
White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/247725/

Version: Published Version

Proceedings Paper:
Li, J., Wang, G., Zheng, S. et al. (2026) Towards mitigating modality bias in vision-language models for Temporal Action Localization. In: Liakata, M., Moreira, V.P., Zhang, J. and Jurgens, D., (eds.) Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 02-07 Jul 2026, San Diego, California, United States. Association for Computational Linguistics, pp. 11087-11104. ISBN: 9798891763906. ISSN: 0736-587X. EISSN: 9061-6464.

https://doi.org/10.18653/v1/2026.acl-long.508

Reuse
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.





Towards Mitigating Modality Bias in Vision-Language Models for Temporal Action Localization

Jiaqi Li[1], **Guangming Wang**[2], **Shuntian Zheng**[1], **Minzhe Ni**[1], **Xiaoman Lu**[1], **Guanghui Ye**[3], **Yu Guan**[1]✉*

[1] UVLab, Department of Computer Science, University of Warwick
[2] Department of Automation, University of Cambridge
[3] Department of Computer Science, The University of Sheffield
{Jiaqi.Li.16, Shuntian.Zheng, Minzhe.Ni, Xiaoman.Lu, Yu.Guan}@warwick.ac.uk,
gw462@cam.ac.uk, gye4@sheffield.ac.uk

Abstract

Temporal Action Localization (TAL) requires identifying both the boundaries and categories of actions in untrimmed videos. While vision-language models (VLMs) offer rich semantics to complement visual evidence, existing approaches tend to overemphasize linguistic priors at the expense of visual performance, leading to a pronounced modality bias. We propose ActionVLM, a vision-language aggregation framework that systematically mitigates modality bias in TAL. Our key insight is to preserve vision as the dominant signal while adaptively exploiting language only when beneficial. To this end, we introduce (i) a debiasing reweighting module that estimates the language advantage—the incremental benefit of language over vision-only predictions—and dynamically reweights language modality accordingly, and (ii) a residual aggregation strategy that treats language as a complementary refinement rather than the primary driver. This combination alleviates modality bias, reduces overconfidence from linguistic priors, and strengthens temporal reasoning. Experiments on THUMOS14 show that our model outperforms state-of-the-art by up to 3.2% mAP. Our code is available at https://github.com/JiaqiLi404/ActionVLM

1 Introduction

Temporal Action Localization (TAL) requires identifying the precise start and end of actions in long, untrimmed videos. This task demands not only robust frame-level recognition but also fine-grained temporal reasoning (Wu et al., 2024). While purely visual models excel at capturing local appearance and motion patterns, they often struggle when actions share highly similar visual frames (Jian et al., 2025). To alleviate such ambiguities, recent research has turned to Vision-Language Models (VLMs), expecting high-level semantics from language to complement visual evidence and enhance temporal reasoning (Xiong et al., 2024; Fei et al., 2024). As illustrated in Fig. 1, language provides additional cues about intent, ordering, and long-range dependencies, thereby offering discriminative semantics to resolve visually ambiguous cases.

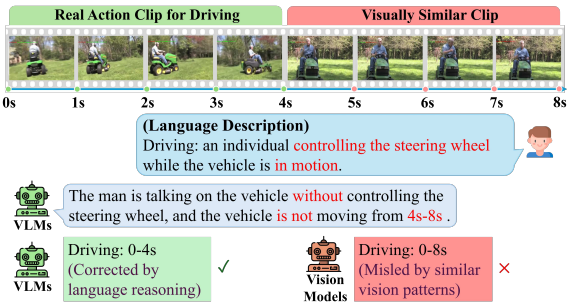


Figure 1: Visually similar actions are hard to distinguish by pixels alone, but language description provides complementary semantics that resolve the ambiguity.

Despite these advantages, however, current VLM-based TAL systems do not consistently outperform vision-only baselines (Huang et al., 2024). This paradox arises because VLMs frequently exhibit a *modality bias*: instead of leveraging vision and language in a balanced manner, the inference is dominated by linguistic priors, causing visual evidence to be underutilized (Cadene et al., 2019; Schrodi et al., 2024; Chen et al., 2020).

The emergence of this bias can be traced to several factors. (i) The *attention sink* phenomenon, in which a subset of tokens or spatial locations consistently monopolizes attention mass regardless of task relevance (Xiao et al., 2023). It is particularly harmful for fine-grained vision tasks, where informative signals are sparse and concentrated in a few vision tokens (Li et al., 2023). When those crucial tokens are suppressed, the model effectively becomes visually blind and overly reliant on language. (ii) *Language-centric* task formulations. As

*Yu Guan is the corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11087–11104
July 2-7, 2026 ©2026 Association for Computational Linguistics

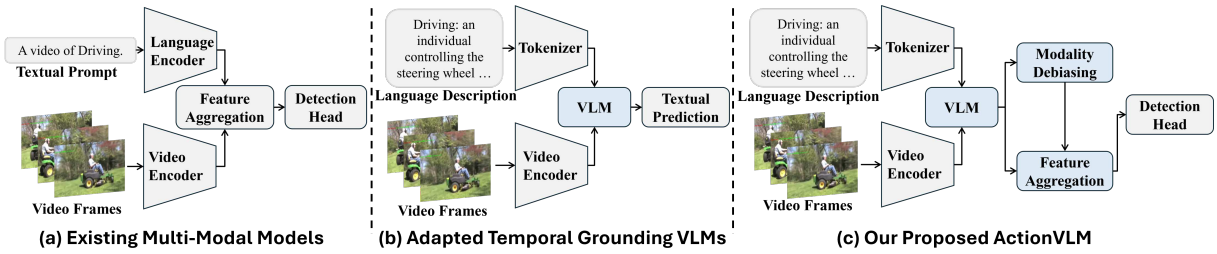


Figure 2: (a) Existing multi-modal TAL models (Liberatori et al., 2024; Lin et al., 2022) distribute uniform weights to vision and language. (b) Language-centric VLMs predict in text. (c) Our proposed ActionVLM mitigates modality bias by adaptively reweighting the language modality and preserving the vision fidelity during aggregation.

shown in Fig. 2(b), VLMs are instructed and outputted in text formats, which encourages reliance on linguistic priors (Xiao et al., 2024). When visual evidence is limited or dataset correlations induce strong linguistic priors, models tend to default to language, producing systematic errors and unwarranted overconfidence (Cadene et al., 2019).

Motivated by Kang et al. (2025) that redistribute attention weights to relieve sinks, we introduce a debiasing module that explicitly estimates the *language advantage*—the incremental benefit of language over vision-only predictions—and adaptively reweights linguistic cues during fusion. Unlike prior debiasing approaches (Cadene et al., 2019) that add extra branches and double computation, our design incurs negligible cost while ensuring language contributes only when it offers measurable gains, thus preventing attention sinks from suppressing critical visual evidence. In parallel, inspired by studies highlighting the risks of language-centric task formulations (Xiao et al., 2024), we reformulate TAL with a residual feature aggregation strategy, where language is treated as a refinement signal rather than the primary driver. This vision-centric design maintains visual features as the backbone of temporal reasoning while leveraging language only as complementary semantics.

Building on these two insights, we propose **ActionVLM**, a vision-centric framework for TAL that integrates explicit debiasing and residual aggregation to mitigate modality bias (Fig. 2c). Our contributions are as follows:

- We identify modality bias as a fundamental challenge in applying VLMs to TAL and introduce a lightweight debiasing unit that adaptively reweights linguistic cues to counteract bias.
- Building on this, we propose a residual aggregation strategy that treats language as a complementary refinement signal, thereby enhancing temporal reasoning without compromising vision as the primary modality.
- We conduct extensive evaluations on three standard TAL benchmarks and demonstrate significant improvements over state-of-the-art methods.

2 Related Work

2.1 Temporal Action Localization

Research on Temporal Action Localization (TAL) has traditionally been dominated by vision-only approaches. Early methods relied on pre-extracted features from pretrained vision backbones (Carreira and Zisserman, 2017; Chen et al., 2024b), while recent approaches move toward end-to-end learning directly from raw videos (Liu et al., 2024; Yang et al., 2023), narrowing the gap between pretraining and downstream objectives. These features are typically fed into detection heads such as ActionFormer (Zhang et al., 2022), which classify action presence at each time step and regress start-end boundaries. Despite this progress, vision-only models remain constrained: actions with similar appearances are easily confused, resulting in inaccurate boundaries.

A central challenge is modeling temporal causality, namely the cause-effect dependencies across poses or events (Ning et al., 2019), which is essential for action localization (Zhou et al., 2018). Yet it is hindered by vision redundancy, where informative cues are easily diluted by background (Li et al., 2023; Wang et al., 2024a). Consequently, models trained with only coarse visual supervision often overfit to spurious correlations. To address this, recent vision-only works attempt to strengthen temporal modeling. ASL (Shao et al., 2023) highlights discriminative frames via a Gaussian-based weighting scheme, while CausalTAD (Li et al., 2024) introduces causal modules. Nevertheless, their reliance on vision alone limits model's robustness.

Additionally, several TAL models (Liberatori et al., 2024; Lin et al., 2022) have explored lever-

aging CLIP (Radford et al., 2021) to generalize to unseen actions, but this mainly benefits zero-shot capacity rather than localization performance. This is largely due to their fixed modality weights, which leave modality bias unresolved (Fig. 2a).

2.2 Language for Temporal Grounding

Parallel works on temporal grounding demonstrate that language can provide complementary supervision (Lin et al., 2023). Language descriptions supply high-level semantics about action intent, ordering, and boundaries, offering explicit anchors that visual features often lack (Lei et al., 2019). Recent efforts adopt video-language models (VLMs). VTimeLLM (Huang et al., 2024) aligns videos with captions via large-scale pretraining and adapts this alignment to instruction-based event localization. Other studies enhance temporal reasoning with explicit temporal indices (Ren et al., 2024; Chen et al., 2024a) or embeddings (Qian et al., 2024).

2.3 Modality Bias

Increasing evidence shows that VLMs often suffer from modality bias (Zheng et al., 2025), where models overfit to linguistic shortcuts and neglect visual cues. This has been widely reported in vision-language tasks such as visual question answering, where removing visual features yields only marginal performance drops (Xiao et al., 2024).

Prior work can be grouped into two directions. Diagnostic works primarily focus on revealing bias. For example, gradient-based analyses (Kwon et al., 2025) and the attention sink studies (Xiao et al., 2023) provide valuable insights into how linguistic shortcuts dominate inference but provide little in terms of mitigation. In contrast, mitigation methods typically rely on either extra branches (Cadene et al., 2019; Clark et al., 2019), which introduce a language-only predictor to discount bias but double computation, or counterfactual data augmentations (Chen et al., 2020), which enforce visual reliance through handcrafted or synthetic mismatched pairs. Although effective, both are costly or unscalable, limiting large-scale video understanding.

3 ActionVLM

3.1 Problem Definition and Motivation

Given an untrimmed video with L frames and C predefined action categories, Temporal Action Localization (TAL) aims to produce frame-wise action proposals consisting of (1) a classification score sequence $\hat{\varphi}_{cls} \in \mathbb{R}^{L\times C}$, and (2) a boundary localization sequence $\hat{\varphi}_{loc} \in \mathbb{R}^{L\times 2}$, where each tuple specifies the start and end timestamps. A vision encoder E_{vis} and a mean pooling layer map raw frames F to per-frame vison features $F_{vis} = E_{vis}(F) \in \mathbb{R}^{L\times D}$, where D is the feature dimension.

Recent works employ VLMs to inject language semantics, but still there is a gap between those models and strong vision-only baselines (Huang et al., 2024). Formally, language models M_{lang} are instructed and output in text, formulated as $(\hat{\varphi}_{cls}, \hat{\varphi}_{loc}) = M_{lang}(F_{vis}, T_{ins})$, where T_{ins} denotes a task-specific instruction prompt, and the proposals $\hat{\varphi}_{cls}$ and $\hat{\varphi}_{loc}$ are embedded into a textual template. This formulation shifts the learning burden onto the language component and encourages the system to explain visual inputs via linguistic shortcuts (Xiao et al., 2024; Cadene et al., 2019). Consequently, visual signals can be down-weighted due to the attention sink (Xiao et al., 2023; Li et al., 2023), which is especially detrimental to fine-grained temporal localization.

Instead of projecting actions into the linguistic embedding space, we treat the VLM as a semantic enhancer that injects language priors into visual representations. Fig. 3 illustrates the overview of ActionVLM. To generate language features F_{lang}, we employ a generation module that distills semantics from action descriptions T_{desc}, defined as $F_{lang} = M_{lang}(F_{vis}, T_{ins}, T_{desc})$. However, directly incorporating F_{lang} risks amplifies modality bias toward language (Zheng et al., 2025). Thus, we introduce a debiasing mechanism, which estimates the language advantage A_{lang}, quantifying the additional information gained from language. Only when $A_{lang} > 0$, indicating complementary semantics rather than redundant bias, are the language features integrated into the visual stream. The fusion is performed by an aggregation module M_{agg}, formulated as $(F_{cls}, F_{loc}) = M_{agg}(F_{vis}, F_{lang}, A_{lang})$, where the refined representations F_{cls} and F_{loc} are passed to a standard detection head M_{dh} (Zhang et al., 2022) to generate the action proposals: $(\hat{\varphi}_{cls}, \hat{\varphi}_{loc}) = M_{dh}(F_{cls}, F_{loc})$.

3.2 Language Feature Generation

Language feature generation leverages VLMs to inject semantic knowledge from action descriptions T_{desc}. Unlike weak visual supervision without annotating the temporal causality, action descriptions can explicitly encode cause-effect relations across ordered poses. As shown in Fig. 1, those linguis-

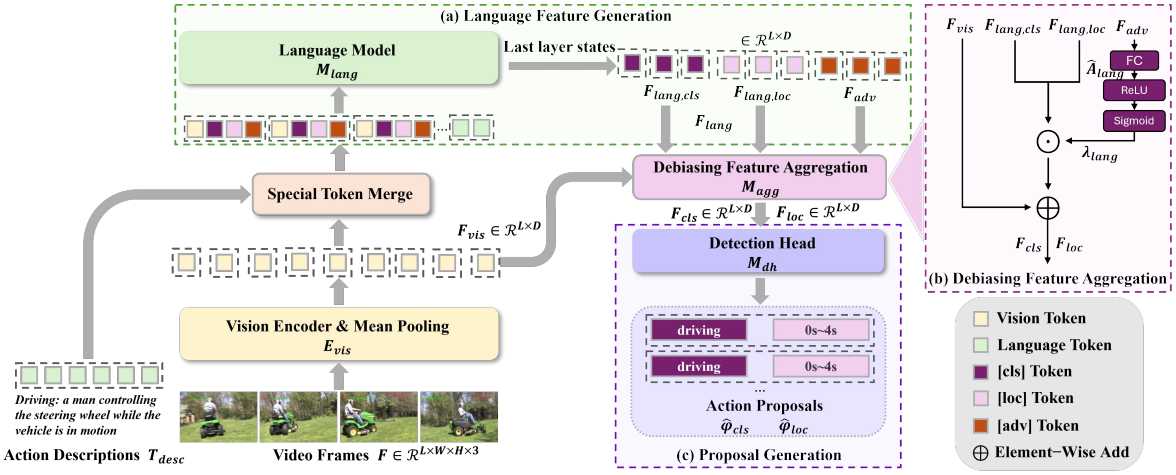


Figure 3: The overview of our ActionVLM framework. **(a) Language Feature Generation:** employ the language model to distill rich contextual information from action descriptions. **(b) Debiasing Feature Aggregation:** scale the language features by estimating the contribution, then refine the vision features with reweighted language features. **(c) Proposal Generation:** predict action proposals from the aggregated features.

tic supervisions provide higher-level guidance that helps disambiguate visually similar actions. In practice, such descriptions can be easily obtained either from human experts or intelligent models. We adopt GPT-4o (OpenAI, 2024) to generate action descriptions, prompting it to specify representative start and end poses for each action without access to visual inputs, detailed in Appx. B.1.

Since VLMs process in discrete tokens while downstream tasks require continuous representations, we follow Yuan et al. (2025) and extract the last-layer hidden states of special tokens to construct continuous features. Specifically, three special tokens are introduced, namely [cls], [loc] and [adv]. [cls] and [loc] summarize information for action classification and localization, respectively, while [adv] estimates the contribution of the language modality. As illustrated in Fig. 3(a), each video frame is represented by its visual feature $F_{vis}^{(l)}$ and three special tokens, with the action description tokens T_{desc} appended to the end of the sequence.

After VLM processing, contextual language priors and critical visual cues are propagated throughout the sequence. The special tokens are trained to summarize target-specific information under this enriched context. Their final hidden states are extracted as continuous features, namely $F_{lang,cls}, F_{lang,loc}, F_{adv} \in \mathbb{R}^{L\times D}$, which serve as adaptable inputs for downstream modules.

3.3 Language Advantage

Language can benefit TAL by providing high-level semantics difficult to infer from visual evidence alone. However, because TAL is fundamentally a vision-centric task, language can also become a shortcut: the model may reduce the training objective by exploiting frequent textual patterns rather than grounding its prediction in the video content. As a result, the multimodal loss alone is not a reliable signal of whether language is truly helpful.

To make this distinction explicit, we compare the task loss of a vision-only branch, denoted as $\mathcal{L}_v$, with that of the corresponding vision-language branch, denoted as $\mathcal{L}_{vl}$. Here, $\mathcal{L}_v$ reflects how well the task can be solved from visual evidence alone, while $\mathcal{L}_{vl}$ measures the loss after language is incorporated. Based on this comparison, we define Language Advantage (LA) as $A_{lang} = \mathcal{L}_v - \mathcal{L}_{vl}$.

This quantity measures the utility of language relative to vision. When language provides complementary semantics and improves prediction beyond the visual baseline, we have $\mathcal{L}_{vl} < \mathcal{L}_v$ and thus $A_{lang} > 0$. When language introduces misleading priors or overrides visual evidence, $\mathcal{L}_{vl}$ can even exceed $\mathcal{L}_v$, yielding a small or negative A_{lang} and indicating harmful modality bias.

This definition also explains why directly optimizing the multimodal branch can be problematic. In conventional VLMs, the model is trained only through $\mathcal{L}_{vl}$, so any shortcut-induced loss reduction would create a positive feedback loop that further increases reliance on language. In contrast, LA uses the vision-only branch as a stable reference, allowing us to determine whether language is providing genuine complementary information or merely acting as a shortcut.

Algorithm 1: Training Pipeline

Input: Dataset $\mathcal{D}$; number of frames L;number of classes C; number of samples in class c: $N^{(c)}$; ground truth class φ_{cls}

 # Vision-Only Epoch

Initialize class-wise loss $\bar{\mathcal{L}}_v^{(c)} \leftarrow 0$ for all actions;
foreach $(F_{vis}, \varphi_{cls}, \varphi_{loc}) \sim \mathcal{D}$ **do**
  Compute task loss $\mathcal{L}_v$ **on vision features** F_{vis};
  Accumulate mean loss per class: $\bar{\mathcal{L}}_v^{(\varphi_{cls})} += \frac{\mathcal{L}_v}{N^{(\varphi_{cls})}}$;
  Backpropagate vision-only task loss: $\mathcal{L}_v$;
end

 # Vision-Language Epoch

foreach $(F_{vis}, F_{lang}, F_{adv}, \varphi_{cls}, \varphi_{loc}) \sim \mathcal{D}$ **do**
  Estimate advantage by Eq. 2: $\hat{A}_{lang} = FC(F_{adv})$;
  Aggregate F_{vis} and F_{lang} by Eq. 4 and 5 to obtain vision-language features F_{cls} and F_{loc};
  Compute task loss $\mathcal{L}_{vl}$ and text generation loss $\mathcal{L}_{tg}$ **on vision-language features** F_{cls} and F_{loc} by Eq. 6 and 8;
  Estimate target advantage by Eq. 1: $A_{lang}^{(l)} = \bar{\mathcal{L}}_v^{(\varphi_{loc})} - \mathcal{L}_{vl}^{(l)}$;
  Compute advantage regression by Eq. 3: $\mathcal{L}_{adv} = \frac{1}{L}\sum_{l=1}^{L}(\hat{A}_{lang}^{(l)} - A_{lang}^{(l)})^2$;
  Backpropagate total loss by Eq. 9: $\mathcal{L} = \mathcal{L}_{vl} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{tg}\mathcal{L}_{tg}$;
end

Alternating-Epoch Estimation. A naive implementation would require an additional vision-only branch to compute $\mathcal{L}_v$ at every iteration. To avoid this overhead, we adopt a lightweight alternating-epoch estimation inspired by reinforcement learning (Schulman et al., 2015). Training alternates between vision-only and vision-language epochs. Vision-only epochs compute class-wise average losses $\bar{\mathcal{L}}_v^{(c)}$, which then serve as a stable proxy for $\mathcal{L}_v$ during vision-language epochs. The language advantage for the l^{th} frame is thus estimated as

$$A_{lang}^{(l)} = \bar{\mathcal{L}}_v^{(c)} - \mathcal{L}_{vl}^{(l)}, l \in \{1, 2, ..., L\}, \quad (1)$$

where L is the number of frames, and c the ground-truth action category of that frame.

To predict LA at inference, the learnable [adv] tokens are trained to regress the approximated advantage from their hidden states F_{adv} by

$$\hat{A}_{lang} = FC(F_{adv}), \quad (2)$$

where FC denotes a fully connected projection. We then define the advantage loss as

$$\mathcal{L}_{adv} = \frac{1}{L}\sum_{l=1}^{L}(\hat{A}_{lang}^{(l)} - A_{lang}^{(l)})^2 \quad (3)$$

Algorithm 1 summarizes our alternating-epoch estimation of LA. During each vision-only epoch, we compute the task loss using only visual features and accumulate the class-wise average loss $\bar{\mathcal{L}}_v^{(c)}$. In the subsequent vision-language epoch, this quantity serves as a lightweight proxy for the vision-only loss $\mathcal{L}_v$. In this way, LA is anchored to a vision-only baseline without requiring an additional vision branch at every iteration.

This estimation is computationally efficient for three reasons. First, the two training phases share the same backbone and differ only in whether the language model is activated, avoiding architectural duplication. Second, vision-only epochs skip the language model entirely and are therefore cheaper than standard multimodal updates, while still providing a principled reference for calibrating modality contributions. Third, because LA is derived directly from task losses rather than dataset-level bias statistics, it is naturally model- and dataset-agnostic, making the same estimation strategy applicable across different TAL benchmarks and VLM backbones.

3.4 Debiasing Feature Aggregation

To dynamically regulate the contribution of language, we derive a reliability weight λ_{lang} from the predicted LA $\hat{A}_{lang}$, using it as a utility signal to determine whether language provides information beyond vision and should therefore participate in vision-grounded localization. To avoid assigning positive weights to negative advantages, we clip it and scale it into $[0, 1)$ through sigmoid mapping:

$$\lambda_{lang} = 2\sigma(ReLU(\hat{A}_{lang})) - 1 \quad (4)$$

Rather than relying solely on language features for downstream tasks, we preserve the visual feature F_{vis} and treat language features as a bounded residual refinement. Inspired by residual learning (He et al., 2016), this design keeps vision dominant and allows language to complement it only when beneficial, thereby mitigating modality bias. As illustrated in Fig. 3(b), the final representations for classification and localization are obtained via element-wise addition:

$$\begin{aligned} F_{cls} &= F_{vis} \oplus \lambda_{lang} F_{lang,cls} \\ F_{loc} &= F_{vis} \oplus \lambda_{lang} F_{lang,loc} \end{aligned} \quad (5)$$

3.5 Detection Head

We adopt the Detection Head (DH) of ActionFormer (Zhang et al., 2022), which employs 1D

convolutions to transform the aggregated features (Eq. 5) into action proposals $\hat{\varphi}_{cls}$ and $\hat{\varphi}_{loc}$. The detection head loss is defined as

$$\mathcal{L}_{dh} = \frac{1}{M}\sum_{l=1}^{L}\left(\mathcal{L}_{cls} + \lambda_{loc}\mathbf{1}_{\mathcal{C}_l}\mathcal{L}_{loc}\right), \quad (6)$$

where $\mathcal{L}_{cls}$ is the focal loss (Tian et al., 2019) for classification, $\mathcal{L}_{loc}$ is the DIoU loss (Rezatofighi et al., 2019) for localization, M is the number of proposals, L is the frame length, $\mathbf{1}_{\mathcal{C}_l}$ indicates positive frames, and λ_{loc} balances localization.

3.6 Training Objectives

To further align the language model with the TAL objective, we retain textual action proposal generation as an auxiliary task. It generates a sentence $\hat{T}_{pred}$ listing the ground truth proposals:

$$\hat{T}_{pred} = M_{lang}(F_{vis}, T_{ins}, T_{desc}) \quad (7)$$

We optimize it using a cross-entropy loss:

$$\mathcal{L}_{tg} = L_{ce}(\hat{T}_{pred}, (\varphi_{cls}, \varphi_{loc})), \quad (8)$$

where the ground truths $(\varphi_{cls}, \varphi_{loc})$ are formatted into a structured textual template.

The overall training objective combines DH loss $\mathcal{L}_{dh}$, the text generation loss $\mathcal{L}_{tg}$ (Eq. 8), and the advantage loss $\mathcal{L}_{adv}$ (Eq. 3):

$$\mathcal{L} = \mathcal{L}_{dh} + \lambda_{tg}\mathcal{L}_{tg} + \lambda_{adv}\mathcal{L}_{adv}, \quad (9)$$

where λ_{tg} and λ_{adv} control the relative weights.

4 Experiment

4.1 Experimental Setting

Datasets. We evaluate our method on THUMOS14 (Idrees et al., 2017) and ActivityNet-1.3 (Caba Heilbron et al., 2015). THUMOS14 and ActivityNet are third-person benchmarks, consisting of 413 and 19,994 videos, respectively. In appendix, we also evaluate on EPIC-Kitchens 100 (Damen et al., 2020), which consists of 700 egocentric videos.

Metrics. We report mean Average Precision (mAP) at various temporal Intersection over Union (tIoU) thresholds, following the common practice (Liu et al., 2019). tIoU is defined as the temporal overlap. At each tIoU threshold, mAP is computed by averaging the precision across actions.

Training. Our training involves three components. The detection head is trained from scratch with a learning rate (lr) of 1e-4. The vision encoder we use is the pre-trained VideoMAEv2 (VM2) (Wang et al., 2023), which is further fine-tuned in our framework with an lr of 5e-6. The language model is fine-tuned using the LoRA (Hu et al., 2022) with a rank of 128, an alpha of 256, and an lr of 5e-6. All training uses DeepSpeed (Rasley et al., 2020) with AdamW on an Nvidia L40 for 120 epochs. The batch size is set to 1 for THUMOS14 and 4 for ActivityNet. Following Zhang et al. (2022), we set the λ_{loc} (Eq. 6) to 1. The auxiliary task weights

			ActivityNet-1.3				THUMOS14					
Model	**Vision**	**Language**	**0.5**	**0.75**	**0.95**	**Avg.**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**Avg.**
pre-extracted												
ASL (Shao et al., 2023)	TSP/I3D	-	54.1	37.4	8.0	36.2	85.3	80.9	73.4	61.1	45.1	69.2
Pred-DETR (Kim et al., 2025b)	I3D/I3D	-	58.4	39.1	9.9	38.6	84.1	80.0	72.2	60.4	45.8	68.5
DiGIT (Kim et al., 2025a)	R(2+1)D/I3D	-	54.4	38.2	**10.7**	37.3	83.6	79.6	71.9	61.5	48.6	69.0
ADI-Diff (Foo et al., 2024)	R(2+1)D/I3D	-	56.9	38.9	9.1	38.3	84.9	81.5	76.5	63.0	48.0	70.8
ActionFormer (Zhang et al., 2022)	TSP/VM2-g	-	55.1	38.3	8.9	37.1	84.0	-	73.0	-	47.7	69.6
MFAM (Tang et al., 2024)	I3D/VM2-g	-	54.7	37.3	8.6	36.6	84.6	80.8	73.5	61.7	48.6	69.8
TriDet (Shi et al., 2023)	TSP/VM2-g	-	54.7	38.0	8.4	36.8	84.8	-	73.3	-	48.8	70.1
DyFADet (Yang et al., 2024a)	TSP/VM2-g	-	58.1	39.6	8.4	38.5	86.0	81.7	76.3	64.5	50.1	71.7
ActionVLM-1.5B (ours)	TSP/VM2-g	QW2.5-1.5B	58.6	39.7	9.3	39.0	87.3	82.6	76.9	64.9	50.7	72.5
ActionVLM-3B (ours)	TSP/VM2-g	QW2.5-3B	**58.8**	**40.0**	9.5	**39.3**	**87.7**	**83.4**	**77.4**	**65.2**	**50.9**	**72.9**
end2end - small vision encoder												
ViT-TAD (Yang et al., 2024b)	VM2-S	-	55.1	37.8	8.8	36.7	79.8	75.2	69.4	56.4	41.7	64.3
Re^{2}TAL (Zhao et al., 2023)	Re^{2}Swin-T	-	55.1	37.1	8.3	36.5	82.1	78.1	60.1	59.8	44.4	66.9
AdaTAD (Liu et al., 2024)	VM-S	-	56.2	39.0	9.1	37.8	84.5	80.2	71.6	60.9	46.9	68.8
ActionVLM-S-1.5B (ours)	VM2-S	QW2.5-1.5B	57.8	39.9	9.4	38.9	86.6	82.3	75.0	63.9	49.0	71.4
ActionVLM-S-3B (ours)	VM2-S	QW2.5-3B	**58.1**	**40.2**	**9.6**	**39.2**	**87.2**	**82.9**	**75.8**	**64.6**	**49.6**	**72.0**
end2end - large vision encoder												
ViT-TAD (Yang et al., 2024b)	VM2-B	-	55.9	38.5	8.8	37.4	85.1	80.9	74.2	61.8	45.4	69.5
Re^{2}TAL (Zhao et al., 2023)	Re^{2}SF-101	-	55.8	38.5	9.4	37.6	84.2	80.3	74.3	62.7	49.6	70.2
AdaTAD (Liu et al., 2024)	VM-S	-	56.8	39.4	9.7	38.4	87.0	82.4	75.3	63.8	49.2	71.5
ActionVLM-B-1.5B (ours)	VM2-B	QW2.5-1.5B	58.4	40.3	9.8	39.4	88.1	84.5	77.4	66.9	51.5	73.7
ActionVLM-B-3B (ours)	VM2-B	QW2.5-3B	**58.8**	**40.6**	**10.2**	**39.7**	**88.8**	**85.1**	**78.2**	**67.1**	**51.8**	**74.2**

Table 1: Results of mAPs (%) at different IoU thresholds on THUMOS14 and ActivityNet-1.3. The first and second best results for each pipeline are highlighted in **bold** and underlined.

Task	Model	THUMOS14								
		0.3	0.4	0.5	0.6	0.7	Avg.	Fixed↓	Infinite↓	LAP↓
QA	VTimeLLM (Huang et al., 2024)	57.4	46.7	33.8	19.7	9.6	33.4	14.6	27.7	14.8
	Timemarker (Chen et al., 2024a)	63.8	55.0	43.6	33.0	18.9	42.9	8.7	11.2	16.5
FA	ActionVLM (only VLM features F_{lang} w/o F_{vis} in Eq. 5)	75.1	70.7	61.4	54.9	34.4	59.3	0	0	8.8
	ActionVLM (only vision features F_{vis} w/o F_{lang} in Eq. 5)	85.0	80.4	73.5	63.1	47.8	69.9	0	0	-
	ActionVLM(full, with the aggregation of F_{lang} and F_{vis})	**87.2**	**82.9**	**75.8**	**64.6**	**49.6**	**72.0**	0	0	**2.6**

Table 2: Quantifying modality bias using Language Advantage on Performance (LAP) and two hallucination rates (%) under two task formulations: Question-Answering (QA) and our Feature Aggregation (FA) approach.



Figure 4: Visualization of modality bias in localizing two challenging actions, *Billiard (left)* and *FrisbeeCatch (right)*. We compare our ActionVLM with a question-answering-based model (TimeMarker) and a vision-only model (AdaTAD). Numbers inside the bars denote the corresponding frame indices.

λ_{tg} and λ_{adv} in Eq. 9 are empirically set to 0.1.

4.2 Comparison with SoTA Methods

Table 1 summarizes results on two standard TAL benchmarks. ActionVLM consistently attains state-of-the-art performance across all training regimes. Among end-to-end models, ActionVLM-B-3B reaches 74.2% mAP on THUMOS14 and 39.7% on ActivityNet-1.3. Compared with strong vision-only baselines built on the same vision backbone, ActionVLM-S surpasses ViT-TAD by 7.7% mAP on THUMOS14 and 2.5% on ActivityNet. Even with frozen pre-extracted features, our method yields a 1.2% gain over DyFADet.

These aggregate numbers indicate that our ActionVLM possesses universality that transcends specific architectures or datasets. The behavior aligns with our analysis in Appx. A.1, where ActionVLM provides clearer benefits when visual evidence is ambiguous, a common challenge in TAL. This suggests that language contributes primarily by reinforcing semantic discrimination, providing stronger semantic cues when uncertain.

To examine robustness to language model capacity, we evaluate two InternVL2.5 (Chen et al., 2024b) variants that use Qwen2.5-1.5B and 3B as their language backbones. InternVL3 is chosen for its strong alignment with visual downstream tasks. ActionVLM-S-3B achieves a 0.6% mAP improvement over the 1.5B counterpart on THUMOS14, suggesting that broader knowledge and richer pretraining better support TAL.

4.3 Analysis on Modality Bias

Measuring Modality Bias. To quantify modality bias at the performance level, we adopt Language Advantage on Performance (LAP), inspired by the diagnostic protocol of Cadene et al. (2019). Unlike the loss-based LA used during training, LAP evaluates how much prediction quality deteriorates when language priors conflict with the visual content. Specifically, we replace the aligned action descriptions with mismatched ones and measure the relative drop in average mAP. To create conflicts, we preferentially swap descriptions between semantically similar actions. When no clear counterpart exists, we randomly shuffle the description-action pairs. A larger LAP indicates that the model is more easily misled by text and therefore more biased toward linguistic cues, whereas a smaller LAP indicates stronger reliance on visual evidence and better robustness to misleading language.

We further examine two extreme hallucination patterns caused by modality bias. 1) Fixed outputs refer to nearly identical temporal segments repeatedly predicted within a sliding video window regardless of the input content, which we detect by checking whether segment intervals remain unchanged across consecutive steps. 2) Infinite listing refers to the model repeatedly emitting previously predicted segments without producing an End-of-Sequence token, which we detect by matching tail segments against earlier predictions. As shown in Table 2, QA-based VLMs exhibit both high LAP and non-trivial hallucination rates, whereas our FA

Model	Easy			Medium			Hard		
	mAP	Gain	mLA	mAP	Gain	mLA	mAP	Gain	mLA
AdaTAD (baseline)	**93.2**	-	-	76.1	-	-	31.1	-	-
Timemarker	71.0	-22.2	-	47.2	-28.9	-	13.7	-17.4	-
ActionVLM (learnable λ_{lang} w/o $\mathcal{L}_{adv}$)	86.6	-6.6	0.16	78.5	2.4	0.20	35.4	4.3	0.23
ActionVLM(full, with $\mathcal{L}_{adv}$)	89.3	-3.9	0.08	**78.6**	**2.5**	0.19	**40.1**	**9.0**	0.36

Table 3: Performance breakdown across different visual ambiguity level (mLA denotes the average frame-level LA).

Model	THUMOS14							
	0.3	0.4	0.5	0.6	0.7	Avg.	*Billiard*	*FrisbeeCatch*
ActionVLM (fixed $\lambda_{lang} = 1.0$)	84.5	79.9	72.7	62.0	46.7	69.2	34.8	26.7
ActionVLM (fixed $\lambda_{lang} = 0.8$)	85.0	80.4	73.1	62.5	46.8	69.6	35.9	27.7
ActionVLM (fixed $\lambda_{lang} = 0.6$)	85.6	81.1	73.6	63.2	47.5	70.2	38.4	30.9
ActionVLM (fixed $\lambda_{lang} = 0.4$)	86.6	81.8	74.2	63.6	48.0	70.8	39.1	31.9
ActionVLM (fixed $\lambda_{lang} = 0.2$)	86.2	81.6	74.0	63.5	47.9	70.6	38.2	31.7
ActionVLM (fixed $\lambda_{lang} = 0.0$)	85.0	80.4	73.5	63.1	47.8	69.9	35.3	27.4
ActionVLM (LLM-generated λ_{lang} w/o $\mathcal{L}_{adv}$)	85.8	81.5	73.3	63.0	47.1	70.1	32.0	32.0
ActionVLM (learnable λ_{lang} w/o $\mathcal{L}_{adv}$)	86.1	81.6	74.1	63.7	48.2	70.7	38.2	32.5
ActionVLM(full, with $\mathcal{L}_{adv}$)	**87.2**	**82.9**	**75.8**	**64.6**	**49.6**	**72.0**	**44.5**	**35.7**

Table 4: Comparison of mAPs and the AP of two visually ambiguous actions under ablations of different language modality weights λ_{lang} (Eq. 5) and the language advantage loss $\mathcal{L}_{\text{adv}}$ (Eq. 3).

formulation reduces LAP to 2.6 and eliminates both failure modes, indicating substantially stronger visual grounding.

Question-Answering (QA) Formulation. As multimodal TAL has rarely been studied beyond zero-shot settings (Liberatori et al., 2024; Lin et al., 2022), systematic analysis of modality bias in TAL is lacking. To this end, we adapt two recent temporal grounding VLMs—VTimeLLM (Huang et al., 2024) and Timemarker (Chen et al., 2024a)—to the TAL setting using a QA formulation.

Our analysis reveals a fundamental limitation of this paradigm. As illustrated in Fig. 4, Timemarker often produces implausible but linguistically preferred boundaries on challenging videos (e.g., coarse rounded timestamps or obviously invalid segments), revealing a reliance on common numerical patterns rather than frame-level evidence.

Quantitatively, Table 2 shows that QA-based VLMs exhibit substantial degradation with up to 16.5% LAP, suggesting a strong modality bias. Such sensitivity confirms that QA formulations, which generate ranked textual proposals, inherently overemphasize linguistic patterns and weaken visual grounding, making them ill-suited for TAL.

Feature Aggregation (FA) Formulation. Instead of casting TAL as a text-generation problem, our Feature Aggregation (FA) formulation avoids LLM explicitly predicting boundaries in language space, and fundamentally removes the structural source of hallucinations observed in QA-based models (Table 2).

However, Table 2 reveals that directly using VLM-derived language features F_{lang} still underperforms the vision-only variant by 10.6% mAP. This gap highlights that features from language-centric VLMs remain insufficient for dense temporal grounding due to the inherent modality bias.

Our full ActionVLM resolves this limitation through our debiasing FA. By explicitly restoring vision as the dominant signal and using language only as a complementary prior, ActionVLM achieves the best overall performance while reducing LAP to 2.6%. As illustrated in Fig. 4, language priors attend to semantically relevant cues (e.g., subtle arm motions in *Billiards*), enabling more precise localization than vision-only AdaTAD, which relies primarily on global visual saliency. This demonstrates that FA leverages language to refine, rather than override, visual grounding.

4.4 Analysis on Language Advantage

To analyze how Language Advantage (LA) affects performance, we group action categories in THUMOS14 into easy, medium, and hard subsets based on the performance of the vision-only baseline (detailed in Appx. A.9). The hard subset includes *Billiard* and *FrisbeeCatch*. While multiple factors may contribute to their lower performance, both actions exhibit notable visual ambiguity: *Billiard* exhibits subtle and sparse motion cues, while *FrisbeeCatch* is easily confused with throwing actions and differs only in temporal pose order (Fig. 4).

Table 3 reveals three key observations. First, higher visual ambiguity is associated with stronger modality bias. When visual cues are limited, mod-

els lack sufficient visual evidence and consequently tend to linguistic priors from pretraining, leading to pronounced degradation on vision tasks.

Second, ActionVLM alleviates modality bias via LA-driven adaptive weighting. For hard cases, ActionVLM assigns higher mLA, allowing language to provide complementary semantics when vision struggles. The strong positive correlation between LA and performance gains in Table 3 confirms that LA reliably identifies bias-sensitive cases and activates language selectively rather than uniformly.

Third, the advantage loss $\mathcal{L}_{adv}$ (Eq. 3) is essential for suppressing redundant language cues. Without $\mathcal{L}_{adv}$, the model assigns non-trivial language weights to easy actions (mLA=0.16), where vision is already sufficient, resulting in performance drops (-6.6 mAP vs. AdaTAD). The full model learns pseudo-optimal modality weights, reducing mLA to 0.08 on easy actions while amplifying language contributions for ambiguous ones, which explains why performance gains concentrate on hard cases.

Table 4 further compares ActionVLM under different settings of the language weight. With fixed or globally learnable weights, models yield only marginal improvements and can even degrade performance when language is overemphasized—confirming that language provides complementary cues in vision tasks while excessive reliance introduces bias. This also highlights that language is beneficial only when adaptively invoked, and effective modality debiasing in TAL requires sample-specific weighting rather than static fusion.

5 Conclusion

In this study, we present ActionVLM, a vision-centric VLM for Temporal Action Localization (TAL) that explicitly addresses modality bias. Rather than allowing language to dominate prediction through shortcut correlations, our framework preserves visual grounding and invokes language only when it provides measurable benefit, through language advantage estimation and debiased residual aggregation. In this way, ActionVLM enables language to refine rather than override visual grounding. Extensive experiments show that this design not only alleviates modality bias and improves robustness under visual ambiguity, but also yields consistent gains across TAL benchmarks with negligible overhead. More broadly, our results suggest a practical multimodal alternative to purely visual scaling, pointing toward robust video understanding by better calibrating language priors rather than relying solely on ever larger visual data and models.

Limitations

While ActionVLM demonstrates strong performance, it has several limitations. First, incorporating vision-language models increases memory usage. Our small-sized variant, however, can be trained end-to-end on 24 GB GPUs, which covers most widely available hardware. Future work may explore lighter adaptations to further reduce resource costs. Second, following prior works (Zhang et al., 2022), we adopt a sliding-window dataloader with a fixed window size, which may introduce label noise by treating partial clips as complete actions. More adaptive loading strategies could improve robustness. Third, language descriptions may introduce inherent bias due to imperfect alignment with fine-grained visual cues. However, our debiasing feature aggregation could dynamically regulate language contributions, mitigating such effects in practice, as evidenced by the reduced sensitivity to misleading descriptions (Table 2). Fully addressing language bias remains an open direction.

Ethics Statement

This work builds upon publicly available video datasets and does not involve private or personally identifiable data. However, vision-language models may inherit biases from pre-training corpora and textual supervision, which can propagate into temporal action localization. If misused in surveillance or monitoring settings, action localization systems could enable harmful profiling or intrusive behavioral analysis. In addition, biased descriptions or uneven data coverage may affect some activities, populations, or environments disproportionately when such systems are deployed in the real world. To mitigate these risks, we recommend dataset auditing, bias evaluation across activity and environment groups, and uncertainty reporting before any deployment. Our method is intended for academic research on robust and fair video understanding rather than high-stakes decision-making.

References

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. 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.

Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. 2019. Rubi: Reducing unimodal biases for visual question answering. *Advances in neural information processing systems*, 32.

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

Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. 2020. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10800–10809.

Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. 2024a. Timemarker: A versatile video-llm for long and short video understanding with superior temporal localization ability. *arXiv preprint arXiv:2411.18211*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. *arXiv preprint arXiv:1909.03683*.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. 2020. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141.

Hao Fei, Shengqiong Wu, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. 2024. Vitron: A unified pixel-level vision llm for understanding, generating, segmenting, editing. *arXiv preprint arXiv:2412.19806*.

Lin Geng Foo, Tianjiao Li, Hossein Rahmani, and Jun Liu. 2024. Action detection via an image diffusion process. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18351–18361.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. 2024. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14271–14280.

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

Pu Jian, Donglei Yu, Wen Yang, Shuo Ren, and Jiajun Zhang. 2025. Teaching vision-language models to ask: Resolving ambiguity in visual questions. *arXiv preprint arXiv:2507.13773*.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. 2025. See what you are told: Visual attention sink in large multimodal models. *arXiv preprint arXiv:2503.03321*.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Ho-Joong Kim, Yearang Lee, Jung-Ho Hong, and Seong-Whan Lee. 2025a. Digit: Multi-dilated gated encoder and central-adjacent region integrated decoder for temporal action detection transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24286–24296.

Jihwan Kim, Miso Lee, Cheol-Ho Cho, Jihyun Lee, and Jae-Pil Heo. 2025b. Prediction-feedback detr for temporal action detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4266–4274.

JuneHyoung Kwon, MiHyeon Kim, Eunju Lee, Juhwan Choi, and YoungBin Kim. 2025. See-saw modality balance: See gradient, and sew impaired vision-language balance to mitigate dominant modality bias. *arXiv preprint arXiv:2503.13834*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2019. Tvqa+: Spatio-temporal grounding for video question answering. *arXiv preprint arXiv:1904.11574*.

Wenbin Li, Di Yao, Chang Gong, Xiaokai Chu, Quanliang Jing, Xiaolei Zhou, Yuxuan Zhang, Yunxia Fan, and Jingping Bi. 2024. Causaltad: Causal implicit generative model for debiased online trajectory anomaly detection. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 4477–4490. IEEE.

Yicong Li, Xun Yang, An Zhang, Chun Feng, Xiang Wang, and Tat-Seng Chua. 2023. Redundancy-aware transformer for video question answering. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 3172–3180.

Benedetta Liberatori, Alessandro Conti, Paolo Rota, Yiming Wang, and Elisa Ricci. 2024. Test-time zero-shot temporal action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18720–18729.

Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. 2023. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2794–2804.

Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard De Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. 2022. Frozen clip models are efficient video learners. In *European Conference on Computer Vision*, pages 388–404. Springer.

Daochang Liu, Tingting Jiang, and Yizhou Wang. 2019. Completeness modeling and context separation for weakly supervised temporal action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1298–1307.

Shuming Liu, Chen-Lin Zhang, Chen Zhao, and Bernard Ghanem. 2024. End-to-end temporal action detection with 1b parameters across 1000 frames. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18591–18601.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2019. Joint reasoning for temporal and causal relations. *arXiv preprint arXiv:1906.04941*.

OpenAI. 2024. Gpt-4o technical report.

Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. 2024. Momentor: Advancing video large language model with fine-grained temporal reasoning. *arXiv preprint arXiv:2402.11435*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 2024. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. 2019. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666.

Simon Schrodi, David T Hoffmann, Max Argus, Volker Fischer, and Thomas Brox. 2024. Two effects, one trigger: On the modality gap, object bias, and information imbalance in contrastive vision-language representation learning. *arXiv preprint arXiv:2404.07983*.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Jiayi Shao, Xiaohan Wang, Ruijie Quan, Junjun Zheng, Jiang Yang, and Yi Yang. 2023. Action sensitivity learning for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13457–13469.

Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. 2023. Tridet: Temporal action detection with relative boundary modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18857–18866.

Yepeng Tang, Weining Wang, Chunjie Zhang, Jing Liu, and Yao Zhao. 2024. Learnable feature augmentation framework for temporal action localization. *IEEE Transactions on Image Processing*, 33:4002–4015.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. 2023. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14549–14560.

Xiao Wang, Qingyi Si, Jianlong Wu, Shiyu Zhu, Li Cao, and Liqiang Nie. 2024a. Retake: Reducing temporal and knowledge redundancy for long video understanding. *arXiv preprint arXiv:2412.20504*.

Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. 2024b. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Tao Wu, Shuqiu Ge, Jie Qin, Gangshan Wu, and Limin Wang. 2024. Open-vocabulary spatio-temporal action detection. *arXiv preprint arXiv:2405.10832*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. 2024. Can i trust your answer? visually grounded video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13204–13214.

Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. 2024. Large language models can learn temporal reasoning. *arXiv preprint arXiv:2401.06853*.

Le Yang, Ziwei Zheng, Yizeng Han, Hao Cheng, Shiji Song, Gao Huang, and Fan Li. 2024a. Dyfadet: Dynamic feature aggregation for temporal action detection. In *European Conference on Computer Vision*, pages 305–322. Springer.

Min Yang, Guo Chen, Yin-Dong Zheng, Tong Lu, and Limin Wang. 2023. Basictad: an astounding rgb-only baseline for temporal action detection. *Computer Vision and Image Understanding*, 232:103692.

Min Yang, Huan Gao, Ping Guo, and Limin Wang. 2024b. Adapting short-term transformers for action detection in untrimmed videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18570–18579.

Guanghui Ye, Huan Zhao, Yixian Shen, Jiaqi Li, Fengnan Li, Zhihua Jiang, and Keqin Li. 2026. Generating multi-modal knowledge clues as an image: Towards improving image-sequence reasoning with assisted visual input. *IEEE Transactions on Circuits and Systems for Video Technology*.

Guanghui Ye, Huan Zhao, Zhixue Zhao, Xupeng Zha, Yang Liu, and Zhihua Jiang. 2025. Knowledge image matters: Improving knowledge-based visual reasoning with multi-image large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21883–21896.

Haobo Yuan, Xiangtai Li, Tao Zhang, Zilong Huang, Shilin Xu, Shunping Ji, Yunhai Tong, Lu Qi, Jiashi Feng, and Ming-Hsuan Yang. 2025. Sa2va: Marrying sam2 with llava for dense grounded understanding of images and videos. *arXiv preprint arXiv:2501.04001*.

Chen-Lin Zhang, Jianxin Wu, and Yin Li. 2022. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*, pages 492–510. Springer.

Chen Zhao, Shuming Liu, Karttikeya Mangalam, and Bernard Ghanem. 2023. Re2tal: Rewiring pretrained video backbones for reversible temporal action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10637–10647.

Xu Zheng, Chenfei Liao, Yuqian Fu, Kaiyu Lei, Yuanhuiyi Lyu, Lutao Jiang, Bin Ren, Jialei Chen, Jiawen Wang, Chengxin Li, et al. 2025. Mllms are deeply affected by modality bias. *arXiv preprint arXiv:2505.18657*.

Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. 2018. Temporal relational reasoning in videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 803–818.

A Supplementary Material

A.1 Analysis on Visual Ambiguity

Quantitative Analysis. To evaluate whether ActionVLM can disambiguate visually similar actions, we design a dedicated benchmark on THUMOS14. Since existing datasets rarely capture such ambiguity, we manually collect 40 challenging ambiguous clips—two for each of the 20 categories—where subjects perform incomplete or misleading poses. Each full video is processed by the model, while evaluation is restricted to these ambiguous segments. As multiple temporal proposals are generated per video (Sec. 3.5), we retain only the highest-confidence one for clarity.

In ambiguous settings, a reliable model should reflect uncertainty—either by reducing its prediction confidence or by refining its temporal boundaries. We therefore introduce two complementary metrics: the mean predicted confidence ($mConf$) and the mean normalized span length ($mLen$), which jointly measure how cautiously and precisely the model localizes actions. Furthermore, we report accuracies at different $mLen$ thresholds, defined as the proportion of predictions with span lengths below the threshold.

As shown in Table 5, incorporating language priors consistently lowers both $mConf$ and $mLen$, indicating better uncertainty calibration and temporal reasoning. Compared to the vision-only baseline (λ_{lang}=0.0), our learnable modality weight reduces confidence by 1.2% and span length by 8.1%, confirming that adaptive language integration helps the model stay grounded under visual ambiguity.

	THUMOS14				
λ_{lang}	$mConf\downarrow$	$mLen\downarrow$	$acc@0.3\uparrow$	$acc@0.5\uparrow$	$acc@0.7\uparrow$
1.0	4.7	52.5	12.5	37.5	67.5
0.8	4.6	52.4	12.5	37.5	70.0
0.6	4.5	51.2	15.0	42.5	72.5
0.4	4.4	50.0	15.0	42.5	75.0
0.2	4.6	51.9	10.0	35.0	67.5
0.0	5.0	56.9	5.0	30.0	57.5
learnable	**3.8**	**48.8**	**15.0**	**45.0**	**80.0**

Table 5: Quantitative results (all in %) on visually similar clips under different language feature weights.

Qualitative Analysis. To further assess ActionVLM's behavior under visual ambiguity, we visualize three distractors and one true instance from the javelin throw and high jump categories. The distractors exhibit strong visual resemblance to the target actions—e.g., a subject holding a javelin without throwing, or running toward the high-jump bar without actually jumping—creating challenging false positives for vision-only models.

As shown in Fig. 5, models without language guidance tend to overfit to low-level motion cues and misclassify incomplete actions as valid ones. In contrast, ActionVLM leverages temporal semantics encoded in language descriptions to infer action completeness, suppressing false positives and tightening temporal spans. This yields notably lower confidence and shorter predictions on distractors, while maintaining strong activation for true actions. In the most challenging case (the second case of high jump), ActionVLM reduces span length by 34% and confidence by 50%, demonstrating its capacity to reason over action intent and completion, rather than surface-level appearance.

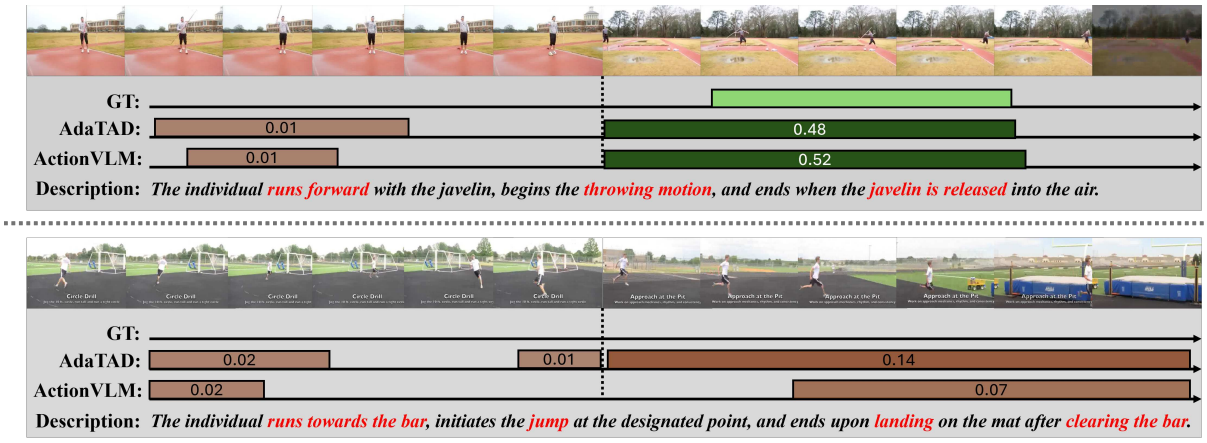


Figure 5: Qualitative visualization under visual ambiguity. From top to bottom: (1) video frames, (2) ground truth (GT), (3) predicted boundaries with confidence (classification score from detection head), and (4) corresponding action descriptions. True positives are shown in green, while visually similar false positives without GT are highlighted in brown. Lower confidence and shorter spans indicate better calibration on ambiguous clips.

	Model	Window Size	Training Latency (ms)	Evaluation Latency (ms)	mAP
(a)	VTimeLLM (Huang et al., 2024)	128	213	928	33.4
	Timemarker (Chen et al., 2024a)	128	229	1079	42.9
(b)	AdaTAD-S (Liu et al., 2024)	768	265	306	68.8
	AdaTAD-B (Liu et al., 2024)	768	601	626	71.5
(c)	**ActionVLM-S-3B (ours)**	768	289	374	72.0
	ActionVLM-B-3B (ours)	768	656	741	74.2

Table 6: Comparison with state-of-the-art methods in terms of efficiency on THUMOS14. (a) Conventional VLMs. (b) Vision-only Baselines. (c) Our proposed ActionVLMs.

			Epic-Kitchens 100					
Model	**Vision**	**E2E**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**Avg.**
Verb Task								
ActionFormer (Zhang et al., 2022)	SlowFast		27.7	26.8	25.6	24.4	20.5	24.9
ActionFormer (Zhang et al., 2022)	VideoMAEv1-L		32.7	31.6	29.1	26.7	23.6	28.7
AdaTAD (Liu et al., 2024)	VideoMAEv1-L	✓	33.1	32.2	30.4	27.5	23.1	29.3
ActionVLM-L-3B (ours)	VideoMAEv1-L	✓	**34.0**	**33.3**	**31.4**	**28.8**	**24.8**	**30.5**
Noun Task								
ActionFormer (Zhang et al., 2022)	SlowFast		25.8	24.7	22.8	20.8	17.5	22.3
ActionFormer (Zhang et al., 2022)	VideoMAEv1-L		31.3	29.7	27.2	25.3	21.3	26.9
AdaTAD (Liu et al., 2024)	VideoMAEv1-L	✓	32.4	31.6	30.1	27.4	24.6	29.3
ActionVLM-L-3B (ours)	VideoMAEv1-L	✓	**32.8**	**32.1**	**30.7**	**28.2**	**26.1**	**30.0**

Table 7: Results of mAPs (%) at different IoU thresholds on Epic-Kitchens 100. Best results are highlighted in **bold**.

A.2 Analysis on Efficiency

Although incorporating VLMs can improve the reliability of reasoning (Ye et al., 2026, 2025), it inevitably incurs additional inference overhead. Table 6 compares ActionVLM with both conventional VLMs and strong vision-only baselines in terms of computational efficiency and accuracy on THUMOS14. We report average training and inference latency per video clip. The window size denotes the number of frames processed in a single temporal chunk, which affects both efficiency and temporal context coverage. We set a smaller window size for conventional VLMs to fit GPU memory, while ActionVLM and AdaTAD use longer windows to capture richer temporal dynamics following the original setting (Liu et al., 2024).

As shown in Table 6, the heavy computation of conventional VLMs severely limits their scalability to downstream tasks. VTimeLLM and Timemarker process short clips with 128 frames yet still exhibit high inference latency due to dense patch tokens and autoregressive decoding. This inefficiency makes direct deployment to long-sequence temporal action localization (TAL) impractical.

To make VLMs viable for video tasks, we introduce two acceleration strategies. First, instead of representing each frame by hundreds of spatial patch tokens, we apply a mean-pooling layer over spatial dimensions, compressing them into a single visual token. This reduces token length by more than one order of magnitude without compromising temporal fidelity, enabling efficient long-sequence modeling. Second, we replace the conventional token-by-token decoding for textual proposal generation (Eq. 8) with a lightweight auxiliary generation scheme. During training, we employ the conventional practice of teacher-forcing strategy (Williams and Zipser, 1989), which enables parallel generation and has little effect on the efficiency, while at inference we restrict the output $\hat{T}_{pred}$ to one token since we rely on feature aggregation while regarding text generation as an auxiliary task. This preserves semantic supervision with negligible computational cost.

As shown in Table 6, these designs substantially improve efficiency. Compared with conventional VLMs, ActionVLM reduces inference latency by more than two times while processing six times longer sequences. When compared with the strong vision-only baseline AdaTAD, ActionVLM introduces only a marginal overhead yet achieves consistent accuracy gains of up to 3.2% mAP. This demonstrates that multimodal reasoning can be achieved efficiently, bridging the gap between language-augmented understanding and practical large-scale TAL deployment.

A.3 Additional Experiments on EPIC-Kitchens

To validate the generalizability of our model on diverse datasets, we further conduct evaluation on

Model	Vision	THUMOS14					
		0.3	0.4	0.5	0.6	0.7	Avg.
AdaTAD (Liu et al., 2024)	VideoMAEv1-B (Tong et al., 2022)	87.0	82.4	75.3	63.8	49.2	71.5
AdaTAD (Liu et al., 2024)	VideoMAEv2-B (Wang et al., 2023)	87.5	83.1	76.0	66.6	51.3	72.8
ActionVLM-3B (ours)	InternViT-300M-V2.5 (Chen et al., 2024b)	72.6	67.5	59.3	48.4	35.4	56.7
ActionVLM-B-3B (ours)	VideoMAEv1-B (Tong et al., 2022)	88.6	84.2	76.1	64.4	49.6	72.6
ActionVLM-B-3B (ours)	InternVideo2-B (Wang et al., 2024b)	**88.8**	84.5	77.3	66.5	51.1	73.6
ActionVLM-B-3B (ours)	VideoMAEv2-B (Wang et al., 2023)	**88.8**	**85.1**	**78.2**	**67.1**	**51.8**	**74.2**

Table 8: Ablation on different vision encoders.

	Model	T_{desc}	[cls]	[loc]	$\mathcal{L}_{tg}$	THUMOS14					
						0.3	0.4	0.5	0.6	0.7	Avg.
(a)	ActionVLM w/o action description T_{desc}		✓	✓	✓	85.9	81.4	74.1	63.8	48.2	70.7
	ActionVLM (full, with priors in description)	✓	✓	✓	✓	**87.2**	**82.9**	**75.8**	**64.6**	**49.6**	**72.0**
(b)	ActionVLM w/o special tokens (extract language feature from vision tokens)	✓			✓	85.8	81.9	74.6	63.5	48.2	70.8
	ActionVLM w/o [loc] tokens (share a single special token)	✓	✓		✓	86.6	82.4	75.2	64.2	49.4	71.6
	ActionVLM (full, with both [cls] and [loc])	✓	✓	✓	✓	**87.2**	**82.9**	**75.8**	**64.6**	**49.6**	**72.0**
(c)	ActionVLM (w/o text generation loss $\mathcal{L}_{tg}$)	✓	✓	✓		86.0	81.9	74.8	64.0	49.3	71.2
	ActionVLM (full, with $\mathcal{L}_{tg}$)	✓	✓	✓	✓	**87.2**	**82.9**	**75.8**	**64.6**	**49.6**	**72.0**

Table 9: Ablation on (a) action descriptions, (b) feature tokens, and (c) text generation loss $\mathcal{L}_{tg}$.

EPIC-Kitchens-100 (Damen et al., 2020). EPIC-Kitchens-100 is a large-scale egocentric benchmark comprising 700 videos (about 100 hours) recorded in 45 kitchens, with fine-grained verb and noun annotations. Its domain specificity and egocentric action patterns differ substantially from the visual and linguistic distributions commonly seen in large-scale VLM pretraining corpora, making high absolute mAP particularly challenging (Liu et al., 2024).

We use GPT-4o-mini to generate short descriptions for each class used in our EPIC-Kitchens-100 evaluation setup. This results in 1,283 description tokens for the 78 verb classes and 2,277 tokens for the 211 noun classes. Importantly, our runtime is dominated by video encoding rather than language modeling, so the text token budget is not the computational bottleneck.

Despite this increased difficulty, ActionVLM-L-3B consistently outperforms AdaTAD across both verb and noun tasks (Table 7), with particularly notable gains on verbs of 1.2% average mAP. This trend is expected: verbs describe actions and motion primitives, whereas language priors provide meaningful semantic constraints that help disambiguate visually subtle or incomplete cues in egocentric footage. In contrast, noun predictions depend primarily on object appearance, where linguistic context contributes less discriminative information, resulting in smaller—but still positive—improvements. These results highlight that ActionVLM not only generalizes beyond standard third-person TAL datasets but also benefits most from scenarios where language provides actionable semantic structure for interpreting complex visual dynamics.

A.4 Additional Ablation on Vision Encoder

Table 8 shows that directly adopting the raw vision encoder from InternVL3 (InternViT-300M-V2.5) (Chen et al., 2024b) yields substantially weaker TAL performance compared with VideoMAE-based (Tong et al., 2022) variants. This degradation is largely attributable to two factors: (i) the InternVL3 vision encoder is not pretrained on video-centric action datasets such as Kinetics (Kay et al., 2017), and (ii) it processes single frames rather than 16-frame snippets, preventing it from modeling motion dynamics and temporal continuity. By contrast, VideoMAE leverages masked video reconstruction over multi-frame inputs, enabling stronger temporal representation learning that better aligns with the requirements of TAL. As a result, replacing the InternVL3 encoder with VideoMAE leads to large gains across all IoU thresholds. Moreover, even under the same vision backbone (VideoMAEv2-B), our ActionVLM-B-3B still surpasses AdaTAD by 1.4% on average mAP, confirming the effectiveness of our overall design.

A.5 Additional Ablation on Language Feature Generation

Action Descriptions. As shown in Table 9(a), removing descriptions leads to a 1.2% mAP drop, confirming their value as auxiliary semantic priors. However, because these descriptions are not visually grounded, mismatches with vision content may

introduce bias. Our proposed reweighting mitigates such mismatches, ensuring language cues remain complementary rather than misleading.

Special Tokens. As shown in Table 9(b), assigning separate tokens for classification and localization disentangles the objectives and enables specialized representations, yielding superior performance compared to shared or vision tokens.

Text Generation. Table 9(c) shows that the auxiliary text generation task strengthens vision-language alignment while providing additional supervision. Removing it weakens the shared semantic space and leads to a 0.7% mAP drop.

A.6 Histogram of Linguistically Preferred Boundaries

As shown in Fig. 4 and discussed in Sec. 4.3, QA-based VLMs exhibit a tendency to predict linguistically preferred temporal boundaries, such as coarse or rounded timestamps. To verify that this behavior is systematic rather than anecdotal, we further analyze the distribution of predicted boundaries on our dataset. Specifically, we construct a histogram of boundary predictions produced by TimeMarker using a fixed window size of 100 frames. For each localization instance, the model receives 100 frames and outputs predicted start and end frame indices, which are aggregated across all samples (Fig. 4 shows the aggregated indices, while Fig. 6 presents the raw ones).

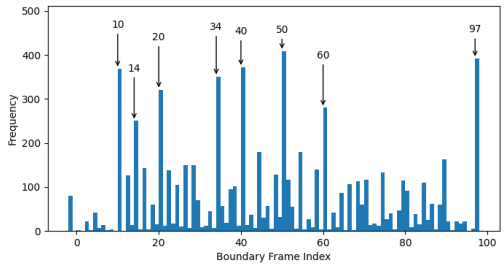


Figure 6: Histogram of frequently predicted temporal boundaries. The eight most frequent frame indices are highlighted with arrows.

As illustrated in Fig. 6, two clear patterns emerge: round-number peaks at multiples of ten (e.g., 10, 40, 60) and dataset-specific peaks aligned with fine-tuning data statistics (e.g., 14, 34, 97). These patterns provide further evidence that the predicted boundaries are influenced by linguistic priors rather than purely visual evidence.

Model	#Token	mAP
w/o description	0	70.7
keywords-only	219	71.0
merged descriptions	659	71.9
ActionVLM-S-3B (full)	742	72.0

Table 10: Sensitivity to language granularity on THUMOS14.

A.7 Model Sensitivity to Language Granularity

Action descriptions improve TAL by providing auxiliary semantic priors, but they also introduce an additional text-token budget. We further ask a practical question: how much description detail is actually needed? In particular, we study whether the full sentence descriptions used in ActionVLM can be replaced by cheaper alternatives without sacrificing accuracy.

We compare two budget-saving variants against the full setting. The first is *keywords-only*, where each class is represented by only three words summarizing the start, middle, and end motions. The second is *merged descriptions*, where semantically similar actions (e.g., LongJump & HighJump) share one coherent sentence. Table 10 reports the results.

While keywords-only descriptions reduce the text budget substantially, they provide only marginal gains over removing descriptions, whereas merged descriptions nearly match the full model. This indicates that preserving coherent action semantics is more important than retaining every class-specific phrase. Since our efficiency analysis in Table 6 shows that our runtime is dominated by video encoding rather than language modeling, we use the full descriptions by default to achieve the best performance.

A.8 Analysis on Performance Gain

In Temporal Action Localization (TAL), even seemingly small improvements in mAP often correspond to substantial underlying advances.

mAP Sensitivity in TAL. TAL is a long-tailed and highly imbalanced task, where many frequent and visually distinctive actions are already well localized by vision-only models. Consequently, mAP—averaged across classes and strict tIoU thresholds—is dominated by these easy cases, making further improvements inherently difficult. As a result, even small increases in overall mAP often correspond to substantial gains on rare or visually ambiguous actions. As shown in Table 3 and Ta-

ble 4, ActionVLM significantly improves AP on visually ambiguous actions, indicating that the observed gains are concentrated on the most challenging cases where modality bias is most harmful.

Annotator Subjectivity and Semantic Alignment. Both THUMOS14 and ActivityNet-1.3 exhibit notable annotator subjectivity, particularly for long or semantically complex actions with ambiguous boundaries. In such cases, annotators implicitly rely on high-level semantic understanding rather than low-level visual cues alone. Language models naturally encode such semantic priors and thus better align with human annotation tendencies. By selectively incorporating language cues through debiasing aggregation, ActionVLM improves robustness to boundary ambiguity without overriding visual evidence, leading to more annotation-consistent predictions.

Backbone Saturation. Recent advances in TAL have largely been driven by increasingly powerful video backbones (Liu et al., 2024; Yang et al., 2024b), which are now approaching performance saturation. With strong temporal encoders already in place, large mAP jumps are no longer realistic. In this regime, consistent improvements across datasets and IoU thresholds signify the correction of errors that cannot be resolved through better visual representations alone. ActionVLM addresses this limitation by mitigating modality bias, offering a complementary axis of improvement beyond backbone scaling.

A.9 Action Categories by Visual Ambiguity

THUMOS14 contains 20 action categories. For the analysis in Sec. 4.4, we group these actions into three subsets based on their visual ambiguity, measured by the performance of a vision-only baseline.

Specifically, actions with high vision-only mAP are considered visually unambiguous, as they can be reliably localized without language cues, while low-performing actions exhibit higher visual ambiguity.

Based on this criterion, we define:

- **Hard**: Billiards, FrisbeeCatch;
- **Easy**: ThrowDiscus, LongJump, HighJump, HammerThrow;
- **Medium**: the remaining actions.

B Prompts

B.1 The Prompts for Action Descriptions Generation

We introduce action descriptions T_{desc} as prior knowledge for localization. These are generated by prompting GPT-4o (OpenAI, 2024) to produce detailed textual depictions of representative start and end poses for each action category, without access to visual input. The prompts used for generation are as follows:

"Describe the action from the start pose to the end pose: e.g. BasketballDunk: The individual starts to jump towards the hoop while holding the basketball and ends when the ball is forcefully pushed through the basket."

The full prompts for each action are recorded in our code. Here we provide the generated results used in our paper as examples:

BaseballPitch: The individual starts by preparing and winding up with the baseball in hand, then initiates the pitching motion, and ends when the ball is released and thrown toward the batter.

Billiards: The individual starts by positioning the cue stick and aiming at the cue ball, then strikes the cue ball with the stick, and ends when the balls come to a complete stop after the shot.

CleanAndJerk: The individual starts by gripping the barbell on the ground, then lifts it to the shoulders in the "clean" phase, pauses briefly, and continues by explosively lifting it overhead in the "jerk" phase, ending when the barbell is held steady overhead with full control.

CliffDiving: The individual starts by standing at the edge of a cliff or platform, then jumps off and performs acrobatic movements while descending, ending when they enter the water below.

CricketBowling: The individual starts by taking a run-up toward the wicket, then releases the ball with a straight arm towards the batsman, ending when the ball reaches the batsman or wicket-keeper.

CricketShot: The individual starts by preparing the bat stance as the ball approaches, then swings the bat to hit the ball, ending when the ball is struck and sent away from the batsman.

Diving: The individual starts by running or standing at the edge of a diving board or platform, then jumps and performs a controlled descent into the water, ending upon water entry.

FrisbeeCatch: The individual starts by tracking the flying frisbee, positions their hands or body to intercept, and ends when they successfully grasp or trap the frisbee.

GolfSwing: The individual begins by addressing the golf ball, then swings the club in a controlled arc to strike the ball, ending when the ball is launched toward the target.

HammerThrow: The individual starts by gripping the hammer, initiates spinning rotations to build momentum, and ends when they release the hammer into the throwing sector.

HighJump: The individual runs towards the bar, initiates the jump at the designated point, and ends upon landing on the mat after clearing the bar.

JavelinThrow: The individual runs forward with the javelin, begins the throwing motion, and ends when the javelin is released into the air.

LongJump: The individual begins running along the track, then takes off from the takeoff board, and ends when they land in the sandpit.

PoleVault: The individual starts by sprinting down the runway holding the pole, plants the pole into the vault box, uses it to propel upward over the bar, and ends when they land safely on the mat.

Shotput: The individual begins by positioning the shot near the neck, then uses a pushing motion to launch the shot forward, ending when the shot lands on the ground.

SoccerPenalty: The individual starts by approaching the ball placed at the penalty mark and ends after striking the ball towards the goal.

TennisSwing: The individual starts by preparing their stance and tracking the incoming ball, then swings the racket to strike the ball, ending when the ball is hit and sent back over the net.

ThrowDiscus: The individual starts by gripping the discus, performs a spinning motion to gain momentum, and ends when the discus is released into the air.

VolleyballSpiking: The individual begins by jumping near the net, swings their arm forcefully to hit the ball downward over the net, and ends when the ball crosses into the opponent's court.

B.2 Instruction Prompt for Textual Action Proposal Generation

We employ an auxiliary textual proposal generation task to better align the vision and language modalities of the VLM. The instruction prompt T_{ins} guides the model to perform temporal action localization (TAL) in a text-to-proposal manner, transforming visual observations into ranked textual predictions based on confidence scores. This design encourages the model to express its temporal reasoning explicitly in natural language, facilitating cross-modal consistency between detection and generation.

"This is a video reasoning for the action localization task. You should identify the start and end frames of each action in the given video and output the ranked list of action proposals according to their confidence. The candidate actions include: $\langle T_{\text{desc}} \rangle$*""*