Invert4TVG: A Temporal Video Grounding Framework with Inversion Tasks Preserving Action Understanding Ability

Anonymous authors
Paper under double-blind review

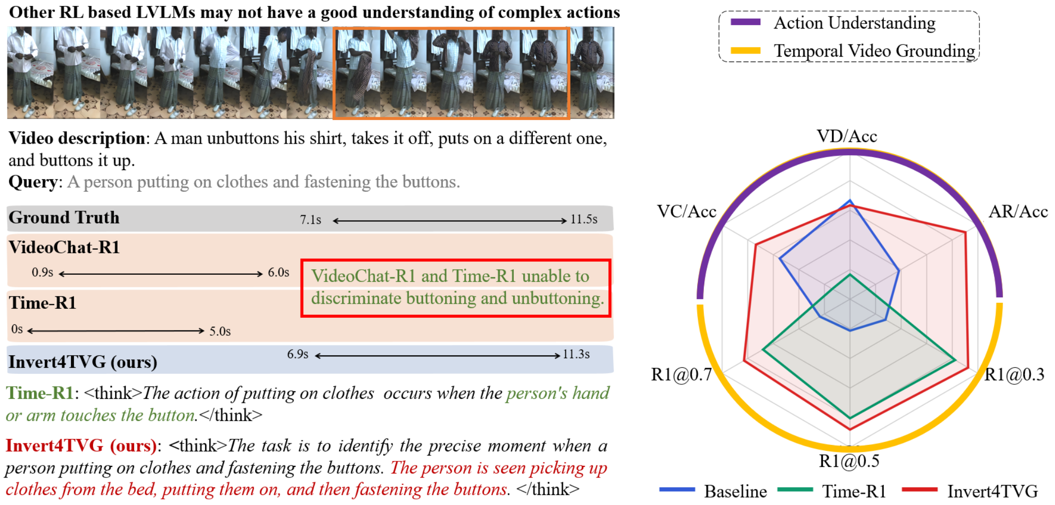


Figure 1: **Left side:** A specific example of temporal video grounding. According to the model's reasoning process,it can be seen that our method achieves better understanding of actions in the video compared to VideoChat-R1 and Time-R1. **Right side:** statistical results demonstrating that Time-R1, which is optimized solely for the IoU loss, reduces action understanding accuracy (where VC, AR, and VD are the proposed three auxiliary inversion TVG tasks measuring multi-granularity action understanding ability). By introducing Inversion-TVG tasks, our method preserves action understanding ability and thus boosts TVG ability (as shown in R1@0.3, R1@0.5, and R1@0.7). Baseline is QWen-2.5-VL-3B.

Abstract

Temporal Video Grounding (TVG) aims to localize video segments corresponding to a given textual query, which often describes human actions. However, we observe that current methods, usually optimizing for high temporal Intersection-over-Union (IoU), frequently struggle to accurately recognize or understand the underlying actions in both the video and query, thus reducing the effectiveness of these methods. To address this, we propose a novel TVG framework that integrates inversion-based TVG as auxiliary objectives to maintain the model's action understanding ability. We introduce three kinds of inversion TVG tasks derived from the original TVG annotations: (1) Verb Completion, predicting masked verbs (actions) in queries given video segments; (2) Action Recognition, identifying query-described actions; and (3) Video Description, generating descriptions containing query-relevant actions given video segments. These inversion tasks are entirely derived from the original TVG tasks and are probabilistically integrated with them within a reinforcement learning framework. By leveraging carefully designed reward functions, the model preserves its ability to understand actions, thereby improving the accuracy of temporal grounding. Experiments show our method outperforms state-of-the-art approaches, achieving a 7.1% improvement in R1@0.7 on Charades-STA for a 3B model.

1 Introduction

Temporal Video Grounding (TVG) is crucial for long-form video understanding (Gaidon et al., 2013; Laptev & Pérez, 2007; Darrell & Pentland, 1993). It localizes a video segment matching a textual query (Gao et al., 2017; Zhang et al., 2023), enabling applications like video-text retrieval (Zhang et al., 2024) and UAV positioning (Ju et al., 2024).

Existing TVG approaches fall into three paradigms: (1) Traditional methods using hand-crafted features, sliding windows, and DETR-like networks (Shi et al., 2022; Gordeev et al., 2024); (2) LVLMs (Bai et al., 2025; Li et al., 2023) that regress segment duration via pretraining; and (3) RL-finetuned LVLMs, such as Time-R1 (Wang et al., 2025a), using Reinforcement Learning (RL) with format rewards for structured reasoning and IoU rewards for alignment.

Although significant progress has been made, we still find wrong grounding results in existing SOTA methods, and most of these wrong cases stem from incorrect action understanding. Figure 1 left shows one case. In the video, a man unbuttons his shirt, takes it off, puts on another one, and then buttons it up. The query is "A person putting on clothes and fastening the buttons", which requires localizing the actions "putting" and "fastening". Both VideoChat-R1 and Time-R1 notice a hand touching a button and localize the action as "buttoning" rather than "unbuttoning", indicating that they seem to focus only on the button itself without distinguishing between buttoning and unbuttoning. We conjecture that these wrong groundings occur because video grounding models are generally optimized only for IoU. Although IoU is improved, this comes at the cost of reduced action understanding capability, which in turn limits their overall video grounding performance. Figure 1 right demonstrates this statistically. Time-R1 (Wang et al., 2025a), optimized for IoU, shows improvement on TVG metrics (e.g., R1@0.3/R1@0.5/R1@0.7) compared to the baseline Qwen2.5-VL-3B. However, it exhibits degradation in action understanding tasks (VC/VD/AR), which ultimately hampers its TVG accuracy.

A key insight of this work is that training a TVG model effectively requires jointly training auxiliary tasks to preserve the model's action understanding capability. A naive approach to achieving this would be to train the TVG task alongside general action understanding tasks such as action recognition/detection/classification. However, these general tasks are not specifically designed for the temporal video grounding objective, and the understanding learned from them may not align well with the precise temporal localization required in TVG.

Unlike general action understanding tasks, we design action understanding tasks that are specifically tailored for the temporal video grounding task. Specifically, by inverting the input and output of the original TVG task, we convert the localization task into understanding task, obtaining a set of Invert-TVG tasks. A key advantage of these Invert-TVG tasks, compared to general action understanding tasks, is that they share the same training data as the original TVG task. On the same video-query data, our method performs both video localization (via the original TVG task) and action understanding (via the Invert-TVG tasks). This tight coupling enables the learned action understanding to be directly aligned with and supportive of the temporal grounding objective, resulting in more effective and synergistic learning.

Specifically, given a video and a natural language query, the original TVG task predicts the temporal segment duration where the action occurs. Inversely, given a video segment, the proposed Invert-TVG tasks infer the action-related information defined in the query from the given segment. We introduce three Invert-TVG tasks: (1) Verb Completion (VC): mask verbs (actions) in the query and then infer the verbs from video segments. (2) Action Recognition (AR): classify the actions in a given video segment where the ground-truth action is in the query. (3) Video Description (VD): generate descriptions for a given video segment, and the descriptions should contain actions provided in the query.

With the well-defined Invert-TVG tasks, we then propose a reinforcement learning framework that optimizes TVG and Invert-TVG tasks together. However, for large-scale models (e.g., 3B/7B parameters), simultaneously optimizing multiple objectives incurs substantial memory overhead. Moreover, the TVG and Invert-TVG tasks are conflict: the ground-truth video segment that TVG is required to produce is precisely the input to an Invert-TVG task, and the original query that an Invert-TVG task may ask for is exactly the input to the TVG task. To address this, we adopt an alternating optimization strategy, executing TVG and Invert-TVG tasks interleavingly. Besides, since temporal video grounding is our main objective while action understanding is an auxiliary objective, we optimize the TVG task with a higher probability while using a lower probability for the Invert-TVG tasks.

Our contributions include:

- We identify action understanding degradation in TVG from IoU over-optimization, and address the problem via inversion TVG tasks.
- We design three inversion TVG tasks which are self-supervised tasks re-purposing TVG annotations for action understanding, including verb completion, action recognition, and video description.
- We propose a reinforcement learning framework dynamically balancing TVG and Invert-TVG tasks, ensuring robust grounding and understanding.

Temporal Video Grounding. Temporal Video Grounding (TVG) (Gao et al., 2017; Hendricks et al., 2017) localizes specific segments in untrimmed videos based on natural language queries. Recent methods fall into two categories: feature-based and frame-based LVLM approaches. Feature-based methods (Carreira & Zisserman, 2017; Lin et al., 2022) extract video and text features using pretrained encoders, then predict timestamps via multimodal fusion. These rely heavily on feature quality, limiting performance. Frame-based LVLM methods have recently gained traction for their strong generalization capabilities. For instance, NumPro (Wu et al., 2025) introduces a frame-numbering mechanism akin to flipping a manga for efficient temporal grounding, while TimeSuite (Zeng et al., 2024) employs grounded tuning to enhance Large Language Models (LLMs) for long-form video understanding. While methods like these and others (Li et al., 2024; Ren et al., 2024) utilize supervised fine-tuning to generate event sequences, they can still struggle with precise boundary detection on benchmarks like Charades-STA compared to specialized feature-based approaches. To address this, Time-R1 (Wang et al., 2025a) employs reinforcement learning (RL) with IoU rewards, achieving state-of-the-art TVG performance. However, its focus on temporal metrics neglects semantic alignment, constraining long-form video understanding

RL in LVLMs. RL has advanced post-training of LVLMs through Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022; Yu et al., 2024) and Reinforcement Learning with Verifiable Reward (RLVR) (DeepSeek-AI, 2025; Chen et al., 2025). RLHF aligns models with human preferences, improving tasks like image captioning, while RLVR enhances deterministic tasks like visual grounding (Liu et al., 2025). However, RL applications in long-form video tasks remain underexplored due to temporal complexity and semantic challenges. TimeZero (Wang et al., 2025b), Time-R1 (Wang et al., 2025a), VideoChat-R1 (Li et al., 2025) apply RL to TVG but overlooks semantic understanding degradation from IoU-focused rewards. Our Invert4TVG framework addresses this by repurposing TVG data into self-supervised tasks, enhancing action semantic understanding and surpassing traditional RL limitations in video grounding.

2 Method

The TVG task aims to temporally localize video segments within long-form videos based on natural language queries. Given a video V, and a language query q, the goal is to identify the temporal boundaries $\tau = [t_s, t_e]$ of the segment of V that best corresponds to q, where $t_s, t_e \in \mathbb{R}^+$. The formal definition of the TVG task is as follows:

$$\text{TVG}(V, q) \rightarrow \tau. \tag{1}$$

In this work, we introduce Invert4TVG, a framework designed to harness the potential of Large Vision-Language Model (LVLM) for the TVG task using Reinforcement Learning (RL) combined with TVG-inversion tasks. The Invert-TVG task is defined as (where q' denotes query-related content):

$$\text{Invert-TVG}(V, \tau) \rightarrow q' \tag{2}$$

Our approach is fundamentally a reinforcement learning algorithm that fine-tunes LVLMs (specifically the Qwen2.5-VL model series) by integrating both TVG and Invert-TVG tasks. In the following, we first introduce the fundamentals of GRPO (i.e., Group Relative Policy Optimization, a

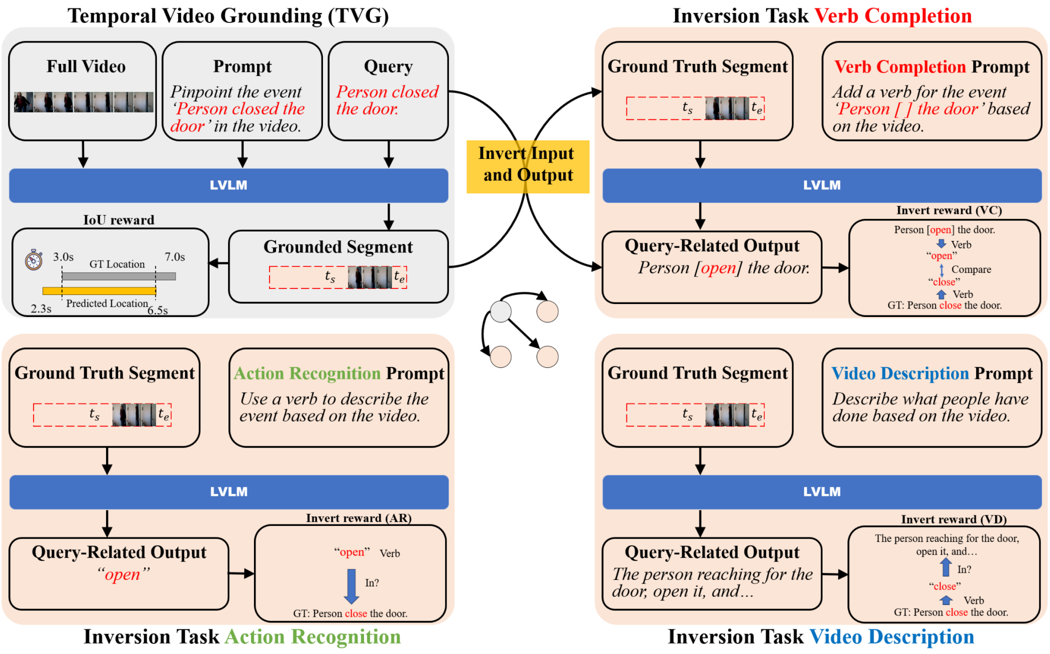

Temporal Video Grounding (TVG)
Full Video
Prompt
Pinpoint the event 'Person closed the door' in the video.
Query
Person closed the door.
LVLM
IoU reward
3.0s
GT Location
7.0s
2.3s
Predicted Location
6.5s
Grounded Segment
t_s
t_e
Invert Input and Output
Inversion Task Verb Completion
Ground Truth Segment
t_s
t_e
Verb Completion Prompt
Add a verb for the event 'Person [ ] the door' based on the video.
LVLM
Query-Related Output
Person [open] the door.
Invert reward (VC)
Person [open] the door.
Verb
"open"
Compare
"close"
Verb
GT: Person close the door.
Ground Truth Segment
t_s
t_e
Action Recognition Prompt
Use a verb to describe the event based on the video.
LVLM
Query-Related Output
"open"
Invert reward (AR)
"open"
Verb
In?
GT: Person close the door.
Inversion Task Action Recognition
Ground Truth Segment
t_s
t_e
Video Description Prompt
Describe what people have done based on the video.
LVLM
Query-Related Output
The person reaching for the door, open it, and...
Invert reward (VD)
The person reaching for the door, open it, and...
In?
"close"
Verb
GT: Person close the door.
Inversion Task Video Description

Figure 2: We propose three Invert-TVG tasks. By partially reversing the inputs and outputs of the TVG task we obtain Verb Completion, Action Recognition and Video Description, which reuse the original TVG dataset by taking ground truth video segments as input to reconstruct the target query related actions. The prompts for the three invert-TVG tasks are not identical. For VC, the verb in the query is removed, and the model is required to complete and fill in this verb. AR asks the model to directly estimate the verb in the video. VD requires the model to describe the video content containing action verbs in the query.

reinforcement learning algorithm proposed in (DeepSeek-AI, 2025)). Next, we introduce the proposed inversion TVG tasks, together with reward functions used to train the TVG and Invert-TVG tasks. Finally, we introduce our Invert4TVG reinforcement learning framework.

2.1 Preliminary of Group Relative Policy Optimization

DeepSeek-R1 (DeepSeek-AI, 2025), an early R1-style open-source LLM, uses GRPO to train policy π_θ for reasoning before answers. For query q, it generates responses $o_1, \ldots, o_G$ with score with $r(\cdot)$, and maximizes:

$$R(o) = \sum_{i=1}^{G} \frac{\pi_\theta\left(o_i\right)}{\pi_{\theta_{\text{old}}}\left(o_i\right)} \cdot \frac{r\left(o_i\right) - \text{mean}\left(\{r\left(o_i\right)\}_{i=1}^{G}\right)}{\text{std}\left(\{r\left(o_i\right)\}_{i=1}^{G}\right)}, \tag{3}$$

where $\pi_\theta(o_i)$ is generation probability, $\pi_{\theta_{\text{old}}}$ is prior state. The full objective with KL is:

$$\max_{\pi_\theta} \mathbb{E}_{o \sim \pi_{\theta_{\text{old}}}(p)} \left[R(o) - \beta \mathrm{D}_{\mathrm{KL}}\left(\pi_\theta \| \pi_{\text{ref}}\right)\right], \tag{4}$$

where β is a scaling coefficient. We omit the clipping operation for simplicity.

2.2 Invert-TVG Tasks and Reward Functions

In temporal video grounding, the accuracy of model inference highly depends on its understanding of both the video V and the query q. Therefore, we do not rely solely on the IoU reward but also introduce additional rewards to keep or even enhance the model's action understanding ability. To this end, as illustrated in Figure 2, we design three Invert-TVG tasks and define reward functions measuring how these tasks are fulfilled. Thanks to advances in NLP, a wealth of mature linguistic toolkits (e.g., SpaCy) can now effortlessly convert verbs into their various tenses or even bring them back to the root form, making it feasible for us to compute reward values for the following inversion tasks.

Verb Completion (fine granularity). Verb Completion is a task that masks verbs in the query and asks the model to recover the verbs from the ground truth video segment. For example, if the original query is “Person closed the door”, the masked sentence is “Person [ ] the door”. The prompt is “Add a verb for the event ‘Person [ ] the door’ based on the video”. As long as the model outputs a sentence successfully recovering the verbs of the ground truth, a reward is given. For instance, if the output sentence is “Person [closes] the door” a full reward is given. Due to the randomness in the model’s output, we are primarily concerned with whether the model comprehends the actions within the relevant segments. Therefore, we treat verbs in different tenses as equivalent using SpaCy. The reward function is as follows:

$$\mathrm{r}_{\mathrm{VC}}(o) = \begin{cases} 0 & \mathrm{SpaCy}(v_{\mathrm{pred}}) \neq \mathrm{SpaCy}(v_{\mathrm{gt}}) \\ 1 & \mathrm{SpaCy}(v_{\mathrm{pred}}) = \mathrm{SpaCy}(v_{\mathrm{gt}}) \end{cases} \tag{5}$$

where v_{pred} represents the predicted verb in the output sentence o, and v_{gt} represents the verb in the ground truth sentence. A full reward is obtained if the root form of the verbs are equal, where $\mathrm{SpaCy}(\cdot)$ brings a verb to its root form.

Action Recognition (middle granularity). Action Recognition task trains and preserves the model’s action perception capability, and the output of the model is fixed to a single verb. We feed the model the ground truth video segment with prompt as “Use a verb to describe the event based on the video”, then compare the predicted verb against any verbs appearing in the ground truth query description. If the model’s predicted verb is present in the reference sentence, it receives a full reward. For example, if the model outputs “walk” and the ground-truth sentence is “A person walks away and laughs”, a full reward is granted because “walk” occurs in the reference. As before, verb in different tenses are treated as equivalent. The reward function is as follows:

$$\mathrm{r}_{\mathrm{AR}}(o) = \begin{cases} 0 & \mathrm{SpaCy}(v_{\mathrm{pred}}) \notin S_{\mathrm{gt}} \\ 1 & \mathrm{SpaCy}(v_{\mathrm{pred}}) \in S_{\mathrm{gt}} \end{cases} \tag{6}$$

where v_{pred} represents the predicted verb o, and S_{gt} is the set of root-formed verbs in the ground truth query.

Video Description (coarse granularity). Video Description task is employed to train and maintain the model’s holistic perception of events, yielding a complete segment level description. Specifically, we feed the model the ground-truth video segment and prompt as “Describe what people have done based on the video”. A full reward is granted as long as ground-truth verbs appear in the output sentence of the model. For instance, if the ground-truth verb is “jump” and the model produces “A person jumps and laughs” the reward is awarded because “jump” is present. The reward function is as follows:

$$\mathrm{r}_{\mathrm{VD}}(o) = \begin{cases} 0 & \mathrm{SpaCy}(v_{\mathrm{gt}}) \notin S_{\mathrm{pred}} \\ 1 & \mathrm{SpaCy}(v_{\mathrm{gt}}) \in S_{\mathrm{pred}} \end{cases} \tag{7}$$

where v_{gt} represents the ground truth verb, and S_{pred} denotes the set of root-formed verbs in the sentence o predicted by the model.

2.3 IoU and Format Reward Functions

For the TVG task, we mainly employ the IoU reward function. Besides, we introduce a Format reward to enforce the model to output the thinking process.

IoU Reward. As stated above, TVG aims at estimating the time interval in the video that is associated with the content of a given textual query. We use the Intersection over Union (IoU) (Yuan et al., 2021) between the time interval predicted by the model and the ground-truth interval as the reward function. This reward function effectively describes the accuracy of the time interval predicted by the model. Given predicted $[t_s, t_e]$ and ground-truth $[t'_s, t'_e]$ segments, the IoU reward can be calculated as follows:

$$r_{\mathrm{IoU}}(o) = \frac{|[t_s, t_e] \cap [t'_s, t'_e]|}{|[t_s, t_e] \cup [t'_s, t'_e]|} \tag{8}$$

where $\cap$ and $\cup$ denote set intersection and union operations.

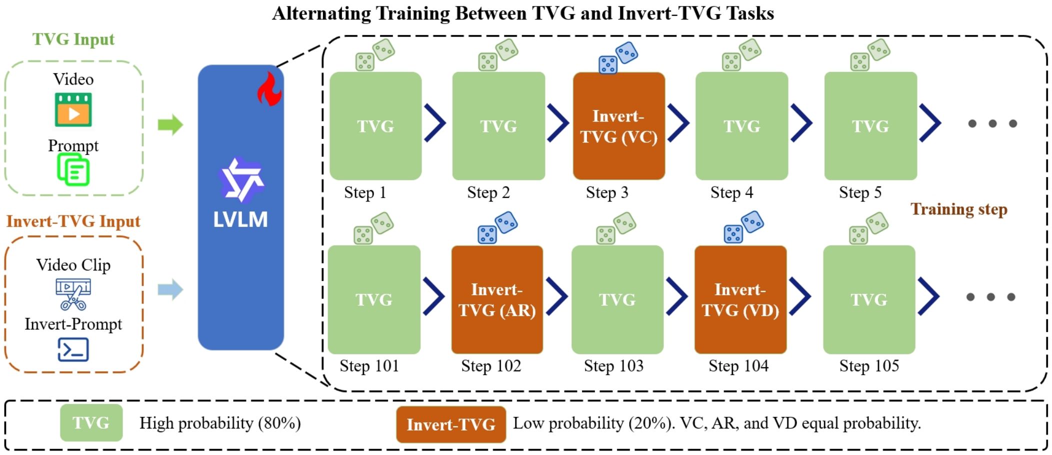


Figure 3: Overview of the proposed Invert4TVG framework. The LVLM dynamically chooses between TVG tasks and Invert-TVG tasks according to different probabilities. Whenever an Invert-TVG task is selected, one of the three variants VC, AR or VD is chosen with equal probability.

Format Reward. Recently, Time-R1 (Wang et al., 2025a) and VideoChat-R1 (Li et al., 2025) employ a Format reward making the model explicitly output its thinking process before making predictions. Following them, we introduce a template-based reasoning reward that incentivizes the model to generate intermediate reasoning steps prior to providing answers. The format is as following: `<think>***</think> <answer>` t_s to t_e `</answer>`. The reward is formulated as:

$$r_{\text{form}}(o) = \begin{cases} 0, & \text{if } o \text{ has wrong format} \\ 1, & \text{if } o \text{ has correct format} \end{cases} \tag{9}$$

2.4 Invert4TVG Reinforcement Learning Framework

While a joint training approach that processes all TVG and Invert-TVG tasks simultaneously might seem straightforward, this method suffers from several critical limitations: (1) Memory inefficiency: maintaining separate computation graphs for multiple tasks drastically increases GPU memory consumption; (2) Optimization conflict: gradient updates from different tasks may interfere with each other, especially when their loss landscapes are not aligned; (3) Training instability: the varying convergence rates of different tasks make it challenging to balance their contributions; (4) Task bias: the model may prioritize easier tasks while neglecting others. These drawbacks motivate us to adopt the training paradigm illustrated in Figure 3.

We implement a probabilistic sampling strategy where each training iteration has a high probability (80% in default) of executing the primary TVG task (using IoU and format rewards) and a low probability of performing an Invert-TVG task. When selecting Invert-TVG, we uniformly sample among VC, AR and VD. This design ensures the model maintains its core action understanding capabilities while primarily focusing on temporal grounding. The asymmetric probability distribution prevents the auxiliary tasks from overwhelming the main objective while still providing regular semantic reinforcement. Formally, the reward for training the TVG task is:

$$r_{\text{TVG}}(o) = r_{\text{format}}(o) + r_{\text{IoU}}(o). \tag{10}$$

The reward used to train an Invert-TVG task is:

$$r_{\text{Invert-TVG}}(o) = r_{\text{format}}(o) + r_{\text{inv}}(o), \tag{11}$$

where r_{inv} is any of r_{VC}, r_{AR}, and r_{VD}. The overall reward function is defined as:

$$r(o) = \alpha r_{\text{TVG}}(o) + \beta r_{\text{Invert-TVG}}(o), \tag{12}$$

where the two coefficients α and β take values in $\{0, 1\}$, with the constraint $\alpha + \beta = 1$. Their joint probability distribution is defined as (where $0 \leq p \leq 1$):

$$P(\alpha, \beta) = \begin{cases} p & \text{if } (\alpha, \beta) = (1, 0), \\ 1 - p & \text{if } (\alpha, \beta) = (0, 1), \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

As mentioned above, $p = 0.8$ is an empirically determined parameter derived from experiments.

3 Experiments

We now evaluate our Invert4TVG model on the task of temporal video grounding. Code is attached.

3.1 Experimental Setup

Benchmarks. We test our model on three temporal video grounding datasets: (1) Charades-STA (Sigurdsson et al., 2016) contains 6,672 long videos capturing indoor human activities. The official split for the TVG task includes 12,408 clip-query pairs for training and 3,720 for testing. (2) ActivityNet (Heilbron et al., 2015) comprises 20K long videos with an average of 3.65 clip-query pairs per video. We use the standard dataset splits with 37,421 training, 17,505 validation, and 17,031 test samples. (3) We further evaluate on QvHighlight (Lei et al., 2021), a high-resolution set of 10,460 long YouTube videos paired with 48k manually annotated clip-queries. To match Charades-STA and ActivityNet formats, multi-segment localizations are split into single-segment tasks, forming a balanced benchmark for fine-grained temporal grounding.

Implementation Details. We implement our LVLM using the Qwen2.5-VL model (Bai et al., 2025) as the backbone. To balance efficiency and memory consumption, we sample video frames at 2 FPS and resize them, resulting in approximately 2.8 million pixels per video (e.g., a 50-second video yields 100 frames of size 96 × 96 × 3). Our implementation utilizes SpaCy's `en_core_web_sm-3.8.0` model (12MB) to extract verbs from sentences and transform them across different tenses. For optimization, we employ the AdamW optimizer (Loshchilov & Hutter, 2019) with the following parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$, a weight decay of 0.0, and a learning rate of 5×10^{-5}. The training time per epoch is approximately 80 hours. To ensure reproducibility, our code, configuration files, and execution scripts are available in the supplementary materials.

Evaluation metrics. For TVG, we adopt the "R1@m" evaluation protocol to compare with state-of-the-art models, which computes the percentage of samples where the top-1 predicted segment has an IoU greater than a threshold m, with $m \in 0.3, 0.5, 0.7$. For brevity, we also adopt mIoU, which calculates the average IoU on all testing data as an alternative metric.

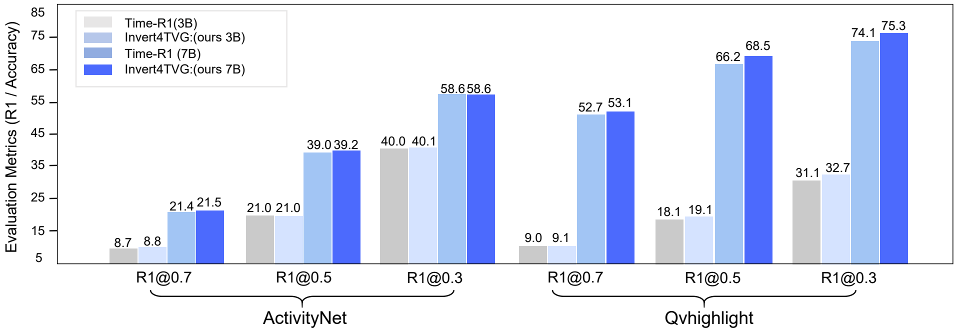


Figure 4: Performance of temporal video grounding on ActivityNet and QvHighlight. We compare our method with Time-R1 (the best-performing among previous methods). All the models are zero-shot tested.

Table 1: Performance of temporal video grounding on Charades-STA. Methods labeled FT with a ✓were fine-tuned on the Charades-STA training set. Methods marked with * were first pre-trained on extra TVG datasets[1] and then fine-tuned on the Charades-STA training set, while those without * are only trained on Charades-STA. We compare our method against existing 3B, 7B open-source LVLM. We highlight our results and the best-performing baselines using bold and underlining for clear comparison.

Type	Method	Size	FT	Charades-STA		
				R1@0.3	R1@0.5	R1@0.7
VLP	2D-TAN*	-	✓	57.3	45.8	27.9
	Moment-DETR*	-	✓	65.8	52.1	30.6
	EaTR*	-	✓	-	68.4	44.9
	SnAG*	-	✓	-	64.6	46.2
SFT	VideoChat-Flash	7B		74.5	53.1	27.6
	TRACE	7B		-	40.3	19.4
	HawkEye*	7B	✓	72.5	58.3	28.8
	TimeSuite*	7B	✓	79.4	67.1	43.0
RL(3B)	Time-R1(3B)	3B		74.6	53.1	26.0
	Time-R1*(3B)	3B	✓	78.7	64.1	36.9
	Invert4TVG (ours 3B)	3B	✓	**80.8**	**69.0**	**44.0**
RL(7B)	Time-R1 (7B)	7B		78.1	60.8	35.3
	Time-R1*(7B)	7B	✓	82.8	72.2	50.1
	Invert4TVG (ours 7B)	7B	✓	**83.0**	**72.5**	**51.4**

[1] YT-Temporal, DiDeMo, QuerYD, InternVid, HowTo100M datasets, our method is not pretrained on those datasets.

3.2 Comparison with State-of-the-Art Approaches

We compare Invert4TVG with state-of-the-art TVG methods, including both traditional video-language pre-training models (VLP), recent large video-language models fine-tuned via SFT and RL-based approaches.

Comparisons on the Charades-STA dataset with fine-tuning. As shown in Table 1, Invert4TVG surpasses not only VLP-based and SFT-based models but also outperforms RL-based approaches under identical conditions. For example, on Charades-STA, the 7B variant of Invert4TVG achieves an R1@0.7 of 51.4, exceeding TimeSuite (43.0), SnAG (46.2), and Time-R1 (50.1). The improvements are more pronounced for the 3B variant. Across R1@0.3, R1@0.5, and R1@0.7, Invert4TVG's 3B model outperforms the 3B version of Time-R1.

Comparisons on the ActivityNet and QvHighlight datasets in zero-shot settings. As shown in Figure 4, in zero-shot settings, Invert4TVG's 3B and 7B variants outperform Time-R1 on R1@0.3, R1@0.5, and R1@0.7 over ActivityNet. On QvHighlight, where we compare single-segment predictions, Invert4TVG consistently outperforms Time-R1 across R1@0.3, R1@0.5, and R1@0.7. ActivityNet contains only 200 action categories, whereas QvHighlight covers a significantly larger and more diverse set, with far more complex scene–action correlations. This disparity underscores the superiority of our method in understanding intricate actions.

3.3 Ablation Study

We conduct a detailed ablation on the Invert4TVG-3B model to investigate the contribution of the design strategies.

Using different combinations of Invert-TVG tasks versus employing them in combination. As shown in Table 2, using VC, AR, or VD alone instead of jointly yields lower performance. Only-VD, which emphasizes contextual understanding, peaks at R1@0.3 but falls short on precise localization. Only-AR, focused on immediate actions, reaches the highest R1@0.7 of 43.8. Only-VC outputs are less random than Only-VD yet less specific than Only-AR, achieving the best R1@0.5 (68.0). The mixed-task Invert4TVG surpasses all three individual tasks across all three metrics, demonstrating that joint training outperforms separate use.

The combination of VC and AR improves R1@0.7 to 43.8, outperforming either task alone (VC: 42.0; AR: 43.8), indicating complementary benefits between verb completion and action recognition. The VC+VD pair achieves the highest R1@0.3 (80.0) among two-task setups, suggesting that video description aids verb-focused localization. Invert4TVG (integrating all three auxiliary tasks) achieves the best overall results (R1@0.7: 44.0), demonstrating that multi-task synergy is maximized when all components are jointly optimized.

Table 2: Ablation study using only TVG (Time-R1), VC (verb Completion), AR (action recognition), VD (video description), and their mixed usage.

Type	Method	Charades-STA		
		R1@0.3	R1@0.5	R1@0.7
RL	Only-TVG	78.7	64.1	36.9
	Only-VD	79.1	64.3	39.4
	Only-AR	78.2	65.2	43.8
	Only-VC	78.8	68.0	42.0
	AR+VD	79.6	67.9	43.6
	VC+AR	78.8	68.1	43.8
	VC+VD	80.0	68.5	42.1
	Invert4TVG	80.8	69.0	44.0

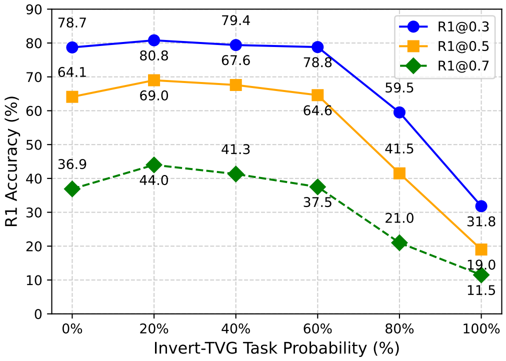


Figure 5: The R1 accuracy curves. Blue, orange, and green show how the three R1 metrics evolve as the Invert-TVG task probability $(1 - p)$ gradually increases.

Exploiting different probabilities for the TVG and Invert-TVG tasks. As shown in Figure 5, Varying the task probability markedly alters training outcomes. At 20% Invert-TVG task probability, the model performs best, raising R1@0.7 from 36.9 (no Invert task) to 44.0. As the Invert-TVG task probability grows, the model increasingly emphasizes action recognition while neglecting temporal grounding. Between 60% and 80% Invert-TVG, temporal video grounding performance steadily declines, falling below the pure TVG baseline. When Invert-TVG probability reaches 100 %, the model performs only the Invert-TVG task and yields the worst results. According to the experiments, we choose to set $p = 0.8$ in Eq. 13.

Table 3: The results using binary Invert-TVG reward or cosine similarity reward for training.

Reward	R1@0.3	R1@0.5	R1@0.7
Cosine Similarity	76.2	62.2	39.8
Binary 0 or 1	80.8	69.0	44.0

Binary Invert-TVG rewards vs. cosine similarity-based rewards. As shown in Table 3, we observe that employing a simple binary Invert-TVG reward (0 or 1) during training yields superior outcomes compared to more intricate reward mechanisms. When training for the same two epochs, the employed Invert-TVG Reward outperforms the cosine similarity reward across all three evaluation metrics (R1@0.3, R1@0.5, R1@0.7). This advantage stems from the controllability and stability of the binary reward design, whereas cosine similarity introduces higher variance and optimization instability. For example, in our implementation, “run” and “eat” yield a cosine similarity of 0.2 despite their weak semantic link. Therefore, binary Invert-TVG reward is a better choice.

4 Conclusion

In this work, we present Invert4TVG, an approach that introduces Invert-TVG tasks, requiring the model to generate query-related content from a video and its ground-truth temporal segment. We design three variants of Invert-TVG, including verb completion, action recognition, and video description. These tasks encourage the model to retain and enhance its action understanding capabilities. We develop a Invert4TVG RL framework that jointly optimizes TVG and Invert-TVG tasks. In addition to standard IoU and format rewards, we introduce Invert-TVG rewards to promote performance on Invert-TVG tasks. During training, the model primarily performs TVG at a high probability, while intermittently switching to Invert-TVG tasks at a lower probability. This balanced strategy ensures robust temporal localization while preserving semantic action-verb alignment. Our work bridges TVG-LVLM gap, unlocking higher extensions in traditional tasks. Experiments demonstrate the effectiveness of our method over existing approaches, achieving significant improvements of grounding accuracy. The reasoning process also shows that the proposed method indeed understands actions better than compared approaches.

ETHICS STATEMENT

This study exclusively utilizes publicly available open-source models and datasets; no proprietary or sensitive information is involved, and all data are free of personally identifiable content. We have strictly followed the corresponding licenses and usage guidelines. Although the present work poses no apparent ethical risks, we caution that—like many machine learning models—its outputs could be misapplied in unforeseen contexts. We therefore advocate responsible use and encourage ongoing efforts to identify and mitigate potential biases inherent in open-source datasets.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of our work, The models, training datasets, prompts, and hyperparameters used in our experiments are fully documented in Section 4.1 and Appendix C. These descriptions should allow researchers to replicate our experimental setup andresults without requiring additional resources beyond those specifed.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and Song. Qwen2.5-VL technical report, 2025.

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

Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. R1-V: Reinforcing super generalization ability in vision-language models with less than $3, 2025. URL https://github.com/Deep-Agent/R1-V.

Trevor Darrell and Alex Pentland. Space-Time gestures. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 335–340, New York, NY, USA, June 1993. IEEE.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Temporal localization of actions with Actoms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(11):2782–2795, 2013. doi: 10.1109/TPAMI.2013.65.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. TALL: Temporal activity localization via language query. In Proceedings of the IEEE International Conference on Computer Vision, pp. 5267–5275, 2017.

Aleksandr Gordeev, Vladimir Dokholyan, Irina Tolstykh, and Maksim Kuprashevich. Saliency-Guided DETR for moment retrieval and highlight detection, 2024.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 961–970, Boston, Mass., 2015. IEEE.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In Proceedings of the IEEE International Conference on Computer Vision, pp. 5803–5812, Venice, Italy, 2017. IEEE.

Hao Ju, Shaofei Huang, Si Liu, and Zhedong Zheng. Video2BEV: Transforming drone videos to BEVs for video-based geo-localization, 2024.

Ivan Laptev and Patrick Pérez. Retrieving actions in movies. In 2007 IEEE 11th International Conference on Computer Vision, pp. 1–8. IEEE, 2007.

Jie Lei, Tamara L. Berg, and Mohit Bansal. QVHighlights: Detecting moments and highlights in videos via natural language queries. In Advances in Neural Information Processing Systems, volume 34, pp. 12611–12623, 2021.

Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. VideoChat: Chat-centric video understanding, 2023.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. Videochat-flash: Hierarchical compression for long-context video modeling, 2024.

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. VideoChat-R1: Enhancing spatio-temporal perception via reinforcement fine-tuning, 2025. URL https://arxiv.org/abs/2504.06958.

Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Z Xu, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, and Weijie Kong. Egocentric video-language pretraining. In Advances in Neural Information Processing Systems, volume 35, pp. 7575–7586, 2022.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-RFT: Visual reinforcement fine-tuning, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In International Conference on Learning Representations. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In Advances in Neural Information Processing Systems, volume 35, pp. 27730–27744. NeurIPS, 2022.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. 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, pp. 14313–14323. IEEE, 2024.

Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. ReAct: Temporal action detection with relational queries. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 105–120. Springer, 2022. doi: 10.1007/978-3-031-20080-9_7.

Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In Proceedings of the European Conference on Computer Vision (ECCV), 2016.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, Xiangnan Fang, Zewen He, Zhenbo Luo, Wenxuan Wang, Junqi Lin, Jian Luan, and Qin Jin. Time-R1: Post-training large vision language model for temporal video grounding, 2025a. URL https://arxiv.org/abs/2503.13377.

Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, Xiangnan Fang, Zewen He, Zhenbo Luo, Wenxuan Wang, Junqi Lin, Jian Luan, and Qin Jin. TimeZero: Temporal video grounding with Reasoning-Guided LVLM, 2025b. URL https://arxiv.org/abs/2503.13377v1.

Yongliang Wu, Xinting Hu, Yuyang Sun, Yizhou Zhou, Wenbo Zhu, Fengyun Rao, Bernt Schiele, and Xu Yang. Number it: Temporal grounding videos like flipping manga. In Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 13754–13765, 2025.

Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. RLHF-V: Towards trustworthy MLLMs via behavior alignment from fine-grained correctional human feedback. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13807–13816, Seattle, WA, 2024. IEEE.

Yitian Yuan, Xiaohan Lan, Xin Wang, Long Chen, Zhi Wang, and Wenwu Zhu. A closer look at temporal sentence grounding in videos: Dataset and metric. In Proceedings of the 2nd International Workshop on Human-Centric Multimedia Analysis, pp. 13–21, 2021.

Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, et al. Timesuite: Improving mllms for long video understanding via grounded tuning. arXiv preprint arXiv:2410.19702, 2024.

Gengyuan Zhang, Mang Ling Ada Fok, Jialu Ma, Yan Xia, Daniel Cremers, Philip Torr, Volker Tresp, and Jindong Gu. Localizing events in videos with multimodal queries, 2024.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Temporal sentence grounding in videos: A survey and future directions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(8):10443–10465, 2023. doi: 10.1109/TPAMI.2023.3258628.

A PRINCIPLE OF ALGORITHM FOR INVERT4TVG

Algorithm 1 GRPO Training with Randomized Invert-TVG Task Selection

```
Input: Video $V$, query $q$, LVLM parameters $\theta$, probability $p$, reward function $r_{\text{TVG}}(o)$, a list of Invert-TVG tasks $T_{\text{Invert-TVG}} = \{\text{Invert-TVG}_i\}_{i=1}^{N}$ with corresponding rewards $\{r_i\}_{i=1}^{N}$, learning rate $\eta$.
Output: Optimized LVLM parameters $\theta$ balancing localization accuracy and video-language alignment.
Define forward TVG task: $\text{TVG}(V, q) \rightarrow \tau$
Define a list of Invert-TVG tasks, where each $\text{Invert-TVG}_i(V, \tau) \rightarrow q'_i$
while not converged do
    Sample a random value $u \sim \text{Uniform}(0, 1)$
    if $u \leq p$ then
        Select reward $r = r_{\text{TVG}}(o)$ {Optimize parameters related to $\tau$ (localization accuracy)}
        Compute gradient $\nabla_\theta r$ w.r.t. parameters affecting $\tau$
    else
        Randomly sample an Invert-TVG task $\text{Invert-TVG}_i \sim T_{\text{Invert-TVG}}$ {Select a task from the list}
        Generate its output $q'_i \leftarrow \text{Invert-TVG}_i(V, \tau)$
        Select the corresponding reward $r = r_i(o)$ {Optimize for the sampled Invert-TVG task}
        Compute gradient $\nabla_\theta r$ w.r.t. parameters affecting $q'_i$
    end if
    Update parameters: $\theta \leftarrow \theta + \eta \nabla_\theta r$ {Gradient ascent to maximize reward}
end while
Result: The model retains both localization accuracy (via $\tau$) and diverse video-language alignment (via various $q'_i$), which are complementary.
```

B THEORETICAL JUSTIFICATION FOR MULTI-TASK RL IN INVERT4TVG

To demonstrate the advantages of incorporating the Invert-TVG task into the RL framework, we provide a theoretical analysis showing that the multi-task approach improves semantic alignment and generalization compared to single-task TVG training. We follow the Pareto optimality framework in multi-task reinforcement learning, adapted to our setting where the joint reward balances temporal localization and semantic fidelity.

Let π_θ denote the policy (LVLM), and D the data distribution over videos V, queries q, and ground-truth segments τ. The single-task objective (TVG-only, as in prior works like Time-R1) maximizes:

$$\max_{\pi_\theta} \mathbb{E}_{o \sim \pi_\theta}[R_{\text{TVG}}(o)] - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}), \tag{14}$$

where $R_{\text{TVG}}(o) = r_{\text{IoU}}(o) + r_{\text{form}}(o)$.

In our multi-task setting, we introduce the joint reward $R_{\text{joint}}(o) = R_{\text{TVG}}(o) + \lambda R_{\text{Invert}}(o)$, with $\lambda > 0$ balancing the tasks. The objective becomes:

$$\max_{\pi_\theta} \mathbb{E}_{o \sim \pi_\theta}[R_{\text{joint}}(o)] - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}). \tag{15}$$

Lemma 1 (Semantic Alignment Improvement)**.** *The Invert-TVG task minimizes a semantic loss $L_{sem} = \mathbb{E}_{(V,\tau)\sim D}[d(q', q)]$, where $d(\cdot, \cdot)$ is a distance metric (e.g., verb matching or KL divergence on embeddings). Then, the joint loss satisfies $L_{joint} \leq L_{TVG} + C$ for some constant $C > 0$, as R_{Invert} provides positive feedback on semantic fidelity.*

Proof. By Jensen's inequality and non-negativity of $R_{\text{Invert}} \geq 0$ (binary rewards in our design), $\mathbb{E}[R_{\text{joint}}] \geq \mathbb{E}[R_{\text{TVG}}] + \lambda \min R_{\text{Invert}} \geq \mathbb{E}[R_{\text{TVG}}]$, assuming $R_{\text{Invert}} \geq 0$. This implies the multi-task policy reduces semantic drift, as Invert rewards enforce alignment (e.g., verb recovery). □

theorem 1 (Pareto Superiority)**.** *The multi-task policy π^*_{joint} is Pareto superior to the single-task policy π^*_{TVG} if there exists θ such that $R_{TVG}(\pi_\theta) \geq R_{TVG}(\pi^*_{TVG})$ and $R_{Invert}(\pi_\theta) > 0$.*

Proof. Consider the convex optimization formulation: minimize $L_{\text{TVG}} + \lambda L_{\text{Invert}}$. Assuming L_{Invert} is convex (e.g., cross-entropy-like semantic loss), the Pareto frontier dominates the single-task optimum. The KL regularizer ensures the multi-task solution lies on a superior frontier, as feedback from Invert reduces divergence: $D_{\text{KL}}(\pi_{\text{joint}} \| \pi_{\text{ref}}) \leq D_{\text{KL}}(\pi_{\text{TVG}} \| \pi_{\text{ref}}) - \Delta$ for $\Delta > 0$ from semantic regularization. □

Corollary 1 (Generalization Bound)**.** *In migrating TVG to LVLMs, semantic drift is reduced by Invert tasks, yielding a generalization error bound: $Err_{joint} \leq Err_{TVG} - \eta\lambda$, where η is a learning rate factor derived from multi-task boosting.*

This analysis justifies the inclusion of Invert-TVG, showing improved alignment and generalization on the Pareto front.

C More Implementation details

We implement our model using the Qwen2.5-VL model as the backbone, selected for its robust feature extraction capabilities in video understanding tasks. To balance training efficiency and memory constraints, we sample video frames at 2 frames per second (FPS), adaptively resizing each frame to maintain approximately 2.8 million pixels per video. For example, a 50-second video yields 100 frames, each with a resolution of approximately 96 × 96 × 3 pixels. During the reinforcement fine-tuning phase, we train the model for 2 epochs with a batch size of 4. All experiments are conducted on a cluster equipped with eight NVIDIA A100 GPUs (40GB memory each), using CUDA 11.8 and Python 3.10. For natural language processing tasks, we employ the `en_core_web_sm-3.8.0` model from the SpaCy library (12MB) to extract verbs from sentences. Random numbers between 0 and 1 are generated using numpy.random. The model is optimized using the AdamW optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.0. The learning rate is set to 5×10^{-5} . Training requires about 80 hours. All code, configurations, and preprocessing scripts are provided in the supplementary materials to ensure reproducibility.

D Ablation study

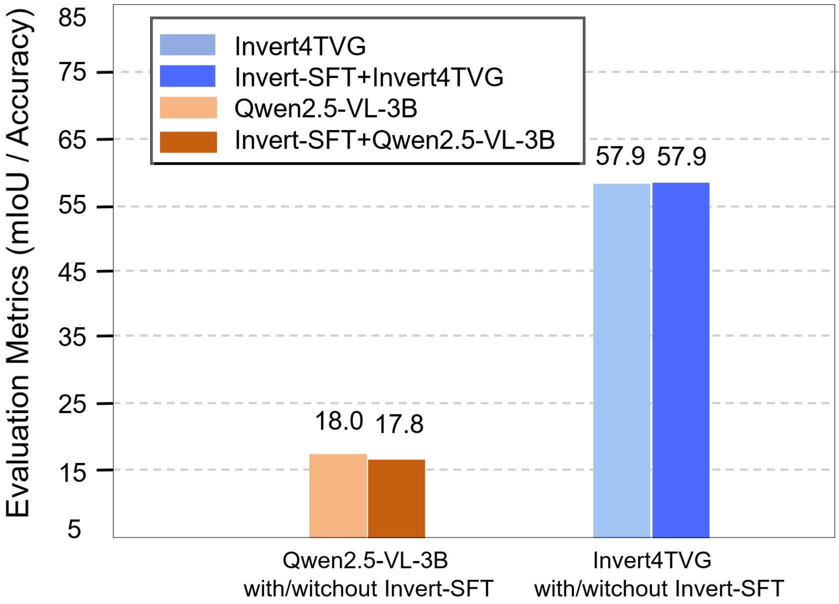
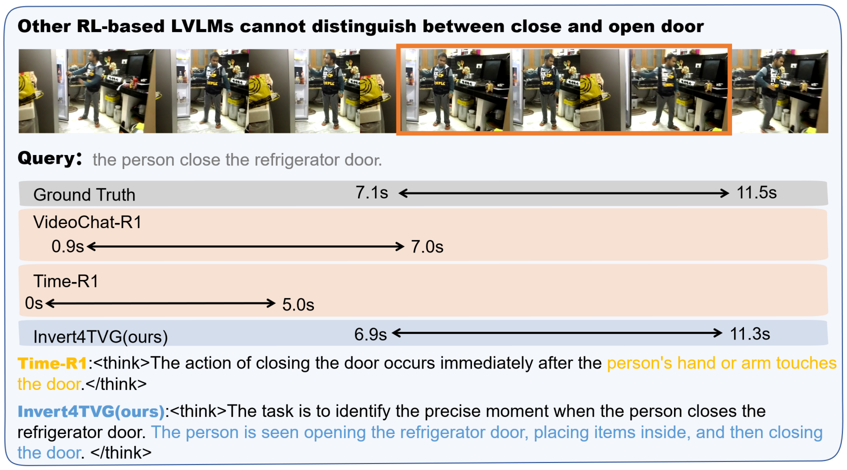


Figure 6: After Invert-SFT, the mIoU of Qwen2.5-VL-3B model and our Invert4TVG method are compared with those without Invert SFT.

Impact of Invert-SFT on Model Training. Invert-SFT refers to feeding ground-truth video clips into the model and requiring the model to output the corresponding event description based on these clips. The ground-truth clips are directly cropped from annotated temporal segments in temporal video grounding datasets, while the ”corresponding event” is the query to be localized. After Invert-SFT, the model’s outputs become more stable, facilitating subsequent training for Invert tasks. As shown in Figure 6, for the Qwen2.5-VL-3B model initialized with Invert-SFT, the mean Intersection-over-Union (mIoU) slightly decreased from 18.0 to 17.8 initially. However, after sufficient training, both Invert4TVG models, with and without Invert-SFT, reached convergence, achieving identical mIoU scores of 57.9.

E Qualitative Result



Other RL-based LVLMs cannot distinguish between close and open door

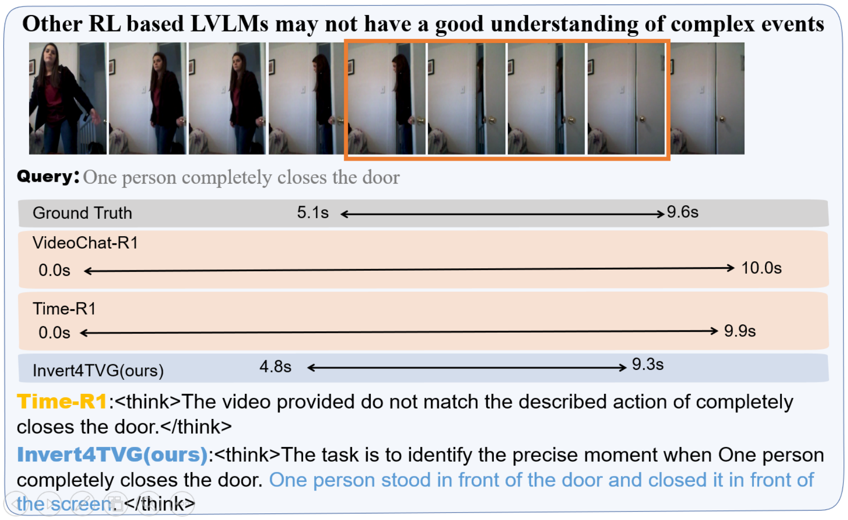
Query: the person close the refrigerator door.

Time-R1:<think>The action of closing the door occurs immediately after the person's hand or arm touches the door.</think>

Invert4TVG(ours):<think>The task is to identify the precise moment when the person closes the refrigerator door. The person is seen opening the refrigerator door, placing items inside, and then closing the door. </think>

Figure 7: success case 1

Time localization of similar actions. As shown in Figure 7, our method can more accurately identify similar actions, such as opening and closing doors. Many other models have insufficient understanding of similar actions, such as picking up and putting things in a box. In the model’s judgment, it is likely to be classified as the same action because the model does not fully recognize the state in which the action occurs and continues. Our method can help the model understand actions more deeply and distinguish similar actions.



Other RL based LVLMs may not have a good understanding of complex events

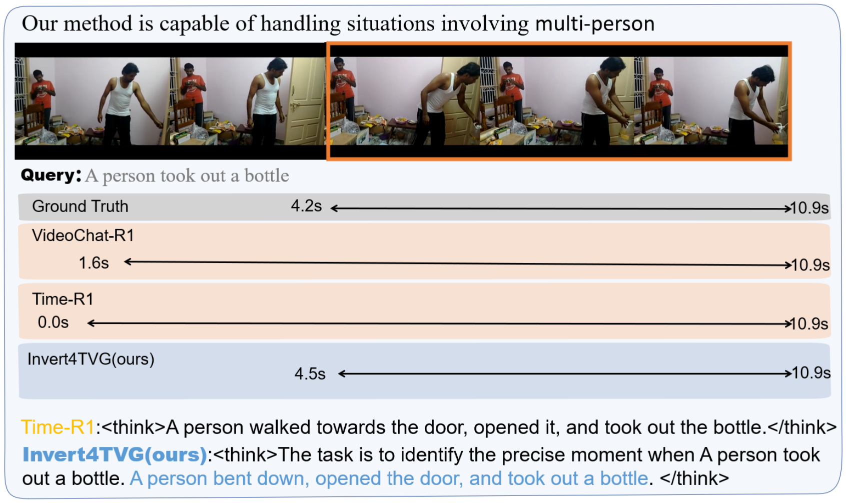
Query: One person completely closes the door

Time-R1:<think>The video provided do not match the described action of completely closes the door.</think>

Invert4TVG(ours):<think>The task is to identify the precise moment when One person completely closes the door. One person stood in front of the door and closed it in front of the screen. </think>

Figure 8: success case 2

Time localization of complex actions. As shown in Figure 8, our model recognizes the completion of the door closing action better. For some complex events, the time point we need to locate may occur at the completion of the action, rather than the beginning of the action. Other models sometimes consider the beginning of the action as the time point to be located, and then proceed with subsequent positioning from this time point. Our model has a clearer understanding of the start and end of the action and can effectively locate the time period when the action is in such a state of completion. Meanwhile, a correct understanding of the beginning and end of an action is also helpful for contextual reasoning.


Our method is capable of handling situations involving multi-person

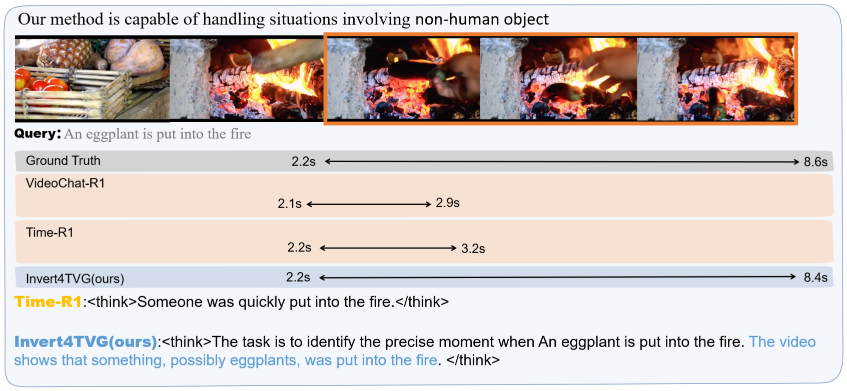
Query: A person took out a bottle

Time-R1:<think>A person walked towards the door, opened it, and took out the bottle.</think>

Invert4TVG(ours):<think>The task is to identify the precise moment when A person took out a bottle. A person bent down, opened the door, and took out a bottle. </think>

Figure 9: success case 3

Situations involving multi-person. As shown in Figure 9, our method demonstrates a strong capability in handling situations involving multiple persons. When two individuals appear in the frame, other models are often susceptible to interference from the secondary person, leading to extended temporal localization periods. In contrast, our model possesses a more accurate understanding of actions, enabling it to precisely identify the core action to be localized and the state of the target person, thereby achieving superior results.


Our method is capable of handling situations involving non-human object

Query: An eggplant is put into the fire

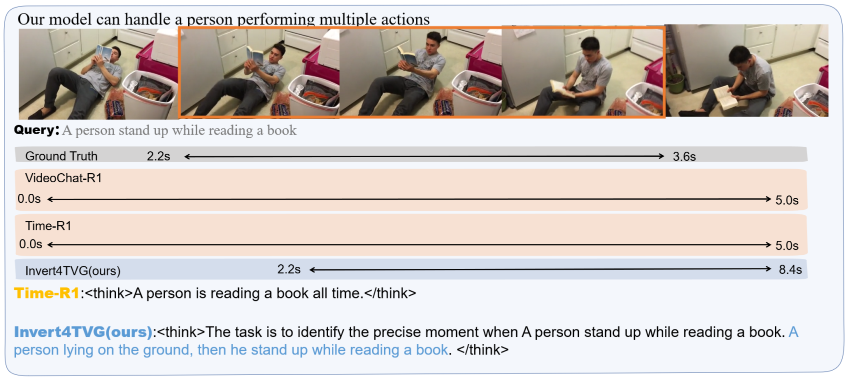
Time-R1:<think>Someone was quickly put into the fire.</think>

Invert4TVG(ours):<think>The task is to identify the precise moment when An eggplant is put into the fire. The video shows that something, possibly eggplants, was put into the fire. </think>

Figure 10: success case 4

Situations involving non-human object. As shown in Figure 10, our method is capable of handling scenarios where no person is present in the frame, such as when only a bonfire is shown and an eggplant is thrown into the fire. Other models, upon recognizing the keyword ”fire,” tend to predict very short temporal segments. Even if these predictions are accurate in timing, their Intersection over Union (IoU) remains low. In contrast, our approach focuses on understanding the action itself, resulting in predicted segments that are longer and closer to the ground truth temporal annotations.

A person performing multiple actions. As shown in Figure 11, the event to be localized involves a person reading a book while standing up. Other models focus only on a single action, namely


Our model can handle a person performing multiple actions
Ground Truth 2.2s 3.6s
VideoChat-R1
0.0s 5.0s
Time-R1
0.0s 5.0s
Invert4TVG(ours) 2.2s 8.4s

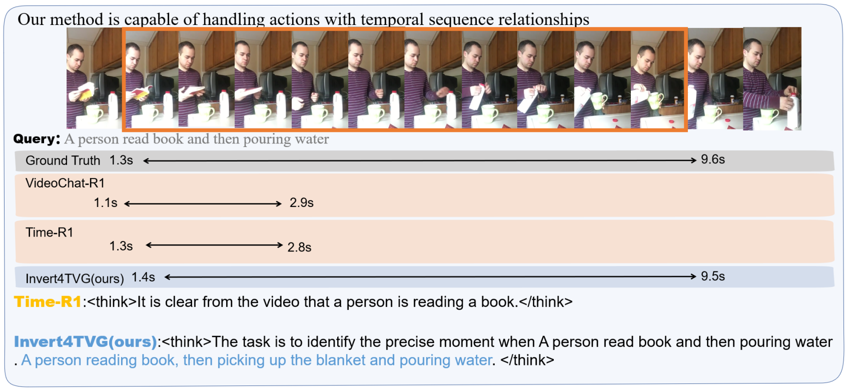
Query: A person stand up while reading a book

Time-R1:<think>A person is reading a book all time.</think>

Invert4TVG(ours):<think>The task is to identify the precise moment when A person stand up while reading a book. A person lying on the ground, then he stand up while reading a book. </think>

Figure 11: success case 5

”reading,” whereas our model disambiguates the actions, accurately identifying both the reading and the simultaneous act of standing up.


Our method is capable of handling actions with temporal sequence relationships
Ground Truth 1.3s 9.6s
VideoChat-R1
1.1s 2.9s
Time-R1
1.3s 2.8s
Invert4TVG(ours) 1.4s 9.5s

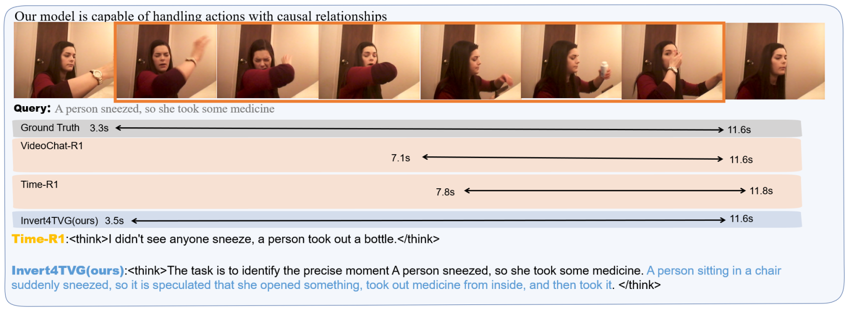
Query: A person read book and then pouring water

Time-R1:<think>It is clear from the video that a person is reading a book.</think>

Invert4TVG(ours):<think>The task is to identify the precise moment when A person read book and then pouring water . A person reading book, then picking up the blanket and pouring water. </think>

Figure 12: success case 6

A person performing multiple actions. As shown in Figure 12, the query contains a temporal cue such as “and then.” While other models treat the consecutive actions as a single event and attend more to the earlier action, our model recognizes both actions and their temporal order, yielding a more accurate localization.


Our model is capable of handling actions with causal relationships
Ground Truth 3.3s 11.6s
VideoChat-R1
7.1s 11.6s
Time-R1
7.8s 11.8s
Invert4TVG(ours) 3.5s 11.6s

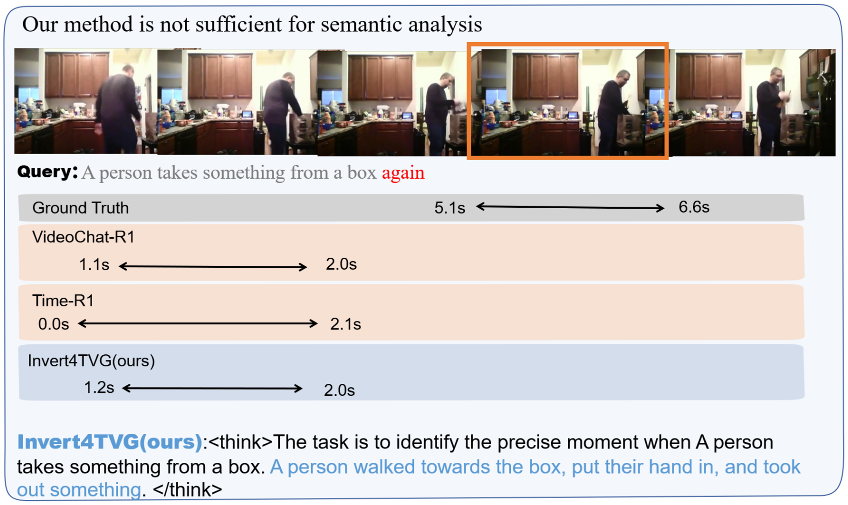
Query: A person sneezed, so she took some medicine

Time-R1:<think>I didn't see anyone sneeze, a person took out a bottle.</think>

Invert4TVG(ours):<think>The task is to identify the precise moment A person sneezed, so she took some medicine. A person sitting in a chair suddenly sneezed, so it is speculated that she opened something, took out medicine from inside, and then took it. </think>

Figure 13: success case 7

A person performing multiple actions. As shown in Figure 13, the event to be localized involves a causal relationship: a person in the video first sneezes and then takes medicine. Other models fail to accurately recognize the action of sneezing, leading them to rely on speculation and only localize the action of taking medicine. In contrast, our model successfully identifies both sneezing and taking medicine, understands the causal relationship between them, and achieves superior temporal localization results.


Our method is not sufficient for semantic analysis
Query: A person takes something from a box again
Ground Truth
5.1s
6.6s
VideoChat-R1
1.1s
2.0s
Time-R1
0.0s
2.1s
Invert4TVG(ours)
1.2s
2.0s

Invert4TVG(ours):<think>The task is to identify the precise moment when A person takes something from a box. A person walked towards the box, put their hand in, and took out something. </think>

Figure 14: failure case

Deep understanding of event semantics. As shown in Figure 14, our method is not sensitive enough to some qualifiers, such as ”again” representing the second occurrence of an action, which requires the model to accurately identify the action while also accurately finding the time period during which the second action occurred. Our method, as well as other models, has some shortcomings in this aspect. When locating time, we may find the first occurrence time as the final answer. This is because the model does not have a deep understanding of the meaning of qualifiers in the query and fully considers it when locating video time.

F USE OF LARGE LANGUAGE MODELS

We used large language models (LLMs) in a limited and auxiliary manner during the preparation of this paper. Specifically, LLMs were employed to improve the fluency and readability of the manuscript by polishing grammar and style, without altering the technical content. Importantly, LLMs were not involved in formulating research ideas, designing methods, conducting experiments, analyzing results, or drawing conclusions. All technical contributions of this paper are solely the work of the authors.