

A Smoothing Analysis

We assess the effect of signal smoothing on model performance. Specifically, we vary the moving-average window size $w \in \{10, 20, 30, 50\}$, applied over the frame-wise SigLIP embeddings before alignment.

Effect on Accuracy. As shown in Figure 7 the Top-1 classification accuracy remains almost stable across a wide range of smoothing widths. This suggests that our method is not overly sensitive to the smoothing parameter.

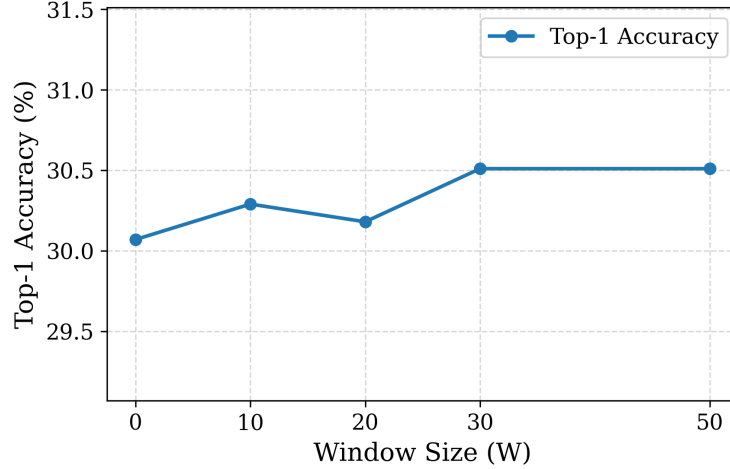


Figure 7: Top-1 accuracy for different smoothing window sizes w . Accuracy is stable across settings, with $w = 30$ selected as default.

Choice of Default. We fix $w = 30$, which corresponds to a 1-second temporal window under 30 FPS videos. This setting balances local context preservation with jitter reduction and yields smooth DTW alignments that more cleanly segment sub-actions.

B Implementation Details and Inference Pipeline

B.1 System Configuration

We conduct all experiments on a server running Ubuntu 20.04 with a single NVIDIA RTX A5000 GPU (24GB VRAM) and 256GB RAM. The pipeline is implemented in PyTorch 2.7 with CUDA 12.8. For subaction generation, we use the GPT-4o API.

B.2 Model Settings

We use the publicly available SigLIP model from Google (`siglip-so400m-patch14-384`), which supports variable input resolutions. Images are embedded using a ViT-based vision encoder with a patch size of 14×14 and an output embedding dimension of $d = 384$. Sub-actions are encoded using the corresponding frozen SigLIP text encoder. All embeddings are ℓ_2 -normalized before computing cosine similarities.

B.3 Runtime Performance

End-to-end inference over all 898 samples—including affinity matrix computation, signal smoothing, temporal alignment, and classification—completes in 41.95 seconds, with an average processing time of 0.04 seconds per video. Dynamic time warping (DTW) alignment accounts for over 90% of the total runtime.

B.4 Scalability Analysis

From an algorithmic standpoint, the scalability is primarily determined by the dynamic time warping (DTW) alignment between the visual frame sequence and sub-action scripts. The overall complexity scales as $\mathcal{O}(NMT)$, where N is the number of candidate classes, M is the average number of sub-actions per class, and T is the number of video frames. In practice, M remains small (on the order of tens) and independent of video length. Because the method operates entirely in a frozen embedding space and requires no gradient updates, it avoids the substantial memory and computational overhead associated with training large video-language models. As a result, ACTALIGN scales gracefully to longer videos and larger label vocabularies. Moreover, since alignment computations for different candidate classes and frame segments are independent, the framework is naturally amenable to parallelization across GPUs or distributed compute resources, offering a straightforward path for further acceleration in large-scale or real-time settings.

B.5 Inference Pipeline Overview

Our pipeline proceeds in six modular stages: (1) videos from ActionAtlas are downloaded and clipped using annotated timestamps, with frames stored in a structured NumPy archive; (2) each frame is encoded using SigLIP’s vision encoder to obtain per-frame embeddings; (3) GPT-4o generates context-rich sub-action sequences for each candidate label using fixed prompt templates; (4) sub-actions are encoded via SigLIP’s text encoder; (5) cosine similarity matrices are computed between visual and semantic sequences, and DTW is applied to compute alignment scores for each candidate; and (6) as a baseline, we compare mean-pooled frame embeddings with text embeddings of class labels using cosine similarity, bypassing alignment. Each step is modular and implemented for fast inference, allowing to easily switch between models and datasets.

C More Related Works

Image–Language Transfer for Video Tasks. Image–language models pretrained on large-scale image–text pairs exhibit strong open-set recognition capabilities. This motivates their adaptation for video tasks. Common strategies include adding temporal modules Wang et al. (2022); Ni et al. (2022); Rasheed et al. (2023); Liu et al. (2024) or introducing learnable prompts Ju et al. (2022); Lin et al. (2022). However, as noted by Rasheed et al. (2023), fine-tuning often compromises the model’s open-set generalization. To preserve zero-shot capabilities, recent work explores training-free adaptations using temporal prompting methods Ahmad et al. (2024); Phan et al. (2024); Yu et al. (2025). For example, TEAR Bosetti et al. (2025) compares mean-pooled frame embeddings with LLM-generated action descriptions for each class. While effective, these approaches either degrade generalization through fine-tuning or ignore temporal structure via mean-pooling - limiting performance on fine-grained and zero-shot video classification.

D Prompt Templates and Generation Settings

We use GPT-4o to automatically convert each high-level class name into a sequence of sub-actions. Two prompt styles were explored:

D.1 Short-Fixed Prompt.

This setting produces terse outputs (e.g., “tie arms,” “deliver knee”) that often lack domain-specific context. The descriptions are directly adopted from the original ActionAtlas Salehi et al. (2024) dataset.

D.1.1 Template

Generate a JSON object from the following context.

For each action option given below, create a key in the JSON object with that action name.

The value for each key must be a list of exactly 10 subactions.

Each subaction must be a minimalist, two-word description.

The descriptions should capture the essential mechanics of the action.

Use the following action options and their descriptions for context:

- <action 1>: <action 1 description>
- <action 2>: <action 2 description>
- <action 3>: <action 3 description>
- ...

Output the JSON object accordingly.

D.2 Context-Rich Prompt.

This setting encourages richer descriptions with more related keywords grounded in the activity domain. Context-rich prompting strategy significantly improves textual grounding during DTW alignment.

D.2.1 Template

You will output a JSON object. Each key is an action name from the list below, and its value is a list of concise, visually distinctive subaction descriptions performed sequentially.

Do NOT fix the list length—choose however many substeps best break down that trick into discriminative subactions.

Each subaction should be:

- Self-contained and visually descriptive with rich <domain> keywords and context.
- Explicitly reference the scene, objects, environment, and motion in the given sport context.
- Discriminative and concise.

Here are the actions to decompose:

- <action 1> in <domain>
- <action 2> in <domain>
- <action 3> in <domain>
- ...

Output ONLY the JSON object, without any additional explanation.

D.3 Examples

Table 7 compares the two prompting strategies for a skateboarding trick example. The corresponding affinity matrices are shown in Figure 6.

Note. Our experiments demonstrate that context-rich prompts significantly outperform short-fixed ones, as the added context helps disambiguate tactics across sports and yields smoother DTW alignment with video frames.

E Dataset Statistics

Dataset Overview. Table 8 summarizes key statistics of the ActionAtlas Salehi et al. (2024) dataset, which we extended to enable alignmnet-based classification. The dataset contains 898 video clips spanning 558 unique tactics across 56 sports domains. Each video is paired with 4 to 6 fine-grained candidate actions (5.25 in average per video). Videos vary significantly in length, with a mean of 173 frames (30 FPS), ranging from extremely short clips (3 frames) to multi-second sequences (up to 900 frames). Notably, ActionAtlas is the most diverse and challenging fine-grained action recognition benchmark to the best of our knowledge. It poses a major challenge in fine-grained recognition: human top-1 accuracy is just 61.64%.

ActionAtlas Extension. To enable subaction-level alignment, we augment the original ActionAtlas dataset with sub-action scripts generated by a large language model for each candidate action (see Figure 8). This transforms the task from multiple-choice classification into alignment-based classification without modifying the original videos or labels. The extended dataset supports zero-shot video classification via sub-action sequence matching.

Prompt Variant	Generated Subaction Script
Short-fixed	<ol style="list-style-type: none"> 1. flip board 2. spin board 3. rotate body 4. catch board 5. kick upward 6. flip reverse 7. control motion 8. hover steady 9. anticipate landing 10. land cleanly
Context-rich	<ol style="list-style-type: none"> 1. Rider rolls forward on flat ground, knees compressed and eyes focused ahead. 2. Back foot scoops the tail into a backside shuvit while front foot flicks a kickflip off the nose-side edge. 3. As the board begins its varial kickflip rotation, the front foot quickly retracts and then kicks downward under the board’s underside. 4. Board completes a varial kickflip and then reverses flip direction mid-air due to the underflip, creating a complex double-flip motion. 5. Simultaneously, rider initiates a 180-degree body varial, rotating in the same direction as the board’s shuvit. 6. Rider tracks the board’s griptape, completes the body spin, and catches the board with both feet aligned over the bolts. 7. Rider lands smoothly, now facing the opposite direction, and rolls away with balance.

Table 7: Sub-action sequence comparison for the *varial kickflip underflip body varial trick* in skateboarding using different prompt variants. Figure 6 represents the heatmap comparisons.

Category	Statistic	Value
<i>Dataset scale</i>	Videos	898
	Unique fine-grained actions	558
	Sport domains	56
	Avg. candidate actions per video	5.26 ± 1.27
	Avg. candidate actions per domain	33.6
	Min-Max candidate actions per domain	4–201
<i>Temporal properties</i>	Avg. video length (frames)	173.2 ± 121.3
	Min-Max video frames	3–900
<i>Recognition accuracy</i>	Human top-1 accuracy (%)	61.64

Table 8: **Summary of ActionAtlas dataset statistics.**

Sub-action Script Statistics. We generate sub-action sequences using different prompting strategies. As summarized in Table 9, context-rich prompts yield fewer but more descriptive sub-actions, while short-fixed prompts produce uniform-length, terse sequences. These linguistic differences influence the textural grounding, and consequently, sequence alignment.

F Performance within Domains

We evaluate our approach on ActionAtlas, which covers over 50 sports domains while including extremely fine-grained actions, to assess its generality. Specifically, we focus on the four largest domains in the dataset

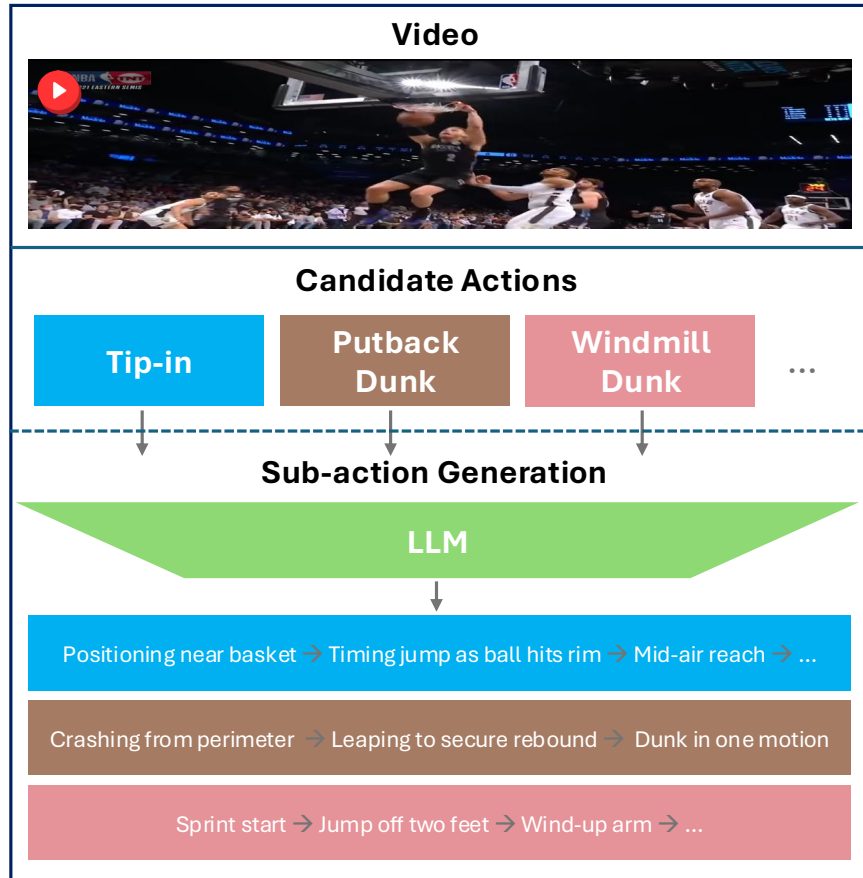


Figure 8: Our pipeline uses an LLM to generate **structured sub-action sequences** for each fine-grained candidate action in ActionAtlas [Salehi et al. \(2024\)](#) data sample. This structured representation enables sequence alignment with video frames for zero-shot recognition.

Prompt Type	Avg. Subactions	Avg. Subactions per Domain	Avg. Words per Subaction
Short-fixed	10.00 ± 0.00	10.00 ± 0.00	2.00 ± 0.03
Context-rich	4.94 ± 0.86	5.01 ± 0.62	13.68 ± 2.78

Table 9: **Linguistic complexity of subaction scripts generated by different prompting strategies.** Context-rich prompts yield fewer but more descriptive subactions with significantly higher word counts compared to the Short-fixed strategy.

and compare performance against the SigLIP baseline (Table [10](#)). Our method consistently outperforms the baseline across all domains.

G Additional Qualitative Example

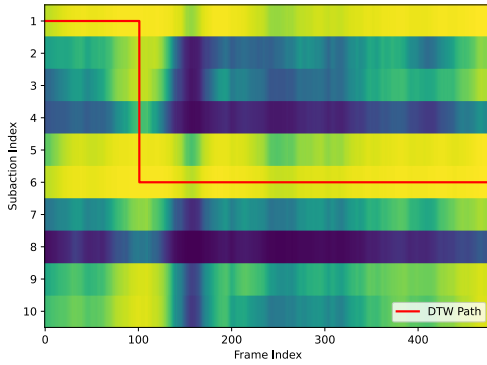
To illustrate how different phrasings of sub-actions can influence alignment behavior, we present an example of *tie-down roping* trick in Rodeo. Figure [9](#) shows the corresponding DTW alignment paths, and Table [11](#) lists the sub-action scripts used in each case.

Domain	Samples	ActAlign (Ours)			SigLIP Baseline			Δ (pp)		
		Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
Soccer	168	27.38	52.98	70.83	23.81	45.83	63.10	+3.57	+7.15	+7.73
Basketball	106	36.79	54.72	70.75	17.92	41.51	58.49	+18.87	+13.21	+12.26
Cheerleading	61	22.95	55.74	67.21	22.95	37.70	60.66	+0.00	+18.04	+6.55
Wrestling	60	26.67	48.33	61.67	16.67	36.67	66.67	+10.00	+11.66	-5.00
Mean	–	28.45	52.94	67.62	20.34	40.43	62.23	+8.11	+12.52	+5.39

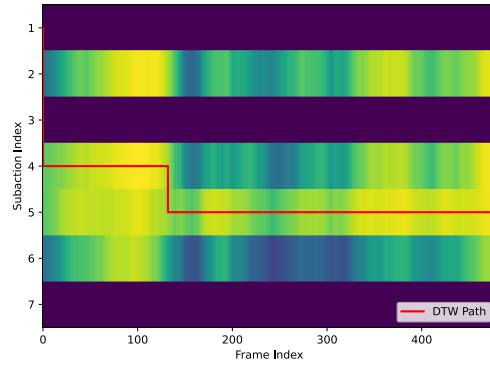
Table 10: **Top- k accuracy (%) comparison between ActAlign and SigLIP baseline across four largest ActionAtlas domains.** ActAlign demonstrates domain-agnostic improvements across all k levels. Δ denotes the absolute percentage-point gain of ActAlign over the baseline. Bold indicates the higher result for each domain and metric.



(a) Video: Performing *tie-down roping* in Rodeo.



(b) Short-fixed sub-actions, $\hat{\gamma} = 0.96$ ✓



(c) Context-rich sub-actions, $\hat{\gamma} = 0.84$ ✓

Figure 9: **DTW alignment paths** comparison for a correctly predicted sample between short-fixed and context-rich sub-actions. The sub-action scripts are provided in Table 11.

H VLM Evaluation Settings

We provide here the detailed evaluation settings for the video–language models reported in Table 1 of the main paper.

Our evaluation strictly follows the official ActionAtlasSalehi et al. (2024) protocol and public leaderboard. Each model is evaluated using multiple-choice question–answer pairs, where the model must select from a fixed set of candidate actions. As part of the standardized pre-processing pipeline, all audio signals, speech transcripts, and any textual cues visible in the video are discarded or blurred prior to inference. This ensures that evaluation reflects purely visual understanding rather than reliance on auxiliary textual or audio information.

Additionally, ActionAtlas provides a comprehensive analysis of how performance varies with different frame rates. The results below correspond to the best-performing frame-rate configuration for each model as reported in the official leaderboard. The number of frames, token configuration, and Top-1 accuracy for each model are shown below:

Step	Short-Fixed Script	Context-Rich Script
1	Pursue calf	Cowboy and horse launch from the roping box as the calf bolts across the arena.
2	Swing lasso	Roper swings the rope fluidly and hurls it in a clean loop that snags the calf’s neck mid-gallop.
3	Catch neck	Rope tightens as horse halts and backs to maintain tension, stopping the calf abruptly.
4	Dismount quickly	Cowboy dismounts in one jump while horse holds the rope taut, sprinting to the flailing calf.
5	Reach calf	Flips the calf onto its side with a quick lift-and-roll maneuver in the loose arena footing.
6	Flip calf	Ties three of the calf’s legs with a piggin’ string in a swift, practiced knot.
7	Tie legs	Throws hands up to signal completion and steps away as the calf remains bound for six seconds.
8	Raise hands	–
9	Step back	–
10	Stop time	–

Table 11: **Comparison of short-fixed and context-rich sub-action scripts for *tie-down roping* in Rodeo.** The DTW alignment paths for both instances are shown in Figure 9

Model	Frames	Tokens	Top-1 Acc. (%)
Qwen2-VL-7B	16	8×576	30.24
VideoLLaMA	16	16×256	22.71
VideoChat2	64	64×196	21.27
mPLUG-Owl-Video	16	16×256	19.49
LLaVA-Next-Video-7B	64	64×144	22.90

Table 12: Evaluation settings and results for baseline VLMs on the ActionAtlas benchmark.

This table reproduces the official ActionAtlas leaderboard conditions under which we report all baseline results in the main text.