite having a shorter context length (128k vs 1M).

Model	FPS	Beauty	Crafts	Dance	Food	Hobbies	Medical	Sports	Overall
GPT-4o	1	72.58	71.64	70.76	87.11	66.97	71.00	66.94	70.12
	2	9.11	70.79	71.34	86.75	68.76	75.00	67.75	70.99 (+0.87)
GPT-4.1	1	75.21	74.37	73.16	88.89	71.04	76.02	68.03	72.71
	2	75.29	72.62	76.09	89.59	68.72	72.73	67.49	71.22 (−1.49)
	4	80.56	85.00	74.25	86.73	70.51	69.49	68.58	71.15 (−0.07)

Table 10. **Zero-shot results while varying video inputs.** Performance gain from the previous setup is highlighted in blue and loss in red. The Molmo2-4B base model has a decrease in performance when shifting from providing the video's middle frame to providing the full-video input. The only other model where this occurs is InternVL3-8B, which is the worst-performing VLM we tested. This suggests that Molmo2 struggles to make effective use of video data. This further suggests that fine-tuning (with our data) an open-weight model which more effectively utilizes the full-video input, such as Qwen2.5-VL or LLaVA-Video, may lead to even better performance than our fine-tuned Molmo2 model.

Model	Input	Beauty	Crafts	Dance	Food	Hobbies	Medical	Sports	Overall
Gemini 2.5 Flash	Middle Frame	60.33	66.79	60.00	79.43	59.92	59.39	57.91	61.56
	Video	70.18	72.69	59.90	86.05	63.85	69.15	60.62	65.14 (+3.58)
	Video w/ Def.	**76.11**	69.23	63.22	86.56	64.10	70.74	59.49	65.56 (+0.42)
Gemini 2.5 Pro	Middle Frame	66.13	65.30	57.33	78.87	58.38	66.00	56.56	60.49
	Video	74.19	73.51	62.26	87.37	62.71	72.36	60.99	65.78 **(+5.29)**
	Video w/ Def.	70.16	70.90	64.18	86.60	63.40	70.85	58.71	65.29 (−0.49)
GPT-4o	Middle Frame	69.35	62.31	60.46	79.12	63.00	57.00	61.07	63.27
	Video	71.90	73.25	61.10	86.49	63.79	66.15	65.72	66.76 (+3.49)
	Video w/ Def.	70.73	73.60	64.86	86.01	63.93	70.05	66.86	67.19 (+0.43)
GPT-4.1	Middle Frame	67.74	68.66	61.78	81.44	62.61	64.00	61.01	64.27
	Video	73.39	75.00	64.18	87.37	65.57	74.00	67.59	69.02 (+4.75)
	Video w/ Def.	73.39	75.48	65.87	87.11	66.11	72.00	66.61	69.12 (+0.10)
GPT-5	Middle Frame	69.35	70.30	63.42	83.76	63.58	69.50	62.94	66.02
	Video	75.00	77.53	**70.07**	**88.40**	67.61	**79.50**	**68.32**	**71.51** (+5.49)
	Video w/ Def.	75.00	**79.39**	69.67	88.25	**68.83**	76.88	66.88	71.36 (−0.15)
InternVL3-8B	Middle Frame	54.84	47.39	52.76	66.75	53.36	54.50	55.37	54.74
	Video	54.03	51.12	54.69	64.18	47.25	54.00	51.63	52.16 (−2.58)
	Video w/ Def.	54.03	52.24	55.89	72.16	52.98	53.00	54.07	55.54 (+3.38)
Qwen2.5-VL-7B	Middle Frame	53.23	48.88	50.96	65.72	53.44	53.50	51.71	53.29
	Video	50.00	51.20	50.00	72.16	55.58	58.00	53.26	55.01 (+1.72)
	Video w/ Def.	62.10	57.46	53.85	73.20	56.19	54.50	55.54	57.24 (+2.23)
LLaVA-Video-7B	Middle Frame	57.26	50.37	52.28	66.24	54.28	55.00	54.32	54.85
	Video	58.87	57.84	51.32	70.36	54.74	58.00	54.89	55.98 (+1.13)
	Video w/ Def.	54.84	58.96	51.44	76.29	55.05	59.50	53.50	56.26 (+0.28)
Molmo2-4B (base)	Middle Frame	53.13	53.63	51.96	57.19	53.36	55.18	58.50	55.19
	Video	51.61	49.70	54.42	73.20	54.48	52.50	51.52	54.35 (−0.84)
	Video w/ Def.	61.29	54.57	51.19	77.58	57.28	57.00	51.21	56.12 (+1.77)
Molmo2-4B (FT)	Middle Frame	65.32	58.69	58.33	70.36	58.86	65.00	56.89	59.66
	Video	75.00	69.66	66.84	76.03	66.51	71.00	63.33	67.36 (+7.70)
	Video w/ Def.	69.35	68.90	62.24	79.38	63.81	72.00	61.67	65.64 (−1.72)

E. Few-shot Results

This section includes category-level results for VLMs, results for traditional computer vision models in a modified evaluation setting, and a discussion of prompt sensitivity & yes/no bias in Gemini 2.5 Pro.

E.1. Category-level Results for VLMs

Table 11 contains category-level results for all models from Figure 5 in the 0-shot, 1-shot, 2-shot, and 3-shot setups. Also provided are results for Qwen2.5-VL-72B, which surpasses all existing 7B models we evaluated in the 0-shot setting.

Table 11. **Few-shot results.** Performance gain from the previous setup is highlighted in blue and loss in red. The gains from few-shot examples are particularly remarkable for the 72B Qwen model and the 8B InternVL model.

Model	k-shot	Beauty	Crafts	Dance	Food	Hobbies	Medical	Sports	Overall
Gemini 2.5 Flash	0	70.18	72.69	59.90	86.05	63.85	69.15	60.72	65.14
	1	79.82	72.47	67.69	91.05	65.46	69.75	61.12	67.88 (+2.74)
	2	79.05	73.66	68.65	87.21	65.25	71.88	62.15	68.07 (+0.19)
	3	77.32	68.72	67.83	**91.86**	64.33	70.00	61.77	67.55 (−0.52)
Gemini 2.5 Pro	0	74.19	73.51	62.26	87.37	62.71	72.36	60.99	65.78
	1	72.58	74.63	67.79	88.40	65.08	68.84	62.70	67.99 (+2.21)
	2	71.67	75.00	68.27	89.69	64.52	68.21	59.85	67.18 (−0.81)
	3	74.17	75.75	69.23	88.66	65.49	70.77	59.41	67.68 (+0.50)
GPT-4o	0	71.90	73.25	61.10	86.49	63.79	66.15	65.72	66.76
	1	70.00	68.85	66.09	84.94	65.28	69.63	66.78	68.18 (+1.42)
	2	67.77	67.81	67.78	83.86	67.18	70.53	67.67	69.17 (+0.99)
	3	72.58	71.64	70.76	87.11	66.97	71.00	66.94	70.12 (+0.95)
GPT-4.1	0	73.39	75.00	64.18	87.37	65.57	74.00	67.59	69.02
	1	78.23	73.46	73.16	88.60	67.77	72.86	68.24	71.67 (+2.65)
	2	78.15	74.03	73.42	86.77	68.82	74.21	67.59	71.76 (+2.09)
	3	75.21	74.37	73.16	88.89	**71.04**	76.02	68.03	72.71 (+0.95)
GPT-5	0	75.00	77.53	70.07	88.40	67.61	**79.50**	68.32	71.51
	1	**79.84**	**79.70**	**74.00**	90.21	68.97	74.00	**68.45**	**72.90** (+1.39)
	2	72.58	77.24	72.00	90.72	68.94	75.50	67.40	71.95 (−0.95)
	3	75.00	79.10	73.44	89.95	70.72	73.50	66.21	72.45 (+0.50)
InternVL3-8B	0	54.03	51.12	54.69	64.18	47.25	54.00	51.63	52.16
	1	68.55	52.61	54.33	68.30	53.06	66.00	57.98	57.06 **(+4.90)**
	2	56.45	57.46	52.64	69.59	53.52	66.50	55.21	56.19 (−0.87)
	3	59.68	58.21	55.77	77.32	56.27	62.50	54.56	58.07 (+1.88)
Qwen2.5-VL-7B	0	50.00	51.12	50.00	72.16	55.58	58.00	53.26	55.01
	1	55.65	52.61	55.53	67.27	51.38	57.00	51.47	54.07 (−0.94)
	2	58.06	55.97	52.76	71.13	55.66	55.50	54.89	56.35 (+2.28)
	3	67.50	56.10	58.00	73.31	55.00	60.00	54.15	57.57 (+1.22)
LLaVA-Video-7B	0	58.87	57.84	51.32	70.36	54.74	58.00	54.89	55.98
	1	61.29	58.96	54.69	72.16	53.90	58.50	55.21	56.78 (+0.80)
	2	58.06	56.34	52.40	68.04	54.82	60.00	55.05	56.03 (−0.75)
	3	62.10	57.84	55.65	77.32	55.66	60.50	56.43	58.35 (+2.32)
Qwen2.5-VL-72B	0	59.84	58.49	54.39	78.50	57.26	63.50	55.13	58.43
	3	59.00	65.81	60.53	84.07	61.35	68.25	59.06	63.16 (+4.73)

E.2. Results for Traditional Models

We also evaluate traditional models (i.e., models that are not VLMs) on VideoNet. In particular, we evaluate 4 recent CLIP models [38, 52, 53, 61] and the 3 convolutional neural networks (CNNs) from [9]. All of the CLIP models except [61] were designed for video inputs; following [52], we uniformly sample 8 frames from the video and average their features when evaluating [61].

These models do not natively support visual question answering with natural language. They also cannot be provided multiple in-context videos. Hence, we adapt our few-shot evaluation setup for these traditional models. We have two separate adaptations: one designed for the CLIP models, one for the CNNs.

In the first, we get CLIP scores for all clips in VideoNet with their corresponding all-lowercase text labels formatted as "«DOMAIN» «ACTION»" (e.g., "figure skating biellmann spin"). We then search for the *optimal threshold* on a balanced[5] validation set constructed from clips in VideoNet which do NOT appear in the test set. To do so, we compute the validation accuracy for all candidate thresholds where the validation accuracy can change.[6] Concretely, if the CLIP score exceeds or equals the threshold, the CLIP model's answer to the test set question is considered "yes"; otherwise, the answer is considered "no". At last, after finding the optimal threshold on the validation set, we present the model with the test set, which contains the same pairs of clips and actions that VLMs see in the normal VideoNet evaluation setup. The results for this setup are in Table 12. The CLIP models struggle immensely, falling short of every VLM we tested. To alleviate concerns that the validation set may have been too small to find a decent threshold, we also search for the optimal threshold directly on the test set in Table 13. Still, the CLIP models struggle, suggesting that they are ill suited for this task.

Table 12. **CLIP results**. Even the best CLIP model fails to match the worst VLM we evaluated, InternVL3-8B. Given that random chance is 50%, these results indicate that CLIP models struggle on the domain-specific action recognition task. NB "acc." is short for accuracy.

Model	Test Acc. (%)	Val Acc. (%)	Threshold
ViCLIP	51.45	52.21	0.2002
LongCLIP-L	52.14	52.44	0.6938
VideoCLIP-XL-v2	50.67	52.21	0.2087
X-CLIP-L/14	50.18	51.75	0.1224

[5]Here, "balanced" denotes that, if the validation set is thought of as containing binary questions, then precisely half the validation set contains binary positive questions (i.e., binary questions where the answer is "yes").

[6]Our validation set contains 2,174 questions. Hence, there are at most 2,175 critical points at which the validation accuracy can change.

Table 13. **CLIP results when "cheating"**. The optimal threshold is now computed over the test set rather than the validation set. The poor results remain, with the best CLIP model beating only the worst VLM, InternVL3-8B. This suggests that the CLIP architecture, rather than the size of our validation set, is at fault for the CLIP models' lackluster performance on VideoNet.

Model	Test Acc. (%)	Threshold
ViCLIP	53.47	0.2121
LongCLIP-L	53.29	0.7400
VideoCLIP-XL-v2	52.83	0.1969
X-CLIP-L/14	51.10	0.1272

The CNNs extract video features from videos, but do not provide a way to align these video features to text. Accordingly, we opt for a k-nearest neighbors classifier (kNN) approach in evaluating the CNNs. In particular, we extract video features from the 3 in-context examples provided in VideoNet for each action and use these features as the support set for a kNN. The kNN then classifies the test samples based on Euclidean distance. We try all $k \in \{1, 2, 3\}$. It is worth noting that no two VideoNet clips for any given action are taken from the same source video, minimizing concerns about a kNN "hacking" correct answers via factors like the video background. The kNN is deemed to answer "does the following video show X" with a "yes" if it classifies the test sample as action X, and "no" if it classifies the test sample as another action. For comparison, we also evaluate the two best CLIP models using this approach by feeding their video features to a kNN. As shown in Table 14, the best CNN, Two-Stream I3D, rivals the best CLIP models. In doing so, it falls short of most open models and considerably far behind all closed models. Similar to CLIP, the I3D models seem poorly suited for the domain-specific action recognition task.

Table 14. **CNN results**. Rows sorted by top-5 accuracy on Kinetics [27]. With some exceptions, higher VideoNet accuracy tends to be correlated with better performance on Kinetics.

Model	VideoNet Accuracy (%)			Kinetics Top-5 Accuracy (%)
	$k = 1$	$k = 2$	$k = 3$	
ViCLIP	54.58	53.73	53.36	98.2
Two-Stream I3D	54.55	53.26	53.03	91.3
RGB-I3D	53.52	52.56	52.09	89.3
Flow-I3D	53.03	52.46	52.45	84.9
LongCLIP-L	54.12	52.87	52.55	-

E.3. Prompt Sensitivity & Yes/No Bias

We observe that model performance on positive clips and negative clips changes significantly when in-context examples are provided (see Table 16). Given the poor performance of open models on our benchmark, we focus on analyzing

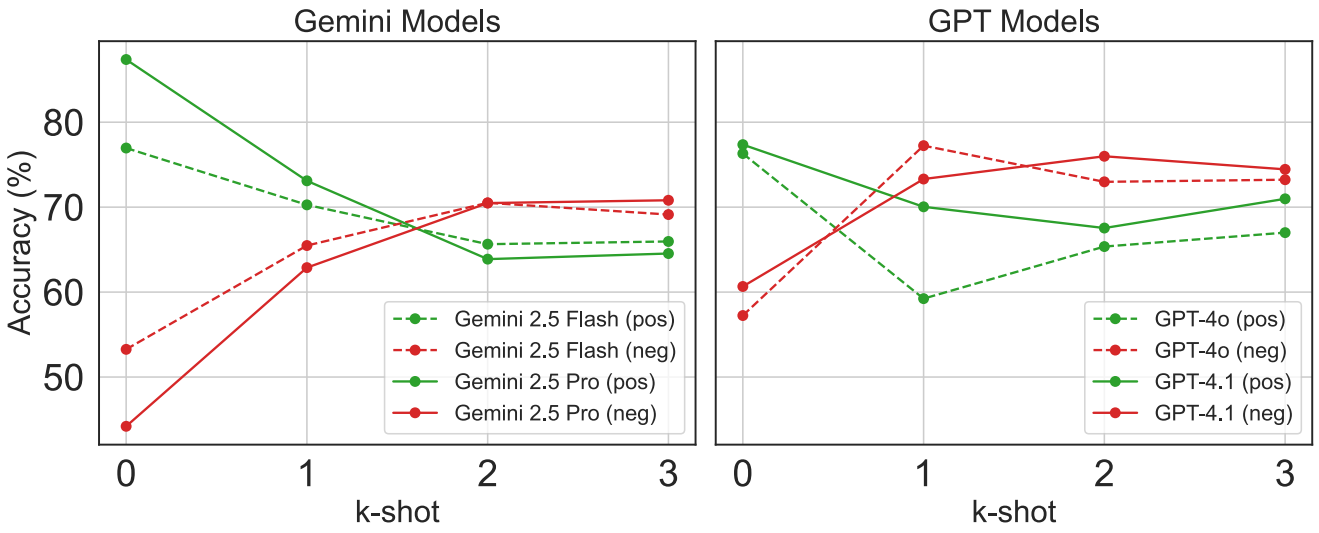


Figure 18. **Positive & negative accuracy with in-context examples.** Accuracy on positive clips is in green; accuracy on negative clips is in red. In both plots, the weaker model is shown with dashed lines, while the stronger reasoning model is shown with solid lines. Note that the GPT models (right), which attain a higher accuracy on VideoNet than the Gemini models (left), see smaller changes in their yes/no bias as additional few-shot examples are provided.

the behavior of Gemini and GPT models (see Figure 18).

Gemini 2.5 Pro exhibits a stark pattern, performing better on negative clips and worse on positive clips as additional in-context examples are provided. GPT-4.1 exhibits a similar pattern, but to a much lesser (and thus, "more acceptable") extent. We believe there are two main hypotheses to explain this phenomenon. One is that Gemini 2.5 Pro over-emphasizes insignificant details from the the in-context examples (e.g., background composition, camera angle, etc.) as opposed to the fine-grained details of the action at-hand. The other is this behavior can be attributed to our prompt.

We test the latter hypothesis by constructing two prompts (see): a "lenient" prompt which should bias models towards saying "yes", and a "balanced" prompt which attempts to eliminate any unintended bias introduced by few-shot examples. (As discussed previously, our "default" prompt seems to bias the model twards saying "no".) We tailor these prompts based on how they impact performance in the weaker models (Qwen, LLaVA, Intern, Gemini), before evaluating their impact on the two strongest models (GPT-4o and GPT-4.1). Table 15 confirms that even the strongest **models are NOT robust to slight changes in the prompt**. Surprisingly, the *overall accuracy is relatively unaffected* by these changes.

Given that small differences in the prompt cause dramatic shifts in yes/no accuracies, we hypothesize that such "prompt sensitivity" is an indicator that these models are not confident in their answers. This is reminiscent of early generations of LLMs, which were often not confident in their answers and hence would easily change their answers based on the smallest of pushback from the user [63].

Table 15. **Prompt sensitivity in GPT models.** The first table shows numbers for GPT-4o; the second shows numbers for GPT-4.1. Reading the first two numeric columns top-to-bottom, we see that accuracies on positive and negative clips change drastically with the choice of prompt. However, reading down the last column, the accuracy across all clips exhibits minimal change. The trend remains equally present in both tables despite GPT-4.1's stronger performance (69.02% vs 66.76%) on VideoNet.

Prompt	Accuracy (%)		
	Positive Clips	Negative Clips	All Clips
Default *biases "no"*	67.00	73.23	70.12
Balanced *minimal bias*	77.55	62.46	70.11
Lenient *biases "yes"*	87.20	55.21	71.22

Prompt	Accuracy (%)		
	Positive Clips	Negative Clips	All Clips
Default *biases "no"*	70.98	74.45	72.71
Balanced *minimal bias*	84.73	59.44	72.12
Lenient *biases "yes"*	90.13	53.94	72.06

Table 16. **Performance on positive vs. negative clips with in-context examples.** Although the benchmark contains the same number of positive and negative clips, the entries in the last column may not be exact averages of the entries in the prior two columns. This is because certain clips are incompatible inputs for certain models (e.g., Gemini rejects certain American Football videos as being too violent).

Model Name	k**-shot**	**Positive Clips**	**Negative Clips**	**Overall**
Gemini 2.5 Flash	0	76.96	53.26	65.14
	1	70.27	65.48	67.88
	2	65.64	70.51	68.07
	3	65.96	69.13	67.55
Gemini 2.5 Pro	0	87.38	44.20	65.78
	1	73.10	62.88	67.99
	2	63.88	70.48	67.18
	3	64.54	70.81	67.68
GPT-4o	0	76.31	57.24	66.76
	1	59.23	77.24	68.18
	2	65.36	72.98	69.17
	3	67.00	73.23	70.12
GPT-4.1	0	77.36	60.66	69.02
	1	70.04	73.31	71.67
	2	67.54	75.99	71.76
	3	70.98	74.45	72.71
GPT-5	0	73.16	69.86	71.51
	1	61.10	84.69	72.90
	2	58.26	85.64	71.95
	3	60.12	84.77	72.45
InternVL3-8B	0	48.94	55.38	52.16
	1	62.19	51.93	57.06
	2	59.61	52.76	56.19
	3	68.68	47.47	58.07
Qwen2.5-VL-7B	0	51.61	58.42	55.01
	1	20.98	87.17	54.07
	2	32.84	79.85	56.35
	3	36.12	78.90	57.57
LLaVA-Video-7B	0	78.79	33.16	55.98
	1	29.35	84.22	56.78
	2	47.70	64.35	56.03
	3	65.82	50.87	58.35
Qwen2.5-VL-72B	0	59.45	57.42	58.43
	3	72.64	53.44	63.16

Default 3-shot Prompt

```
The following 3 videos show examples of <a OR an> <ACTION>, which is <a OR an>
<SUBDOMAIN> in <DOMAIN>.

<VIDEO EXAMPLES>

Now consider the following video. Is it also <a OR an> <ACTION>?

Please reason through your answer. It is critical that you output
'yes' or 'no' on the final line of your answer.
```

Balanced 3-shot Prompt

```
The following 3 videos show examples of <a OR an> <ACTION>, which is <a OR an>
<SUBDOMAIN> in <DOMAIN>.

<VIDEO EXAMPLES>

Now consider the following video. Is it also <a OR an> <ACTION>?

An appropriate instance of <ACTION> must include all essential defining
elements, but minor variations or slight differences in style or execution
are acceptable. Analyze carefully, explicitly noting the presence or absence
of essential elements, while considering natural variations. Clearly explain
your reasoning and justify your final decision. It is critical that you output
'yes' or 'no' on the final line of your answer.
```

Lenient 3-shot Prompt

```
The following 3 videos show examples of <a OR an> <ACTION>, which is <a OR an>
<SUBDOMAIN> in <DOMAIN>.

<VIDEO EXAMPLES>

Now consider the following video. Is it also <a OR an> <ACTION>?

Focus on identifying the core defining elements rather than expecting an
exact match to the examples. The action may have natural variations in
execution while still being the same action. Please reason through your
answer. It is critical that you output 'yes' or 'no' on the final line
of your answer.
```

Figure 19. **Default, Balanced, and Lenient Prompts.** Observe that there are only small differences between each prompt.

F. Benchmark Qualitative Analysis

In Section 5.1, we offer a systematic analysis of why VLMs struggle on VideoNet. Here, we take a qualitative approach to understanding what causes VLMs to fail at the domain-specific action recognition task.

Open models–which are weaker across existing video benchmarks–suffer from failures that are not unique to our tasks. This includes degenerative behavior such as generating the same sentence repeatedly until the max token count is reached, failing to follow the prompt or answer the question, and reasoning towards one answer but outputting a different final answer. They also fail due to fundamental perception errors, i.e., claiming that an atomic action (such as a soccer ball hitting a goal post) occurs when it simply does not.

Among closed models, the vast majority of failures arise from these aforementioned perception failures (see Figure 22). The failures persist despite the inclusion of in-context examples. Furthermore, the detailed reasoning presented by these models–even when no action definition is provided–confirms that *their language backbones have a sufficient understanding of the actions* in VideoNet to answer the questions. (See Figure 20 for a zero-shot example where the model has adequate world knowledge; see Figure 21 for a similar example with in-context demonstrations.) Thus, it is the vision component of these systems that limits their performance.

We provide three concrete examples of model success and failure below. These examples are sourced from Gemini 2.5 Pro because of its extensive reasoning traces. We use examples from the Figure Skating domain since (1) its questions are especially difficult since actions usually cannot be answered with a single frame, and (2) its short clip length allows the videos to be compressed into the storyframes printed below.

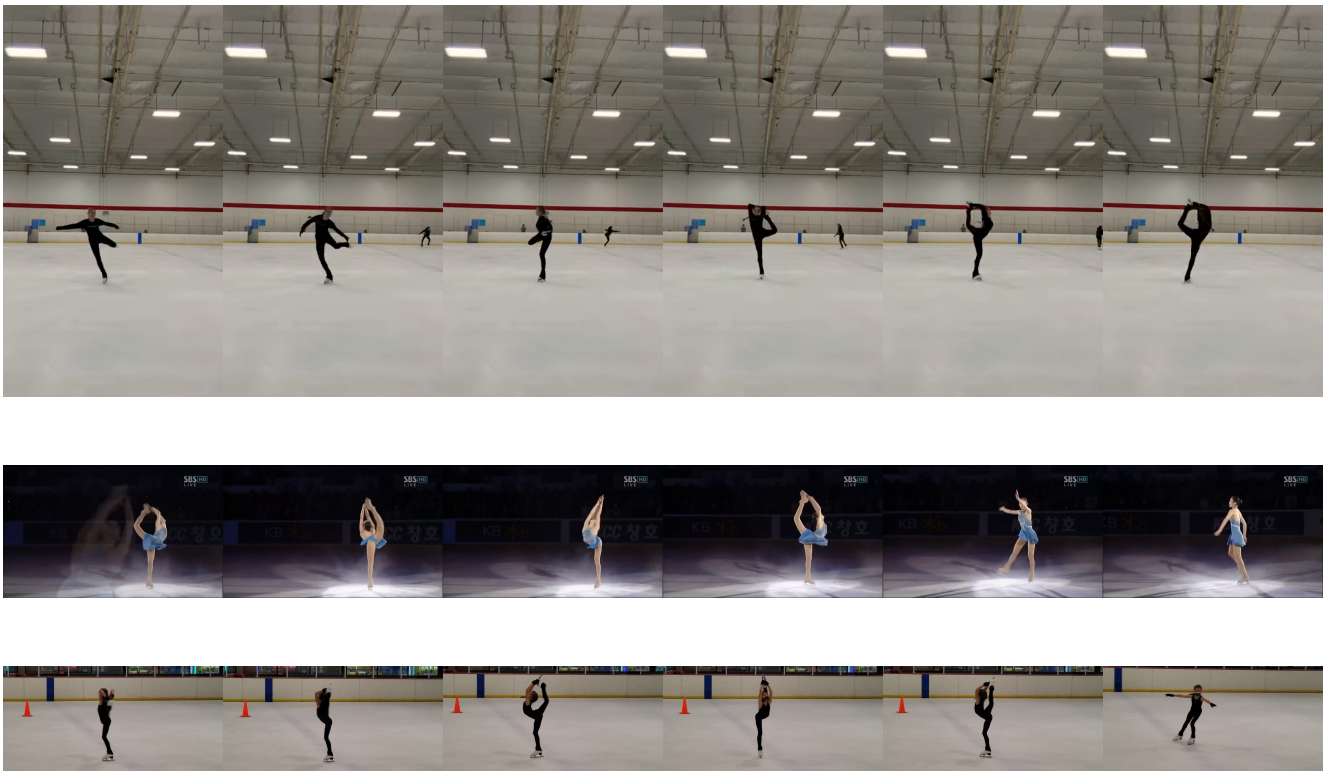
```
A Biellmann Spin is characterized by the skater spinning in an upright
position while pulling their free leg up from behind towards their head,
holding the skate blade with both hands, so the leg is extended vertically
above and behind the head.

In the video:

The skater is performing an upright spin.
The skater reaches back with both hands to grab the blade of their
free skate.
The skater pulls the free leg up so that it is extended vertically above
and behind their head.

All the defining characteristics of a Biellmann Spin are present in the
video from approximately 0:01 to 0:04.
```

Figure 20. **Language grounding success.** In a 0-shot setting without the action definition, Gemini 2.5 Pro provides the *correct* response above when asked if the video above contains a Biellmann Spin. This demonstrates the ability of the strongest closed models to clearly delineate the visual elements that must be present in the video for the answer to be “yes".



```
Okay, let's break down the characteristics of a Biellmann Spin based on the
first two videos:

1. Upright Spin: The skater spins on one leg while maintaining a generally
upright torso position (though arched backward).
2. Free Leg Position: The skater lifts their free leg backward and upward.
3. Grip: The skater reaches back with both hands to grasp the skate blade
(or sometimes the boot) of the free leg.
4. Final Pose: The free leg is pulled up high, typically above and behind
the head, causing the skater's back to arch significantly. This creates a
distinctive teardrop-like shape.

Now let's examine the third video:

-   The skater is performing an upright spin on one leg.
-   She lifts her free leg backward and upward.
-   She reaches back with both hands and grasps the skate blade of her
    free leg.
-   She pulls the leg up above and behind her head, arching her back
    into the characteristic Biellmann pose.

All the key elements identified in the example videos are present
in the third video.
```

Figure 21. **Visual grounding success.** In a 2-shot setting without the action definition, Gemini 2.5 Pro is provided the first two videos and asked if the third video contains a Biellmann Spin. It provides the *correct* response printed above.

```
A Biellmann Spin is characterized by the skater spinning in an upright
position while pulling one leg up from behind towards their head, with the
skate typically held above and behind the head.

In the video:

The skater begins an upright spin.
Around the 0:05 mark, she reaches back with both hands, grabs the
blade of her free skate, and pulls her leg up over her head.
She holds this position, spinning with her leg extended vertically above
and behind her head, until around the 0:10 mark.

This action perfectly matches the description of a Biellmann Spin.
```

Figure 22. **Perception failure example.** In a 0-shot setting without the action definition, Gemini 2.5 Pro provides this *incorrect* response. Crucially, the model claims that the skater "reaches back with both hands and grabs the blade of her free skate" – this never happens in the frames above.

G. Human Evaluation

We have four versions of the human evaluation UI, depending on if the human is shown few-shot examples and whether they are shown the action definition. For brevity, only one of these setups is shown. The HTML/CSS for all four configurations is available on our GitHub repository. Both humans and models are shown silenced videos since the clips sometimes have audio containing the action name.

In Table 17, we show the breakdown of human performance on VideoNet with different configurations, namely 0-shot vs. 3-shot and with vs. without definition. We additionally report the performance with random negatives to better understand the sources of human errors.

Across the board, we see humans excel at identifying positive clips, achieving high accuracy (above 85%) even without definitions or examples. They even attain accuracies above 91% when provided with examples (in the 3-shot setting). However, humans struggle with identifying negative clips, especially in the hard negative setup. Despite being given 3 example videos and a definition, humans get only 71.92%, while the 0-shot with-definition configuration attains a mere 51.58%.

Promisingly, we see a steady improvement in negative clip accuracy as more in-context examples and the action definition are provided. In fact, 3-shot humans armed with action definitions achieve notably high accuracy on random negatives (95.42%), nearly solving the task.

Overall, these findings suggest that while providing definitions and in-context examples significantly helps humans distinguish general in-domain actions, additional domain expertise or perceptual skills might be needed to reliably differentiate highly similar actions.

Table 17. **Human performance on VideoNet**. Metrics are reported separately for positive clips, negative clips, and overall accuracy across different negative sampling strategies (hard vs random). For reference, the best 3-shot video model, GPT-4.1, achieves an overall accuracy of 72.71% and 77.43% on hard and random negatives, respectively.

Human Evaluation	Positive Clips	Negative Clips	Overall
Hard Negatives			
0-shot without definition	85.96	43.27	64.61
0-shot with definition	86.53	51.58	69.05
3-shot without definition	91.98	65.61	78.80
3-shot with definition	**93.41**	**71.92**	**82.66**
Random Negatives			
0-shot with definition	**93.41**	69.63	81.52
3-shot with definition	91.69	**95.42**	**93.55**

H. Additional Training Details

This appendix elaborates on Section 4.

H.1. Dataset Construction

In Section 4.1 we explained how we derive sets of clips with one action label each. Here we walk through the construction of VQA pairs from those labeled clips.

During training, we construct three questions from each clip: one binary question where the answer is "yes" (i.e., binary positive), one binary question where the answer is "no" (i.e., binary negative), and one multiple-choice question (i.e., MCQ). For the binary negative question, we randomly select one action that is not the ground truth from the action list for that domain. For the MCQ, we randomly choose three negative options that are not the ground truth from the action list for the relevant domain. Although the VideoNet benchmark only consists of binary questions, initial experiments showed that including MCQs in the training mixture improves binary accuracy. We also experimented with 10-way MCQs (i.e., a MCQ with 9 negative distractors), but decided against it because it induced a much higher *binary bias* (which we define as the absolute difference between binary positive accuracy and binary negative accuracy).

H.2. Training Setup

In Section 4.2 we detailed the model architecture and our frame sampling approach. Here we include additional information on our training procedure. During training, we train the ViT, the connector, and the LLM using learning rates 5×10^{-6}, 5×10^{-6} and 1×10^{-5} respectively. We employ a cosine learning rate decay to 0.1 of the initial learning rate. Following [13], the connector uses features from the third-to-last and ninth-from-last ViT layers. For each frame, 3×3 patch windows are pooled into a single vector using a multi-headed attention layer, where the mean of the patches serves as the query and the pooled features are projected using an MLP to the LLM's token space. For each training video sample, we pack multiple question-answer (QA) pairs. The LLM attention mask is customized such that text from one QA pair does not attend to the text from another pair. (As mentioned above (§H.1), each video clip is accompanied by three QA pairs.) For additional inquires about the model, please refer to [11].

I. Data Filtering Strategies

The three data filtering strategies we employed are briefly described in Section 4.1. Here we explain our intuition behind each filtering strategy, the per-domain yields of each strategy, the category-level results of post-training a Molmo2-4B model on each strategy, and a brief analysis of these training results.

We began with the hypothesis that having as many independent signals align as possible would yield the highest-quality labels. There were two signals that were easily extracted at scale: the presence of an action in the video's title ("title match"), and the presence of an action in the video's transcript ("transcript match"). Adhering to our philosophy of having an *extremely strict filter*, we chose to require the action to be said within one second of the clip for the "transcript match" to count. This resulted in the TRANSCRIPTLOCALIZEDTITLEMATCH filter. While our hypothesis of such a strict filter yielding high-quality data was largely confirmed by initial experiments on domains like skateboarding, this filter's yield was too low on domains like whittling and fencing (see Table 18). A natural solution to increasing the number of clips yielded by a filter is to relax the filter's strictness. Hence, we dropped the title match requirement, thereby keeping all clips with a transcript match; this is the TRANSCRIPTLOCALIZED filter. In many cases, TRANSCRIPTLOCALIZED yielded more clips than TRANSCRIPTLOCALIZEDTITLEMATCH, largely solving our problem of low yields. Once we had derived a filter (TRANSCRIPTLOCALIZED) by relaxing the title match requirement of TRANSCRIPTLOCALIZEDTITLEMATCH, it seemed fitting to derive a filter by relaxing the transcript match requirement. After some experimentation, we landed on SINGLEACTION. The intuition here is that if there is a title match, then the video is likely to contain at least one clip of that action; if our localizer only finds one clip of that domain, then that clip must be of the title action. To make an analogy to the classic pigeonhole problem, if there is one pigeon (i.e., action from the title) and only one hole (i.e., clip found by localizer), then the pigeon must be assigned to that hole (i.e., the title action must be assigned to the one and only clip). Thus we arrived at our filtering strategies.

We train three models, one each for the data yielded by each filtering strategy. The overall accuracies of these models are reported in Table 5, as are category-level results. Domain-level results are in Table 7. Even though SINGLEACTION attains the best overall performance on VideoNet, it only achieves the best performance on 19 out of 38 domains, affirming the domain-to-domain variation in filtering strategy effectiveness.

Perusing these tables, the question naturally arises: why do certain filtering strategies fare better than others in terms of downstream performance on VideoNet? Unlike other tasks [24] where dataset size has a profound impact on downstream performance, the filter with the best VideoNet performance is actually the smallest in size. Hence, scale itself cannot explain the differences in downstream performance. Rather, we hypothesize that downstream performance is primarily impacted by *clip quality and intra-domain uniformity*. Concretely, clip quality refers to the accuracy with which action labels are assigned to clips by a filtering strategy, and intra-domain uniformity refers to the extent to which the counts of clips labeled by each action (within a domain) follows the uniform distribution. The intuition for the former is trivial; for the latter, since the test set presents a uniform # of questions for each action in a domain, we believe that a training dataset which contains an equal numbers of clips for each action within a domain is poised to perform best.[7]

Clip quality is difficult to measure at a statistically significant scale without employing a large army of experts, so we focus on quantifying the intra-domain uniformity. For each of the 38 domains, we calculate the Shannon entropy for the distribution of clips yielded by each of the three filtering strategies.[8] For each domain, this yields three entropy numbers; one per filtering strategy. Recall that a higher entropy suggests a more uniform distribution [14]. Hence, for each domain, we can order the filtering strategies by entropy; in doing so, we are ordering the filtering strategies by the uniformity of their data for that domain. We can also order the filtering strategies by their downstream accuracy on that domain's subset of VideoNet. This gives two orderings for each domain. If the orderings are identical, then for that domain there is undeniably a correlation between higher entropy (and thus higher intra-domain uniformity) and downstream performance.[9] In our case, there are **10 domains** where the orderings are identical. Recall that we are trying to ascertain whether there is a correlation between intra-domain uniformity and downstream performance; the existence of such a correlation would result in *more* than average identical pairings, so we shall test in that direction.

Since there are 3 filtering strategies, there are $3! = 6$ possible orderings, and $3! \times 3! = 36$ pairs of orderings. Of these 36 pairs, only 6 pairs exist where both orderings are the same. Thus, there is a $p = \frac{6}{36} = \frac{1}{6}$ probability of two 3-item orderings being the same. Since we repeat this analysis of comparing entropy orderings and accuracy orderings for 38

[7] Additionally, given that certain filtering strategies yield quite skewed distributions for certain domains–e.g., the TRANSCRIPTLOCALIZED gym data contains nearly 30k clips of `squats`–we believe that seeing such a disproportionate number of squat clips during training will make the model worse at discerning other gym actions such as pushups or deadlifts.

[8] The "distribution of clips" is a list of integers, where each integer gives the count for the number of clips labeled with a particular action. This list's length equals the number of actions in that domain.

[9] Of course, a correlation could exist even if the orderings are not exactly the same, but statistical tests like Spearman's rank correlation or the Pearson correlation do not provide statistically significant results for $n = 3$. Hence we are forced to limit our analysis to cases where there is a perfect correlation (in these cases, Spearman's coefficient is 1.)

domains,[10] this process can be modeled by a binomial distribution with $n = 38$ and $p = \frac{1}{6}$. Formally, $X \sim \text{Bin}(38, \frac{1}{6})$ where X is the number of domains with identical pairings. Let the null hypothesis be that this process is purely random (i.e., that entropy has no effect on accuracy). Since we are wondering if the number of identical pairings is *better* than average, we can use a one-tailed test. We compute $P(X \geq 10) = 0.08902$. A p-value of 0.089 is ambiguous. Under the commonly-used significance level of 0.05, we would fail to reject the null hypothesis and find that there is no correlation between intra-domain uniformity and downstream accuracy. However, given that we hypothesized that clip quality *and* intra-clip distribution impact downstream performance but only tested for the latter, we believe this finding actually supports our hypothesis.[11] In other words, we advocate for interpreting the p-value of 0.089 as suggesting both a correlation between intra-domain uniformity and downstream performance *and* the presence of a confounding variable. Namely, we believe this confounding variable to be clip quality, although we present no evidence in support of this claim.

[10] We assume the yields and accuracies to be independent between domains.

[11] In this sentence, "hypothesis" refers to our original hypothesis from three paragraphs ago, not the null hypothesis established in this paragraph.

Table 18. **Filtering strategy yields.** The last three columns list yields for the filtering strategies in decreasing order of total yield: TRANSCRIPTLOCALIZED, TRANSCRIPTLOCALIZEDTITLEMATCH, and SINGLEACTION. For a given row, compare the relative ranking of values in the last three columns of this table to the relative ranking of values in the last three columns of Table 19; such a comparison proves that the yield of a filtering strategy is a poor indicator of downstream performance on the VideoNet benchmark.

Category Name	Domain Name	# Actions	Transcript Localized	Transcript Localized Title Match	Single Action
Beauty & Self Care	Hairstyling	14	**5,775**	2,029	1,401
	Spa Massage	11	**3,259**	1,351	759
	Tattooing	6	**782**	145	**782**
Crafts & Art	Calligraphy	8	**5,508**	231	105
	Crochet	38	7,565	4,572	**10,990**
	Hand Sewing / Embroidery	41	688	460	**6,552**
	Knots	55	**4,099**	3,889	20,371
	Painting	8	**2,949**	1,472	489
	Pottery	10	**5,889**	3,077	817
	Woodworking / Whittling	4	**1,140**	29	11
Dance	Ballet	39	**17,362**	5,567	3,471
	Bharatanatyam	24	862	316	**3,200**
	Break Dance	34	**2,573**	1,507	395
	Salsa	21	**8,436**	2,036	2,432
	Tap Dance	29	**13,130**	3,579	1,400
Food & Beverage	Bartending	30	2,017	1,273	389
	Coffee	16	**3,369**	2,440	611
	Cooking	51	**82,814**	35,530	2,878
Hobbies	Bouldering	23	2,275	950	**5,391**
	Gardening	20	**4,064**	2,298	1,471
	Gym	22	**79,793**	68,869	21,333
	Juggling	26	**1,582**	941	348
	Parkour	40	6,425	4,035	**7,109**
	Pen Spinning	33	**6,877**	2,827	2,318
	Skateboarding	49	**52,851**	10,079	16,910
	Yo-yo	55	**6,754**	3,530	2,257
Medical	Neurological Abnormalities	21	**2,912**	983	433
	Neurological Assessments	15	**820**	381	319
	Suturing	14	751	417	**1,443**
Sports	American Football	54	**36,327**	13,610	11,664
	Basketball	46	**82,883**	13,213	11,219
	Cheerleading	23	1,027	622	**3,504**
	Cricket	46	**8,457**	3,614	7,033
	Fencing	20	**2,058**	667	918
	Figure Skating	40	**17,525**	2,709	4,592
	Ice Hockey	39	**4,258**	1,960	2,247
	Soccer	43	**12,120**	4,251	4,388
	Tennis	19	**3,404**	2,128	1,187
All	*All*	1,087	**501,380**	207,587	163,137

Table 19. **Per-domain performance of different filtering strategies.** The last three columns contain accuracy percentages for the three filtering strategies in decreasing order of total yield: TRANSCRIPTLOCALIZED, TRANSCRIPTLOCALIZEDTITLEMATCH, and SINGLEACTION. Please keep the number of questions for each domain, listed in the third column, in mind when considering the significance of a change in accuracy. Note that the base model achieves the best performance for calligraphy, coffee, and fencing, suggesting that our training data for these domains is poorly-labeled.

Category Name	Domain Name	# Questions	Base Model	Transcript Localized	Transcript Localized Title Match	Single Action
Beauty & Self Care	Hairstyling	56	50.00	73.21	**76.79**	75.00
	Spa Massage	44	52.27	**81.82**	72.73	**81.82**
	Tattooing	24	54.17	62.50	**66.67**	62.50
Crafts & Art	Calligraphy	32	**56.25**	53.13	**56.25**	53.13
	Crochet	152	51.97	57.24	60.53	**65.79**
	Hand Sewing / Embroidery	164	42.68	63.41	56.71	**73.17**
	Knots	220	49.09	67.27	67.27	**73.18**
	Painting	32	62.50	68.75	62.50	**71.88**
	Pottery	40	57.50	**77.50**	70.00	70.00
	Woodworking / Whittling	16	50.00	56.25	**62.50**	50.00
Dance	Ballet	156	51.92	69.23	65.38	**69.87**
	Bharatanatyam	96	53.13	57.29	59.38	**69.79**
	Break Dance	136	66.18	**67.65**	64.71	66.18
	Salsa	84	47.62	**71.43**	67.86	67.86
	Tap Dance	116	50.00	57.76	**63.79**	60.34
Food & Beverage	Bartending	120	74.17	**90.00**	80.83	77.50
	Coffee	64	**75.00**	73.44	67.19	64.06
	Cooking	204	72.06	73.04	74.51	**78.92**
Hobbies	Bouldering	92	50.00	**58.70**	52.17	51.09
	Gardening	80	68.75	**86.25**	81.25	77.50
	Gym	88	68.18	76.14	75.00	**79.55**
	Juggling	104	48.08	**64.42**	60.58	62.50
	Parkour	160	60.62	71.25	70.63	**71.88**
	Pen Spinning	132	50.00	**68.94**	60.61	66.67
	Skateboarding	196	50.51	55.61	55.61	**63.78**
	Yo-yo	220	50.45	**64.55**	64.09	64.09
Medical	Neurological Abnormalities	84	48.81	**67.86**	60.71	61.90
	Neurological Assessments	60	58.33	73.33	65.00	**76.67**
	Suturing	56	51.79	67.86	66.07	**78.57**
Sports	American Football	216	51.39	55.09	58.80	**59.72**
	Basketball	184	44.02	55.98	59.24	**59.78**
	Cheerleading	92	57.61	79.35	77.17	**88.04**
	Cricket	184	51.09	55.43	58.15	**59.24**
	Fencing	80	**53.75**	51.25	51.25	52.50
	Figure Skating	160	52.50	63.13	62.50	**66.87**
	Ice Hockey	156	55.77	64.74	64.10	**69.23**
	Soccer	172	51.16	55.23	**61.05**	60.47
	Tennis	76	51.32	63.16	**65.79**	60.53
All	*All*	4,348	54.35	65.11	64.21	**67.36**

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, 14

[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 Szpektor, 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, 18

[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, 27

[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] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *CVPR*, 2025. 27

[14] Holger Dell, Dieter van Melkebeek, and Mahnaz Akbari. Lecture 17: Randomness extractors, 2013. 28

[15] 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

[16] 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

[17] Kristen Grauman et al. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives, 2024. 2

[18] 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

[19] Rong Gao, Xin Liu, Zhuozhao Hu, Bohao Xing, Baiqiang Xia, Zitong Yu, and Heikki Kälviäinen. Fsbench: A figure skating benchmark for advancing artistic sports understanding. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13595–13605, 2025. 2

[20] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. Soccernet: A scalable dataset for action spotting in soccer videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. 2

[21] 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, Abrham 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

[22] 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

[23] 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

[24] 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, 28

[25] Derek L. Hansen, Patrick J. Schone, Douglas Corey, Matthew Reid, and Jake Gehring. Quality control mechanisms for crowdsourcing: peer review, arbitration, & expertise at familysearch 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

[26] Wenyi Hong*, Yean Cheng*, Zhuoyi Yang*, Weihan 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

[27] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017. 18

[28] 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

[29] 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

[30] Hongwei Li and Bin Yu. Error rate bounds and iterative weighted majority voting for crowdsourcing, 2014. 3

[31] 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

[32] 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

[33] Shenglan Liu, Xiang Liu, Gao Huang, Hong Qiao, Lianyu Hu, Dong Jiang, Aibin Zhang, Yang Liu, and Ge Guo. Fsd-10: A fine-grained classification dataset for figure skating. *Neurocomputing*, 413:360–367, 2020. 2

[34] Shenglan Liu, Aibin Zhang, Yunheng Li, Jian Zhou, Li Xu, Zhuben Dong, and Renhao Zhang. Temporal segmentation of fine-gained semantic action: A motion-centered figure skating dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 2

[35] Sheng-Lan Liu, Yu-Ning Ding, Gang Yan, Si-Fan Zhang, Jin-Rong Zhang, Wen-Yue Chen, and Xue-Hai Xu. Fine-grained action analysis: A multi-modality and multi-task dataset of figure skating, 2024. 2

[36] 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

[37] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, and Aude Oliva. Moments in time dataset: one million videos for event understanding, 2019. 2

[38] 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, 18

[39] OpenAI. Gpt-4o system card, 2024. 14

[40] OpenAI, 2025.

[41] OpenAI. Gpt-5, 2025. 14

[42] Yulu Pan, Ce Zhang, and Gedas Bertasius. Basket: A large-scale video dataset for fine-grained skill estimation. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28952–28962, 2025. 2

[43] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar 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

[44] 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

[45] 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

[46] 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

[47] 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

[48] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding, 2020. 2

[49] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild, 2012. 2

[50] Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. 14

[51] 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

[52] Jiapeng Wang, Chengyu Wang, Kunzhe Huang, Jun Huang, and Lianwen Jin. Videoclip-xl: Advancing long description understanding for video clip models, 2024. 5, 18

[53] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2023. 5, 18

[54] 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

[55] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. 4

[56] Chengming Xu, Yanwei Fu, Bing Zhang, Zitian Chen, Yu-Gang Jiang, and Xiangyang Xue. Learning to score figure skating sport videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4578–4590, 2020. 2

[57] 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

[58] Jinglin Xu, Guohao Zhao, Sibo 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

[59] 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

[60] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi 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 Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024. 4

[61] 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, 18

[62] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data, 2024. 5, 14

[63] Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models, 2021. 19

[64] 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 Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. 5, 14