# BEYOND OPTIMISM: SELECTIVE TEMPORAL REINFORCEMENT LEARNING FOR VIDEO GROUNDING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Video temporal grounding localizes the temporal boundaries and occurrences of queried events. Existing methods either return a final prediction in a single pass or iteratively correct predictions based on candidate validity and overlap. The former leave residual localization errors uncorrected, while the latter overlook the benefit and degradation risk of changing the current answer. Our diagnostic study across three benchmarks shows that accepting the candidate with the highest verifier score degrades more queries than it improves. To address this issue, we propose **SeTeR**, a **Selective Temporal Reinforcement Learning** framework that learns the value of temporal edits and decides when to apply them. SeTeR revisits regions identified by a frozen grounder to propose boundary adjustments and splits of merged occurrences. We further introduce a learnable revision verifier that compares the current and proposed interval sets, jointly predicting improvement probability, degradation risk, and positive gain under supervision from the change in complete answer utility. Conditioned on these signals, a reinforcement learning policy selects edits or stops, optimizing grounding gains with penalties for degradation and unnecessary modifications. Experiments with TimeLens2-8B demonstrate improvements across all twelve reported metrics on three temporal grounding benchmarks, including mIoU gains of 1.22, 1.58, and 1.35 points. Transfer from QVHighlights to OMTG further improves event count accuracy by 8.12 points and F1 at the event level by 5.85 points, demonstrating the effectiveness of SeTeR for both boundary localization and event separation.

## 1 INTRODUCTION

Video temporal grounding aims to identify the temporal intervals in a video that correspond to a language query (Gao et al., 2017). It provides visual evidence for video retrieval, editing, and grounded question answering, where both accurate boundaries and complete event coverage are essential. A prediction may identify the relevant activity while omitting its beginning, including unrelated content, or merging separate occurrences into one interval. Reliable temporal grounding therefore requires precise localization of individual events and an accurate representation of how those events are distributed over time.

Early grounding research developed dedicated architectures for semantic alignment and temporal localization, as reviewed by Lan et al. (2023) and Zhang et al. (2023). TALL combined visual and language representations with boundary regression (Gao et al., 2017). 2D-TAN modeled relations among candidate moments on a temporal map, while VSLNet formulated localization as span prediction (Zhang et al., 2020b;a). Moment-DETR introduced direct set prediction of moments and saliency (Lei et al., 2021). These advances established temporal alignment, boundary estimation, and structured prediction as foundations for subsequent video language models.

Building on this foundation, recent approaches follow two main directions. The first improves the initial grounding prediction through instruction tuning, temporal representation learning, and reinforcement learning, returning it directly as the final answer. Models such as TimeLens and TimeLens2 improve timestamp prediction and support increasingly diverse temporal queries (Zhang et al., 2026; Zhu et al., 2026). The second introduces iterative observation and correction. TimeRefine learns successive boundary corrections, while VideoMind revisits candidate regions and uses a learned verifier to select a temporal prediction (Wang et al., 2026; Liu et al., 2026). FlashMMR



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

further considers interactions among moments and introduces verification supervised by temporal overlap (Cao et al., 2025). These methods create opportunities to correct errors through repeated prediction and candidate assessment.

Neither direction resolves how to correct residual errors while protecting accurate predictions. Returning the initial answer leaves errors intact, whereas iterative correction guided by candidate validity or overlap does not explicitly assess the benefit and degradation risk of replacing the current answer. The missing quantity is the change in complete answer utility induced by an edit. For example, a candidate with an IoU of 0.8 improves a prediction with an IoU of 0.6 but degrades one with an IoU of 0.9. Its correctness label remains unchanged under a fixed IoU threshold, although the appropriate revision decision is reversed. Candidate correctness therefore cannot determine whether an edit is beneficial. Reliable correction requires evaluating the proposed change relative to the existing answer.

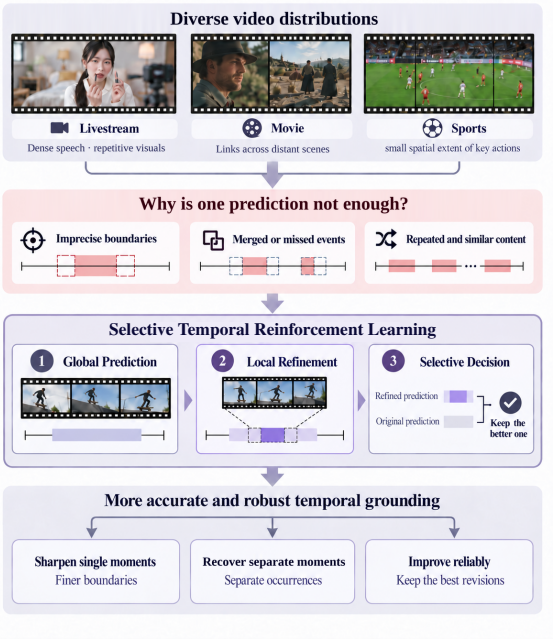


Figure 1: **Beyond optimism.** Local observation supplies temporal edits. A revision verifier estimates their benefit and risk, and a reinforcement learning policy selects edits or retains the current answer.

The distinction also extends beyond the quality of individual intervals. When a query refers to multiple occurrences, a locally accurate candidate may duplicate an event already covered by another interval, while splitting a broad prediction may improve event separation, temporal coverage, and event count accuracy. The value of an edit therefore depends on its effect on the complete interval set. Our diagnostic study across three benchmarks shows that direct selection by verifier score degrades more queries than it improves (Section 4.4). On QVHighlights, 939 queries degrade and 299 improve. Thus, available corrections can harm grounding accuracy when their effects on the current answer are overlooked. These findings motivate jointly learning edit value and when to revise or retain the answer.

To address this issue, we propose **SeTeR**, a **Selective Temporal Reinforcement Learning** framework that couples revision assessment with edit selection and stopping (Figure 1). A frozen grounder produces an initial interval set and revisits surrounding regions at finer temporal resolution. The resulting candidates adjust boundaries or separate merged occurrences.

We further introduce a learnable revision verifier that compares the current interval set with the proposed edited set. Its supervision comes from the change in complete answer utility, accounting for boundary accuracy and, for multiple moments, coverage and event structure. It jointly estimates **improvement probability**, **degradation risk**, and **positive gain**, characterizing the likelihood, risk, and magnitude of a beneficial change.

Building on these signals, we learn a policy that chooses a boundary edit, a split edit, or STOP. Reinforcement learning optimizes the return from actual utility changes, with penalties for degradation and unnecessary edits. The verifier assesses proposed changes; the policy learns how to use this assessment to act. Both editing and stopping decisions are therefore trained against their grounding outcomes.

Experiments evaluate SeTeR on both individual moment localization and grounding multiple occurrences. With TimeLens2-8B, SeTeR improves all twelve reported metrics across three benchmarks, with mIoU gains of 1.22, 1.58, and 1.35 points on Charades-TimeLens, ActivityNet-TimeLens, and QVHighlights-TimeLens, respectively. Transfer from QVHighlights to OMTG further improves

event count accuracy by 8.12 points and F1 at the event level by 5.85 points. Ablations compare candidate verification, direct gain regression, greedy selection, and the full learned policy, supporting both explicit revision assessment and learned decisions.

Our contributions are threefold. **First**, we introduce selective temporal reinforcement learning for video grounding, framing revision as a decision about the utility of edits to the complete answer. **Second**, we develop a revision verifier that learns improvement probability, degradation risk, and positive gain, explicitly connecting candidate assessment to boundary accuracy and event structure. **Third**, we develop a reinforcement learning policy that selects edits and learns when to stop, optimizing grounding gains while penalizing degradation and unnecessary changes. Experiments and ablations support the framework across three benchmarks and transfer to multiple occurrences.

## 2 Related Work

**Learning temporal grounding capabilities.** Temporal grounding requires associating query semantics with a location on the video timeline. Dedicated models such as VSLNet, Moment-DETR, and UniVTG learn this association through span localization, moment prediction, and unified temporal supervision (Zhang et al., 2020a; Lei et al., 2021; Lin et al., 2023). $R^2$-Tuning transfers image representations to video grounding through efficient temporal adaptation (Liu et al., 2024). With video language models, the challenge also involves expressing temporal positions in generated answers. TimeChat, VTimeLLM, LITA, and Momentor introduce temporal representations and supervision to connect visual content with timestamp prediction (Ren et al., 2024; Huang et al., 2024a;b; Qian et al., 2024). VTG-LLM incorporates timestamp knowledge, while Grounded-VideoLLM combines a dual stream encoder with discrete temporal tokens (Guo et al., 2025a; Wang et al., 2025a). TimeMarker uses temporal separators and adaptive sampling, and NumPro exposes frame indices through visible numerical prompts (Chen et al., 2024; Wu et al., 2025). A complementary direction improves the supervision itself: TimeSuite ties detailed descriptions to timestamps through grounded tuning, and Seq2Time converts image and clip sequences into temporal training examples (Zeng et al., 2025; Deng et al., 2025). TRACE explicitly generates structured events containing timestamps, saliency, and captions, making their dependencies part of the prediction process (Guo et al., 2025b). Time-R1 and TimeLens further study reinforcement learning for localization, with TimeLens also emphasizing annotation quality (Wang et al., 2025b; Zhang et al., 2026). SeTeR applies reinforcement learning to edit selection and stopping over the predictions of a frozen grounder.

**Repeated observation and temporal refinement.** A limited frame budget makes precise localization difficult when observations are spread over a long video. Revisiting a smaller region increases the temporal detail available for that region. HawkEye and ReVisionLLM exploit this principle by recursively narrowing the search region, while UniTime uses adaptive frame scaling to handle different temporal granularities (Wang et al., 2024; Hannan et al., 2025; Li et al., 2025). TimeRefine instead reformulates timestamp prediction as repeated correction: the model produces a rough interval and then predicts offsets that bring it closer to the target (Wang et al., 2026). VideoChat-R1.5 and VideoZoomer learn to acquire finer visual evidence during inference (Yan et al., 2025; Ding et al., 2026). ReCrossVLLM couples coarse grounding with reflective observation, whereas MoD-VLLM allocates different observation granularities to relevant and background segments (Feng et al., 2026a;b). Attention cues offer another way to guide cropping or masking before regenerating an answer (Du et al., 2026). SeTeR evaluates each local prediction as an optional edit to the complete answer.

**Candidate verification and revision value.** Verification assesses whether temporal predictions match the queried event. VideoMind separates planning, grounding, verification, and answering through a Chain-of-LoRA workflow (Liu et al., 2026). Its verifier examines an expanded candidate region and learns a binary judgment using an IoU threshold of 0.5; inference ranks candidates by the predicted validity score. FlashMMR goes beyond independent moment prediction by incorporating interactions among moments, constrained boundary adjustment, and verification supervised by candidate overlap with reference intervals (Cao et al., 2025). Our distinction concerns the supervision target: a candidate's validity or overlap describes its quality, whereas an edit's utility describes the change induced in an existing answer. In a temporal set, that change can also affect duplication,

event count, and coverage supplied by other intervals. SeTeR supervises revision utility through improvement probability, degradation risk, and positive gain.

**Multiple moments and selective revision.** Queries can refer to several occurrences, making the completeness and structure of the answer important alongside boundary precision. MUSEG reinforces timestamp awareness through grounding multiple segments, and TimeLens2 extends grounding to diverse temporal queries (Luo et al., 2026; Zhu et al., 2026). OMTG makes the recovery of multiple relevant events an explicit evaluation objective (Xu et al., 2026). These developments motivate representing refinement as an operation on an interval set: splitting a merged span changes the answer's structure as well as its boundaries. More broadly, Self-Refine uses feedback to revise generated outputs, while studies of intrinsic self correction show that another revision can also reduce accuracy (Madaan et al., 2023; Huang et al., 2024c). SelectiveNet learns when to abstain, and learning to defer optimizes when to delegate a prediction to an expert (Geifman and El-Yaniv, 2019; Mozannar and Sontag, 2020). SeTeR learns whether to revise or retain the current grounding answer through rewards based on complete answer utility.

# 3 METHOD

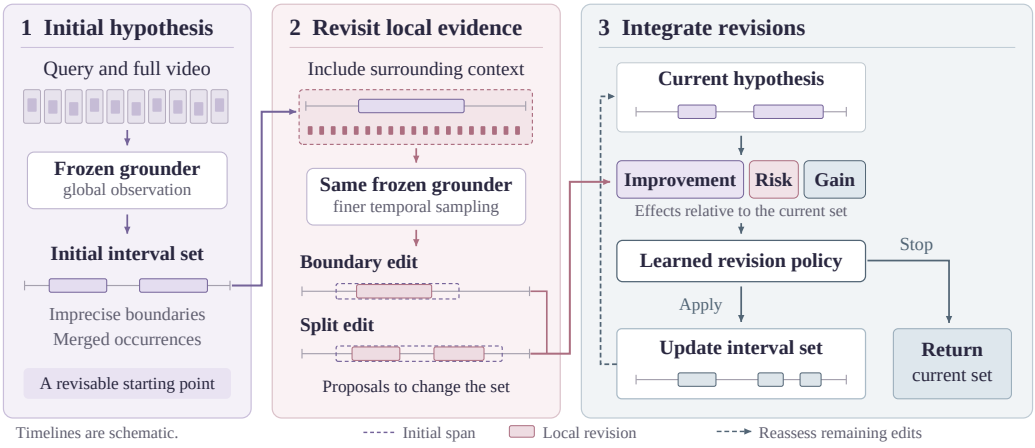


Figure 2: **Selective temporal reinforcement learning.** A frozen grounder produces an initial interval set and local boundary or split edits. The revision verifier estimates improvement probability, degradation risk, and positive gain. A reinforcement learning policy uses these signals to select an edit or STOP. Accepted edits update the set and the assessment of remaining proposals; stopping returns the current answer. Timelines are schematic.

## 3.1 THE INITIAL TEMPORAL HYPOTHESIS

Given an untrimmed video $V$ of duration $T$ and a language query $q$, video temporal grounding predicts an interval set:

$$S = \{I_k\}_{k=1}^{K}, \qquad I_k = [s_k, e_k], \quad 0 \le s_k < e_k \le T, \tag{1}$$

where $K$ varies with the number of relevant occurrences and equals one for grounding a single moment. Let $G$ denote the reference interval set and $M(S, G)$ a grounding utility. The utility measures interval IoU for one moment or accounts for the structure of a set when multiple moments are present.

SeTeR learns to assess and select temporal edits while keeping the base grounder $f_{\text{base}}$ frozen (Figure 2). We instantiate the grounder as TimeLens2-8B (Zhu et al., 2026). Its observation of the full video produces the initial temporal hypothesis:

$$S^{(0)} = f_{\text{base}}(V, q). \tag{2}$$

The intervals in $S^{(0)}$ guide further observation. SeTeR converts local predictions into candidate edits and learns which to accept, with the current set always available as an answer. An edit can change both boundaries and event count, so its effect is evaluated on the complete interval set.

### 3.2 Revisiting Local Evidence and Proposing Edits

For an interval $I_i = [s_i, e_i] \in S^{(t)}$, let $\ell_i = e_i - s_i$ and let $\rho \geq 0$ determine the surrounding context. The observation window is:

$$C_i^{(t)} = \left[\max(0, s_i - \rho\ell_i),\ \min(T, e_i + \rho\ell_i)\right]. \tag{3}$$

This expansion makes evidence outside the predicted endpoints available for correction. We query the same frozen grounder on this crop at finer temporal resolution using $q$. Its local prediction is an interval set in crop coordinates:

$$L_i^{(t)} = \{\tilde{I}_{ij}\}_{j=1}^{m_i}, \tag{4}$$

which is mapped back to the full video timeline. A denser local observation can adjust inaccurate boundaries or separate nearby occurrences covered by one coarse interval.

Each local prediction proposes an edit to the current hypothesis:

$$e_i^{(t)}:\ S^{(t)} \mapsto S_{e_i}^{(t)} = \left(S^{(t)} \setminus \{I_i\}\right) \cup L_i^{(t)}. \tag{5}$$

When $m_i = 1$, the edit revises a boundary; when $m_i > 1$, it replaces a merged region with several occurrences. Retaining $I_i$ remains possible. We generate four local predictions per current interval, keeping candidates with at least one valid interval. After coordinate conversion and exact deduplication, at most four candidates remain per interval. The available proposals form:

$$\mathcal{E}^{(t)} = \{e_1^{(t)}, \ldots, e_N^{(t)}\}. \tag{6}$$

### 3.3 Learning Revision Benefits and Risks

The value of a local prediction depends on the current answer. Training covers initial predictions and states reached by random edits. The reference set $G$ supplies the utility difference associated with each edit:

$$\Delta(e; S^{(t)}, G) = M(S_e^{(t)}, G) - M(S^{(t)}, G), \tag{7}$$

where positive values indicate improvement and negative values indicate degradation. We derive three supervision targets from this relative effect.

**Improvement probability.** The improvement signal estimates whether the edit improves the current set beyond a margin $\delta_{\text{imp}} = 0.01$:

$$y_{\text{imp}}(e) = \mathbf{1}[\Delta(e) > \delta_{\text{imp}}], \tag{8}$$

with prediction $p_{\text{imp}}(e) \in [0, 1]$. This target identifies edits that improve the current answer, rather than candidates that merely satisfy a correctness threshold.

**Degradation risk.** The risk signal estimates whether the edit damages the current answer beyond margin $\delta_{\text{deg}} = 0.01$:

$$y_{\text{deg}}(e) = \mathbf{1}[\Delta(e) < -\delta_{\text{deg}}], \tag{9}$$

with prediction $p_{\text{deg}}(e) \in [0, 1]$. Conditioning this signal on both the current and edited sets makes it specific to the proposed revision.

**Positive gain.** The gain signal captures the positive part of the utility change:

$$y_{\text{gain}}(e) = \max(\Delta(e), 0), \tag{10}$$

with prediction $g_+(e) \geq 0$. This target records the magnitude of a beneficial change and is zero for edits that do not improve the set.

For each current set and proposed edit, the learned evaluator uses available grounding evidence and temporal structure to predict the three signals. Its supervised objective is:

$$\mathcal{L}_{\text{cal}} = \mathcal{L}_{\text{BCE}}(p_{\text{imp}}, y_{\text{imp}}) + 2\mathcal{L}_{\text{BCE}}(p_{\text{deg}}, y_{\text{deg}}) + \lambda_{\text{gain}}\mathcal{L}_{\text{reg}}(g_+, y_{\text{gain}}), \tag{11}$$

where $\mathcal{L}_{\text{reg}}$ is a robust regression loss. At inference, a conservative edit priority combines improvement and gain signals with a degradation penalty:

$$A(e) = p_{\text{imp}}(e)\, g_+(e) - \lambda_{\text{risk}} p_{\text{deg}}(e). \tag{12}$$

Multiplying $p_{\text{imp}}$ and $g_+$ emphasizes proposals that have both a positive gain estimate and a high improvement probability. The risk penalty discounts edits likely to damage the current hypothesis. These learned estimates guide acceptance without requiring the reference intervals at inference.

### 3.4 Selective Temporal Reinforcement Learning

The verifier supplies estimates of edit value. We learn how to use these estimates to choose an edit or preserve the current answer through a short decision process. At step $t$, the policy observes the current hypothesis, available proposals, and their estimated effects:

$$x_t = \left(V, q, S^{(t)}, \{e, p_{\text{imp}}(e), p_{\text{deg}}(e), g_+(e)\}_{e \in \mathcal{E}^{(t)}}\right). \tag{13}$$

The policy $\pi_\theta$ chooses an edit or stops revision:

$$a_t \in \{\textsc{Stop}\} \cup \mathcal{E}^{(t)}, \tag{14}$$

where STOP returns the current hypothesis. Accepting edit $e$ updates the state through:

$$S^{(t+1)} = S_e^{(t)}, \tag{15}$$

after which the remaining proposals are evaluated relative to the updated set. A Set Transformer policy combines verifier embeddings and predicted revision effects with temporal geometry and overlap features, using a pointer head for edits and a separate STOP token.

We train this policy using rewards derived from the actual effect of its decisions. The immediate utility change is:

$$d_t = M(S^{(t+1)}, G) - M(S^{(t)}, G). \tag{16}$$

The reward encourages improvements while penalizing harmful and unnecessary changes:

$$r_t = d_t - 1.0 \max(-d_t, 0) - 0.005\, \mathbf{1}[a_t \neq \textsc{Stop}], \tag{17}$$

where the degradation penalty favors retaining accurate predictions and the edit cost encourages stopping when a change offers little benefit. A STOP action leaves the answer unchanged and has zero immediate reward. At termination, an additional reward $r_{\text{terminal}} = 0.5\big(M(S^{(T)}, G) - M(S^{(0)}, G)\big)$ reinforces improvement in the final answer. With $H$ policy decisions, the objective is:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{H-1} r_t + r_{\text{terminal}}\right]. \tag{18}$$

We initialize the policy by behavior cloning with the priorities in Eq. (12) and reference gains, then optimize Eq. (18) using REINFORCE with a learned value baseline (Williams, 1992). Each episode starts from the global prediction and accepts at most three edits. Appendix A gives the architecture and training settings.

### 3.5 Inference

Starting from $S^{(0)}$, SeTeR constructs local edits and estimates their improvement probability, degradation risk, and positive gain. The policy applies an accepted edit, updates the set, and reevaluates the remaining proposals. Revision ends at STOP, when no proposal exceeds the revision threshold, or after three accepted edits, yielding:

$$S^* = S^{(T)}. \tag{19}$$

The output may retain an original interval, change its boundaries, or replace it with several occurrences. The initial hypothesis guides where to look, while learned revision determines which parts of the new evidence enter the final answer.

# 4 EXPERIMENTS

## 4.1 EXPERIMENTAL SETUP

We evaluate whether revision improves both temporal boundaries and event structure. Table 1 presents results on Charades-TimeLens, ActivityNet-TimeLens, and QVHighlights-TimeLens, the three reannotated subsets of TimeLens-Bench (Zhang et al., 2026). Their source datasets are Charades-STA (Gao et al., 2017), built on Charades (Sigurdsson et al., 2016); ActivityNet Captions (Krishna et al., 2017), built on ActivityNet (Heilbron et al., 2015); and QVHighlights (Lei et al., 2021). The main comparison uses the TimeLens reannotations rather than the original annotations. We report mIoU and Recall@1 at temporal IoU thresholds of 0.3, 0.5, and 0.7, denoted R@0.3, R@0.5, and R@0.7. SeTeR uses TimeLens2-8B (Zhu et al., 2026) as its base grounder.

**Baselines.** We compare with GPT-5 and Gemini 2.5 Pro, alongside open source models including Qwen3.5, Qwen3-VL, LLaVA-OneVision-2, and InternVideo3 (OpenAI, 2025; Gemini Team, 2025; Qwen Team, 2026; Bai et al., 2025; An et al., 2026; Yan et al., 2026). Dedicated grounding comparisons include TimeSuite, TRACE-uni, Grounded-VideoLLM, Time-R1, TimeLens, and TimeLens2 (Zeng et al., 2025; Guo et al., 2025b; Wang et al., 2025a;b; Zhang et al., 2026; Zhu et al., 2026). All comparisons use the TimeLens reannotations.

TimeLens2-8B remains frozen throughout SeTeR. Supervised learning trains the revision verifier to predict improvement probability, degradation risk, and positive gain from utility changes. Behavior cloning followed by REINFORCE trains the selection and stopping policy. Both modules use the initial predictions and evidence from local revisiting.

## 4.2 REFINING THE BOUNDARIES OF INDIVIDUAL MOMENTS

Table 1: **Single moment temporal grounding on TimeLens-Bench.** All metrics are percentages (↑); **bold** and underline denote the highest and second highest scores, respectively.

| Method | Charades-TimeLens | | | | ActivityNet-TimeLens | | | | QVHighlights-TimeLens | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | mIoU | R@0.3 | R@0.5 | R@0.7 | mIoU | R@0.3 | R@0.5 | R@0.7 | mIoU | R@0.3 | R@0.5 | R@0.7 |
| *Closed source generalist MLLMs* | | | | | | | | | | | | |
| GPT-5 | 40.5 | 59.3 | 42.0 | 22.0 | 42.9 | 57.4 | 44.9 | 30.4 | 56.8 | 72.4 | 60.4 | 46.4 |
| Gemini 2.5 Pro | 52.8 | 74.1 | 61.1 | 34.0 | 58.1 | 72.3 | 64.2 | 47.1 | 70.4 | 84.1 | 75.9 | 61.1 |
| *Open source generalist MLLMs* | | | | | | | | | | | | |
| Qwen3.5-397B-A17B | 47.5 | 69.3 | 47.8 | 24.9 | 53.8 | 70.6 | 59.1 | 41.7 | 65.8 | 82.1 | 71.6 | 56.1 |
| Qwen3-VL-235B-A22B | 47.8 | 71.7 | 50.8 | 24.5 | 52.2 | 69.0 | 57.5 | 39.3 | 64.6 | 79.6 | 70.2 | 54.5 |
| LLaVA-OneVision-2-8B | 52.6 | 73.1 | 59.6 | 34.0 | 52.4 | 65.6 | 57.9 | 40.9 | 65.7 | 78.2 | 70.0 | 57.0 |
| Qwen3-VL-8B | 47.2 | 68.5 | 49.3 | 26.2 | 48.0 | 63.8 | 51.1 | 35.0 | 59.4 | 74.4 | 62.9 | 49.3 |
| InternVideo3-8B | 53.2 | 75.8 | 61.7 | 32.8 | 46.3 | 60.0 | 51.4 | 33.4 | 59.2 | 72.2 | 62.6 | 48.9 |
| *Specialized Temporal Grounding Models* | | | | | | | | | | | | |
| TimeSuite | 38.1 | 56.3 | 35.5 | 18.0 | 19.8 | 27.1 | 17.5 | 8.6 | 21.7 | 27.1 | 16.9 | 9.9 |
| TRACE-uni | 28.1 | 38.2 | 22.9 | 10.4 | 33.6 | 44.3 | 35.1 | 22.6 | 39.8 | 49.9 | 40.0 | 29.2 |
| Grounded-VideoLLM | 30.0 | 43.3 | 28.7 | 13.5 | 30.0 | 39.2 | 29.6 | 19.5 | 33.4 | 43.7 | 33.8 | 22.5 |
| Time-R1-7B | 36.6 | 57.9 | 32.0 | 16.9 | 33.1 | 44.8 | 31.0 | 19.0 | 49.2 | 65.8 | 51.5 | 36.1 |
| TimeLens-8B | 57.0 | 78.8 | 65.8 | 37.6 | 56.3 | 71.8 | 62.0 | 43.8 | 66.8 | 81.2 | 71.7 | 57.1 |
| **TimeLens2-8B** | 58.60 | 80.90 | 68.00 | 38.90 | 58.60 | 74.00 | 65.60 | 46.60 | 70.20 | 84.40 | 76.10 | 61.20 |
| *Selective Temporal Reinforcement Learning (Ours)* | | | | | | | | | | | | |
| **SeTeR (ours)** | **59.82** | **81.68** | **69.44** | **40.85** | **60.18** | **75.56** | **66.48** | **48.25** | **71.55** | **84.82** | **77.42** | **63.47** |

**Results.** SeTeR achieves the highest scores across all twelve metrics. Relative to TimeLens2-8B, mIoU improves by **1.22**, **1.58**, and **1.35** points on Charades-TimeLens, ActivityNet-TimeLens, and QVHighlights-TimeLens, respectively, while R@0.7 improves by **1.95**, **1.65**, and **2.27** points. Relative to the strongest baseline in each mIoU column, the gains are **1.22**, **1.58**, and **1.15** points, respectively.

### 4.3 Recovering Event Structure through Transfer

Table 2: **QVHighlights → OMTG transfer.** Baseline is TimeLens2-8B. All metrics are percentages (↑); Δ is the absolute gain.

| Metric | Baseline | Ours | Δ |
|---|---|---|---|
| C-Acc | 16.88 | **25.00** | +8.12 |
| tF1@0.3 | 49.04 | **58.26** | +9.22 |
| tF1@0.5 | 39.30 | **49.42** | +10.12 |
| tF1@0.7 | 26.74 | **35.94** | +9.20 |
| tIoU | 48.93 | **51.56** | +2.63 |
| EtF1 | 13.21 | **19.06** | +5.85 |

Figure 3: **Recovering two occurrences on OMTG 4.** Gray: ground truth; red: TimeLens2; blue: SeTeR. Parentheses give interval counts.




**Transfer setting.** Recovering separate occurrences requires revising the structure of the answer as well as its boundaries. We apply the framework trained on QVHighlights (Lei et al., 2021) directly to OMTG (Xu et al., 2026) without training on OMTG. Table 2 compares its output with the original TimeLens2-8B predictions.

**Metrics and results.** Following Xu et al. (2026), C-Acc measures exact interval count agreement, tF1 evaluates matched intervals, tIoU measures overlap between temporal unions, and EtF1 additionally penalizes incorrect counts. SeTeR improves all six metrics, including gains of **8.12** points in C-Acc and **5.85** points in EtF1.

**Recovering separate occurrences.** Figure 3 shows a query describing a girl wearing a face mask with a heart on it walking on the street. TimeLens2 merges two separate occurrences into one interval. SeTeR separates them into two intervals, recovering the correct event count and increasing Set-IoU from 0.654 to 0.946. Removing the gap between occurrences improves the structure of the complete answer alongside its temporal overlap. Additional cases appear in Appendix D.2.

Additional QV-M$^2$ results and revision error analyses appear in Appendices C.1 and D.3.

### 4.4 Why Revisions Need Selective Integration

**Setup and results.** We evaluate *score argmax*, which selects the candidate with the highest verifier score without a gate that can retain the initial prediction. C-STA, AN, and QV denote Charades-STA, ActivityNet, and QVHighlights. Score argmax falls **4.07 to 5.86 mIoU points** below TimeLens2. Degraded queries outnumber improved queries on every dataset: on QVHighlights, 939 queries degrade and 299 improve. Detailed counts and recall comparisons appear in Appendix B.1. These results motivate assessing edits relative to the current answer.

Table 3: **Ungated candidate selection.** Scores are mIoU (%); $\Delta$ = Score argmax − TimeLens2.

| | C-STA | AN | QV |
|---|---|---|---|
| TimeLens2 | 58.60 | 58.60 | 70.20 |
| Score argmax | 54.53 | 54.27 | 64.34 |
| Δ mIoU | −4.07 | −4.33 | −5.86 |

### 4.5 Correction Opportunities in Candidate Pools

**Available correction opportunities.** Repeated predictions expose alternatives that a single output does not reveal. For query $i$, let $\mathcal{C}_i$ be a fixed candidate pool and let $u_i(c) \in [0, 1]$ be the IoU score of candidate $c$ for that query under the corresponding evaluation protocol. Using ground truth, the oracle selects the best candidate separately for every query:

$$O(\mathcal{C}) = \frac{100}{N} \sum_{i=1}^{N} \max_{c \in \mathcal{C}_i} u_i(c). \tag{20}$$

The oracle measures the best localization available in each candidate pool and provides a reference for evaluating selection quality. Appendix B.3 derives the corresponding upper bound.

Table 4: **Oracle headroom in current candidate pools.** Scores are mIoU (%). $\Delta_{\text{sel}} = \text{SeTeR} - \text{TimeLens2}$ and $\Delta_{\text{orc}} = \text{Oracle} - \text{TimeLens2}$. Larger raw pools are reported in Appendix B.2.

| Dataset | TimeLens2 | SeTeR | Oracle | $\Delta_{\text{sel}}$ | $\Delta_{\text{orc}}$ |
|---|---|---|---|---|---|
| Charades-STA | 58.60 | 59.82 | **64.48** | +1.22 | +5.88 |
| ActivityNet | 58.60 | 60.18 | **65.55** | +1.58 | +6.95 |
| QVHighlights | 70.20 | 71.55 | **77.10** | +1.35 | +6.90 |

**Available and realized improvement.** Table 4 shows oracle gains of **5.88**, **6.95**, and **6.90** mIoU points over TimeLens2. The gaps above SeTeR remain 4.66, 5.37, and 5.55 points. Together with the lower scores under score argmax, these results identify revision assessment as a central step in converting candidate diversity into grounding gains.

### 4.6 Revision Behavior, Ablations, and Cost

**Verifier and policy ablations.** Table 5 compares verifier thresholding, direct regression of $\Delta$, greedy selection with SeTeR's edit priority, and the full policy. Verifier thresholding exceeds TimeLens2 by 0.43 mIoU points. Direct gain regression adds another 0.62 mIoU and 0.65 R@0.7, supporting the value of learning relative improvement beyond a verifier score. The three revision signals with greedy selection further improve over direct regression by 0.10 and 0.58 points. The full learned policy adds 0.20 mIoU and 0.14 R@0.7 over greedy selection, supporting learned edit and stopping decisions. Together, revision assessment and policy learning yield gains of **1.35 mIoU** and **2.27 R@0.7** points over TimeLens2.

Table 5: **Verifier and policy ablations on QVHighlights.** Scores are percentages (↑). Revision variants share initial predictions and candidate pools.

| Method | mIoU | R@0.7 |
|---|---|---|
| TimeLens2 | 70.20 | 61.20 |
| Verifier with tuned threshold | 70.63 | 62.10 |
| Direct gain regression | 71.25 | 62.75 |
| SeTeR with greedy selection | 71.35 | 63.33 |
| **SeTeR (full)** | **71.55** | **63.47** |

**Actual boundary revisions.** SeTeR revises the predicted boundaries for 23.7%, 17.4%, and 23.1% of queries on QVHighlights, Charades-STA, and ActivityNet, respectively. Improvements outnumber degradations on all three datasets: 200 versus 122, 327 versus 208, and 527 versus 395. Contraction is most frequent on QVHighlights and ActivityNet, whereas expansion dominates Charades-STA. Appendix C.2 provides outcomes by operation, duration, content, and initial prediction quality.

**Inference cost.** On two NVIDIA A100-SXM4 80 GB GPUs, SeTeR adds 1 h 03 min to the baseline evaluation time of 6 h 20 min, corresponding to 16.6% overhead (Appendix C.3).

## 5 Conclusion

We introduced SeTeR, a selective temporal reinforcement learning framework that connects revision value to editing decisions. A learned verifier assesses both benefit and risk by predicting improvement probability, degradation risk, and positive gain relative to the current complete answer. A reinforcement learning policy uses these signals to select boundary or split edits and decide when to stop, optimizing grounding gains while penalizing harmful and unnecessary changes.

SeTeR improves all twelve grounding metrics across three TimeLens benchmarks and transfers from QVHighlights to OMTG, where it improves event count accuracy by 8.12 points and F1 at the event level by 5.85 points without training on OMTG. Future work will explore adaptive allocation of local observations, richer compositions of temporal edits, and transfer across grounders and video domains.

## AI USE STATEMENT

In preparing this manuscript, the authors used generative AI tools for language editing and drafting assistance. All text produced with AI assistance was reviewed and revised by the authors, who take responsibility for the final content. Any additional uses of generative AI tools will be disclosed in the final submission in accordance with the ICLR 2027 AI Policy for Authors.

## REFERENCES


Xiang An, Yin Xie, Feilong Tang, Yunyao Yan, Huajie Tan, Didi Zhu, Changrui Chen, Xiuwei Zhao, Bin Qin, Kaicheng Yang, Yifei Shen, Yuanhan Zhang, Kaichen Zhang, Wenkang Zhang, Zheng Cheng, Nansen Zhang, Chunsheng Wu, Chunjiang Ge, Zimin Ran, Dehua Song, Chunyuan Li, Shikun Feng, Ming Hu, Zhangquan Chen, Junbo Niu, Bo Li, Ziyong Feng, Ziwei Liu, Zongyuan Ge, and Jiankang Deng. LLaVA-OneVision-2: Towards Next-Generation Perceptual Intelligence. *arXiv preprint arXiv:2605.25979*, 2026. URL https://arxiv.org/abs/2605.25979.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Zhuo Cao, Heming Du, Bingqing Zhang, Xin Yu, Xue Li, and Sen Wang. When One Moment Isn't Enough: Multi-Moment Retrieval with Cross-Moment Interactions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/faaa187af979e46dc54f4d43f8b815b2-Abstract-Conference.html.

Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. TimeMarker: A Versatile Video-LLM for Long and Short Video Understanding with Superior Temporal Localization Ability. *arXiv preprint arXiv:2411.18211*, 2024. URL https://arxiv.org/abs/2411.18211.

Andong Deng, Zhongpai Gao, Anwesa Choudhuri, Benjamin Planche, Meng Zheng, Bin Wang, Terrence Chen, Chen Chen, and Ziyan Wu. Seq2Time: Sequential Knowledge Transfer for Video LLM Temporal Grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13766–13775, 2025. URL https://arxiv.org/abs/2411.16932.

Yang Ding, Xin Lai, Yizhen Zhang, Wei Li, Ruihang Chu, and Yujiu Yang. VideoZoomer: Reinforcement-Learned Temporal Focusing for Long Video Reasoning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=ARHCFvgx6G.

Dazhao Du, Liao Duan, Jian Liu, Tao Han, Yujia Zhang, Eric Liu, Xi Chen, and Song Guo. MLLMs Know When Before Speaking: Revealing and Recovering Temporal Grounding via Attention Cues. *arXiv preprint arXiv:2605.21954*, 2026. URL https://arxiv.org/abs/2605.21954.

Wei Feng, Xin Wang, Hong Chen, Yu-Wei Zhan, Zihan Song, Bin Huang, Kecheng Zheng, and Wenwu Zhu. Reflective cross-granularity grounding with preference optimization for long video understanding. In *Proceedings of the International Conference on Multimedia Retrieval*, 2026a. doi: 10.1145/3805622.3810717.

Wei Feng, Xin Wang, Yu-Wei Zhan, Yuwei Zhou, and Wenwu Zhu. Modularized dynamic-granularity video llm for multi-event long video understanding. *arXiv preprint arXiv:2607.15778*, 2026b. URL https://arxiv.org/abs/2607.15778.

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 (ICCV)*, 2017. URL https://arxiv.org/abs/1705.02101.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A Deep Neural Network with an Integrated Reject Option. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 2151–2159, 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Gemini Team. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. Technical report, Google, 2025. URL https://storage.googleapis.com/deepmind-media/gemini/gemini_v2_5_report.pdf.

Yongxin Guo, Jingyu Liu, Mingda Li, Dingxin Cheng, Xiaoying Tang, Dianbo Sui, Qingbin Liu, Xi Chen, and Kevin Zhao. VTG-LLM: Integrating Timestamp Knowledge into Video LLMs for Enhanced Video Temporal Grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3302–3310, 2025a. doi: 10.1609/aaai.v39i3.32341.

Yongxin Guo, Jingyu Liu, Mingda Li, Qingbin Liu, Xi Chen, and Xiaoying Tang. TRACE: Temporal Grounding Video LLM via Causal Event Modeling. In *International Conference on Learning Representations (ICLR)*, 2025b. URL https://arxiv.org/abs/2410.05643.

Tanveer Hannan, Md Mohaiminul Islam, Jindong Gu, Thomas Seidl, and Gedas Bertasius. ReVisionLLM: Recursive Vision-Language Model for Temporal Grounding in Hour-Long Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2411.14901.

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 (CVPR)*, pages 961–970, 2015. URL https://openaccess.thecvf.com/content_cvpr_2015/html/Heilbron_ActivityNet_A_Large-Scale_2015_CVPR_paper.html.

Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. VTimeLLM: Empower LLM to Grasp Video Moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14271–14280, 2024a. URL https://arxiv.org/abs/2311.18445.

De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. LITA: Language Instructed Temporal-Localization Assistant. In *European Conference on Computer Vision (ECCV)*, 2024b. URL https://arxiv.org/abs/2403.19046.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large Language Models Cannot Self-Correct Reasoning Yet. In *International Conference on Learning Representations (ICLR)*, 2024c. URL https://arxiv.org/abs/2310.01798.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-Captioning Events in Videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 706–715, 2017. URL https://arxiv.org/abs/1705.00754.

Xiaohan Lan, Yitian Yuan, Xin Wang, Zhi Wang, and Wenwu Zhu. A Survey on Temporal Sentence Grounding in Videos. *ACM Trans. Multimedia Comput. Commun. Appl.*, 19(2):51:1–51:33, 2023. doi: 10.1145/3532626.

Jie Lei, Tamara L. Berg, and Mohit Bansal. Detecting Moments and Highlights in Videos via Natural Language Queries. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 11846–11858, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/62e0973455fd26eb03e91d5741a4a3bb-Abstract.html.

Zeqian Li, Shangzhe Di, Zhonghua Zhai, Weilin Huang, Yanfeng Wang, and Weidi Xie. Universal Video Temporal Grounding with Generative Multi-modal Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. URL https://arxiv.org/abs/2506.18883.

Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. UniVTG: Towards Unified Video-Language Temporal Grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2794–2804, 2023. URL https://arxiv.org/abs/2307.16715.

Ye Liu, Jixuan He, Wanhua Li, Junsik Kim, Donglai Wei, Hanspeter Pfister, and Chang Wen Chen. $R^2$-Tuning: Efficient Image-to-Video Transfer Learning for Video Temporal Grounding. In *European Conference on Computer Vision (ECCV)*, 2024. URL https://arxiv.org/abs/2404.00801.

Ye Liu, Kevin Qinghong Lin, Chang Wen Chen, and Mike Zheng Shou. VideoMind: A Chain-of-LoRA Agent for Temporal-Grounded Video Reasoning. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2503.13444.

Fuwen Luo, Shengfeng Lou, Chi Chen, Ziyue Wang, Chenliang Li, Weizhou Shen, Jiyue Guo, Peng Li, Ming Yan, Ji Zhang, Fei Huang, and Yang Liu. MUSEG: Reinforcing Video Temporal Understanding via Timestamp-Aware Multi-Segment Grounding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 35549–35561, 2026. doi: 10.18653/v1/2026.acl-long.1644.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. *arXiv preprint arXiv:2303.17651*, 2023. URL https://arxiv.org/abs/2303.17651.

Hussein Mozannar and David Sontag. Consistent Estimators for Learning to Defer to an Expert. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 7076–7087, 2020. URL https://proceedings.mlr.press/v119/mozannar20b.html.

OpenAI. GPT-5 System Card. OpenAI, 2025. URL https://openai.com/index/gpt-5-system-card/.

Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing Video Large Language Model with Fine-Grained Temporal Reasoning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pages 41340–41356, 2024. URL https://proceedings.mlr.press/v235/qian24a.html.

Qwen Team. Qwen3.5: Towards Native Multimodal Agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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 (CVPR)*, 2024. URL https://arxiv.org/abs/2312.02051.

Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *European Conference on Computer Vision (ECCV)*, 2016. URL https://arxiv.org/abs/1604.01753.

Haibo Wang, Zhiyang Xu, Yu Cheng, Shizhe Diao, Yufan Zhou, Yixin Cao, Qifan Wang, Weifeng Ge, and Lifu Huang. Grounded-VideoLLM: Sharpening Fine-grained Temporal Grounding in Video Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 959–975, 2025a. doi: 10.18653/v1/2025.findings-emnlp.50.

Xizi Wang, Feng Cheng, Ziyang Wang, Huiyu Wang, Md Mohaiminul Islam, Lorenzo Torresani, Mohit Bansal, Gedas Bertasius, and David Crandall. TimeRefine: Temporal Grounding with Time Refining Video LLM. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026. URL https://arxiv.org/abs/2412.09601.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025b. URL https://arxiv.org/abs/2503.13377.

Yueqian Wang, Xiaojun Meng, Jianxin Liang, Yuxuan Wang, Qun Liu, and Dongyan Zhao. Hawk-Eye: Training Video-Text LLMs for Grounding Text in Videos. *arXiv preprint arXiv:2403.10228*, 2024. URL https://arxiv.org/abs/2403.10228.

Ronald J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13754–13765, 2025. URL https://arxiv.org/abs/2411.10332.

Qi Xu, Yue Tan, Shihao Chen, Jiahao Meng, Anna Wang, Shunping Ji, Hao Fei, and Jason Li. Towards One-to-Many Temporal Grounding. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2606.06294.

Ziang Yan, Yinan He, Xinhao Li, Zhengrong Yue, Xiangyu Zeng, Yali Wang, Yu Qiao, Limin Wang, and Yi Wang. VideoChat-R1.5: Visual Test-Time Scaling to Reinforce Multimodal Reasoning by Iterative Perception. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. URL https://arxiv.org/abs/2509.21100.

Ziang Yan, Sheng Xia, Jiashuo Yu, Yue Wu, Tianxiang Jiang, Songze Li, Kanghui Tian, Yicheng Xu, Yinan He, Kai Chen, Limin Wang, Yu Qiao, and Yi Wang. InternVideo3: Agentify Foundation Models with Multimodal Contextual Reasoning. *arXiv preprint arXiv:2606.12195*, 2026. URL https://arxiv.org/abs/2606.12195.

Xiangyu Zeng, Kunchang Li, Chenting Wang, Xinhao Li, Tianxiang Jiang, Ziang Yan, Songze Li, Yansong Shi, Zhengrong Yue, Yi Wang, Yali Wang, Yu Qiao, and Limin Wang. TimeSuite: Improving MLLMs for Long Video Understanding via Grounded Tuning. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.19702.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based Localizing Network for Natural Language Video Localization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6543–6554, 2020a. doi: 10.18653/v1/2020.acl-main.585.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Temporal Sentence Grounding in Videos: A Survey and Future Directions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(8):10443–10465, 2023. doi: 10.1109/TPAMI.2023.3258628.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, Ying Shan, and Limin Wang. TimeLens: Rethinking Video Temporal Grounding with Multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. URL https://arxiv.org/abs/2512.14698.

Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2D Temporal Adjacent Networks for Moment Localization with Natural Language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12870–12877, 2020b. doi: 10.1609/aaai.v34i07.6984.

Yuhan Zhu, Changlian Ma, Xiangyu Zeng, Xinhao Li, Zhiqiu Zhang, Songze Li, Jun Zhang, Tianxiang Jiang, Yuandong Yang, Ziang Yan, Zikang Wang, Xinyu Chen, Haoran Chen, Shaowei Zhang, and Limin Wang. TimeLens2: Generalist Video Temporal Grounding with Multimodal LLMs. *arXiv preprint arXiv:2607.17423*, 2026. URL https://arxiv.org/abs/2607.17423.

# SUPPLEMENTARY MATERIAL

Appendix A describes the refinement components and evaluation setup. Appendix B analyzes candidate selection, Appendix C reports additional results and revision behavior, and Appendix D presents qualitative examples.

## A IMPLEMENTATION AND EVALUATION DETAILS

### A.1 GROUNDER AND VIDEO INPUT

SeTeR combines a frozen TimeLens2-8B grounder with a supervised revision verifier and a reinforcement learning policy for edit selection and stopping. The grounder supplies both the initial prediction and the local alternatives. Local timestamps are mapped to the full video timeline, and exact duplicates are removed when constructing the proposal set.

**Global observation on QVHighlights.** We decode videos with `decord` and sample frames uniformly along the timeline at 2 FPS. The frame count is constrained to a multiple of two, with a minimum of 4 and a maximum of 768 frames. A video of 150 seconds therefore supplies approximately 300 frames. Sampling is deterministic, so repeated processing of the same video selects the same temporal positions. The sampled frames are supplied jointly to the frozen grounder to predict the initial interval $[s, e]$.

**Global localization prompt.** For a query $\{\texttt{query}\}$, we use the following prompt:

```
Please find the visual event described by the sentence
'{query}', determining its starting and ending times.
The format should be:
'The event happens in <start time> - <end time> seconds'.
```

**Local candidates.** For each current interval, we generate four local predictions in its observation window. Each prediction may contain one or more intervals. Local timestamps are clipped to the crop range, intervals are sorted by their start times, and intervals with zero duration, reversed endpoints, or invalid bounds are removed. A candidate is retained only if at least one valid interval remains. We then convert its timestamps to the full video timeline. Among candidates targeting the same current interval, only predictions with exactly identical interval sets are deduplicated. We use neither nonmaximum suppression nor merging of nearby candidates, leaving at most four valid candidates per current interval.

### A.2 EVALUATION SETUP

The main comparison uses the TimeLens reannotations of Charades-STA, ActivityNet Captions, and QVHighlights. Our local evaluation of TimeLens2 yields mIoU scores of 58.58, 58.27, and 70.01 on Charades-STA, ActivityNet, and QVHighlights, respectively. These are lower than the corresponding scores of 58.60, 58.60, and 70.20 reported by Zhu et al. (2026). For a fair comparison, we use the higher reported scores as the TimeLens2 baseline. For transfer, the refinement framework trained on QVHighlights is applied directly to OMTG.

### A.3 TRAINING UTILITY AND SUPERVISION

**Task utility.** For a single moment, the utility is $M(S, G) = \mathrm{IoU}(S_1, G_1)$. For multiple moments, we combine temporal coverage, event matching, and count accuracy:

$$M(S, G) = 0.5\ \mathrm{SetIoU}(S, G) + 0.3\ \mathrm{tF1}_{0.5}(S, G) + 0.2\ \mathrm{CAcc}(S, G). \tag{A.1}$$

Here, $\mathrm{SetIoU}$ is the intersection over union of the union of predicted intervals and the union of reference intervals. For $\mathrm{tF1}_{0.5}$, we form a bipartite graph connecting predicted and reference intervals whose temporal IoU is at least 0.5, and find a matching with the maximum number of pairs. If this number is $m$, then $\mathrm{tF1}_{0.5} = 2m/(|S| + |G|)$. Count accuracy is $\mathrm{CAcc}(S, G) = \mathbf{1}[|S| = |G|]$. All three components and the resulting utility use the $[0, 1]$ scale.

**Relative labels.** Each candidate is supervised by $\Delta(e; S, G) = M(S_e, G) - M(S, G)$. We set both classification margins to 0.01: an edit is labeled as improving when $\Delta > 0.01$ and degrading when $\Delta < -0.01$. The gain target is $\max(\Delta, 0)$, as in Eq. (10). Edits within the two margins receive zero labels for both classification heads.

**Initial and edited states.** Half of the supervised fine tuning (SFT) samples are drawn from the complete candidate collections at initial prediction states. The other half come from states obtained by randomly applying zero to two legal edits. At each resulting state, we recompute the utility differences and labels of the remaining legal edits relative to the updated full interval set. Allowing zero edits also includes initial states in this second source. Sampling is stratified by the improvement and degradation labels so that, for each classification target, negative examples number at most three times the positive examples. This training exposes the evaluator to the changing context in which the revision policy operates.

### A.4 Evaluator and policy architecture

**Revision evaluator.** SFT trains the revision evaluator using the utility difference in Eq. (7) and the objective in Eq. (11). Its inputs are the query, local video frames, the complete current interval set, the complete set after the proposed edit, the interval being replaced, and the crop position. A frozen TimeLens2 visual language backbone encodes these inputs with an evaluator specific LoRA adapter. The representation of the last valid token feeds three independent prediction heads, each a two layer MLP. The improvement and degradation heads use sigmoid outputs, and the positive gain head uses a softplus output.

Only the LoRA parameters and the three prediction heads are trained in this stage. LoRA has rank 16, scaling parameter $\alpha = 32$, and dropout 0.05, and is applied to the language layers' `q_proj`, `k_proj`, `v_proj`, and `o_proj` modules. The degradation BCE term has weight 2, while the positive gain head is supervised by the regression term in Eq. (11). Global and local interval generation uses the original frozen grounder; the trained adapter serves revision evaluation.

**Selection policy.** For each candidate edit, the policy receives the candidate embedding from the SFT evaluator together with $p_{\text{imp}}$, $p_{\text{deg}}$, $g_+$, and $A(e)$. Temporal features describe the interval being edited, the proposed boundaries, the relative change in duration, the change in interval count, and the number of intervals in the current set. Additional features describe overlap and duplication between the candidate and the other intervals. These inputs connect the predicted revision effects to the geometry of the evolving answer.

The policy uses a Set Transformer with two layers, a hidden size of 512, eight attention heads, and dropout of 0.1. A pointer head scores the candidate edits, while a separate STOP token represents termination. A learned value head supplies the baseline for policy gradient training.

### A.5 TRAINING THE SELECTIVE TEMPORAL POLICY

**Behavior cloning.** We initialize the policy for two epochs at a learning rate of $10^{-4}$. At each training state, the target is the edit with the largest reference utility gain when the maximum predicted priority $\max_e A(e)$ exceeds 0.01 and that edit has positive gain. Otherwise, the target is STOP. This supervision uses the learned priorities to gate revision and reference gains to choose the target edit.

**Episodes and rewards.** Each episode starts from the global prediction $S^{(0)}$. The evaluator scores the available candidates, and the policy selects an edit or STOP. After an edit is applied, all remaining candidates are evaluated again relative to the updated interval set. An episode accepts at most three edits. The immediate reward is given in Eq. (17), with a degradation coefficient of 1.0 and a cost of 0.005 for each accepted edit. At termination, the episode receives an additional reward:

$$r_{\text{terminal}} = 0.5\big(M(S^{(T)}, G) - M(S^{(0)}, G)\big). \tag{A.2}$$

We optimize the policy objective in Eq. (18) with REINFORCE and a learned value baseline. With a discount factor of 1.0, the return from a decision is the sum of its remaining immediate rewards and the terminal reward, which is counted once. Subtracting the predicted state value gives the advantage used for the policy update. The value head learns to predict this return; generalized advantage estimation is not used. Table A.1 lists the optimization settings. The entropy coefficient decreases linearly from 0.01 to 0.001 over reinforcement learning.

Table A.1: **Policy training configuration.** Behavior cloning precedes REINFORCE.

| Setting | Value |
|---|---|
| Algorithm | REINFORCE with a learned value baseline |
| Discount factor | 1.0 |
| Advantage estimate | Return minus predicted state value |
| Policy learning rate | $5 \times 10^{-5}$ |
| Value head learning rate | $10^{-4}$ |
| Optimizer | AdamW |
| Weight decay | 0.01 |
| Entropy coefficient | 0.01 to 0.001, linear decay |
| Value loss weight | 0.5 |
| Gradient clipping | 1.0 |
| Episodes per update | 64 |
| Maximum accepted edits | 3 |
| Reinforcement learning epochs | 5 |
| Precision | BF16 |
| Behavior cloning epochs | 2 |
| Behavior cloning learning rate | $10^{-4}$ |

**Inference.** SeTeR constructs local proposals from the global prediction and evaluates their effects on the current interval set. After applying an accepted edit, it updates the set and reassesses all remaining proposals. The procedure returns the current answer when the policy selects STOP, no proposal exceeds the revision threshold, or three edits have been accepted. The inference environment and computational cost are reported in Appendix C.3.

# B Additional Candidate Selection Analyses

## B.1 Detailed outcomes of ungated selection

Table B.1 expands the verifier score argmax comparison in Table 3 with query counts and revision outcomes. Improved and Degraded count queries whose IoU increases or decreases after selection, respectively. Score argmax degrades more queries than it improves on all three datasets, motivating supervision of the relative value of a revision.

Table B.1: **Detailed ungated candidate selection.** Scores are mIoU (%); $\Delta = \text{Score argmax} - \text{TimeLens2}$.

| | Charades-STA | ActivityNet | QVHighlights |
|---|---|---|---|
| TimeLens2 | 58.60 | 58.60 | 70.20 |
| Score argmax | 54.53 | 54.27 | 64.34 |
| $\Delta$ mIoU | $-4.07$ | $-4.33$ | $-5.86$ |
| Queries | 3,363 | 4,500 | 1,541 |
| Improved | 810 | 982 | 299 |
| Degraded | 1,247 | 1,490 | 939 |

**Additional recall results.** Recall follows the same pattern. On ActivityNet, score argmax obtains 71.58 R@0.3, 60.82 R@0.5, and 40.13 R@0.7, compared with 74.00, 65.60, and 46.60 for TimeLens2. On QVHighlights, selection degrades 939 queries and improves 299. These observations support evaluating a proposed revision against the prediction it would replace.

## B.2 Oracle results for larger raw candidate pools

Table B.2 evaluates the correction opportunities in larger raw candidate pools. Oracle selection exceeds TimeLens2 by 12.61, 11.55, and 7.65 mIoU points on Charades-STA, ActivityNet, and QVHighlights, respectively.

Table B.2: **Oracle results for larger raw candidate pools.** Scores are mIoU (%); $\Delta_{\text{orc}} = \text{Oracle} - \text{TimeLens2}$.

| Dataset | TimeLens2 | Oracle | $\Delta_{\text{orc}}$ |
|---|---|---|---|
| Charades-STA | 58.60 | **71.21** | $+12.61$ |
| ActivityNet | 58.60 | **70.15** | $+11.55$ |
| QVHighlights | 70.20 | **77.85** | $+7.65$ |

Both the raw pools here and the current pools in Table 4 attain oracle scores substantially above TimeLens2. This availability of useful alternatives motivates learning which revisions to integrate into the final prediction.

## B.3 Interpretation of the oracle bound

The oracle in Eq. (20) defines the highest mIoU attainable by selecting from the given candidate pools. For a selector $h$ with $h(i) \in \mathcal{C}_i$, the per-query inequality $u_i(h(i)) \leq \max_{c \in \mathcal{C}_i} u_i(c)$ gives the bound after averaging over queries. The difference between oracle and selected scores therefore measures the opportunity for improving selection from those pools.

# C Additional Experiments

## C.1 Additional evaluation on QV-M$^2$

Table C.1 reports results on the QV-M$^2$ test set. SeTeR increases GmAP from 22.85 to 25.63 and set-IoU from 0.4762 to 0.4929, providing additional evidence of improved grounding of multiple moments.

Table C.1: **Results on the QV-M$^2$ test set.** GmAP is reported as a percentage; set-IoU is on the $[0, 1]$ scale.

| Metric | Original | SeTeR |
|---|---|---|
| GmAP | 22.85 | **25.63** |
| set-IoU | 0.4762 | **0.4929** |

## C.2 Revision behavior and reliability

**Boundary revision outcomes.** A boundary revision changes at least one endpoint of the original interval. Table C.2 reports candidate changes, the subset with identical endpoints, and the resulting boundary revisions. Improved, degraded, and equal outcomes compare the final and original IoU for each revised query.

Table C.2: **Candidate changes and boundary revisions.** Actual revisions equal candidate changes minus identical intervals. Revision rates use all queries; outcome counts describe the revised queries.

| Dataset | Queries | Candidate changes | Identical intervals | Actual revisions (rate) | Improved / degraded / equal |
|---|---|---|---|---|---|
| QVHighlights | 1,541 | 543 | 178 | 365 (23.7%) | 200 / 122 / 43 |
| Charades-STA | 3,363 | 1,392 | 808 | 584 (17.4%) | 327 / 208 / 49 |
| ActivityNet | 4,500 | 1,575 | 536 | 1,039 (23.1%) | 527 / 395 / 117 |

Improvements outnumber degradations on all three datasets. Among revised queries, the degradation rates are 33.4%, 35.6%, and 38.0% on QVHighlights, Charades-STA, and ActivityNet, respectively; these correspond to 7.9%, 6.2%, and 8.8% of all queries. The equal IoU category records revised intervals whose overlap score is unchanged.

**Revision outcomes for accurate initial predictions.** Table C.3 groups revised queries whose initial IoU is at least 0.9. The degradation rates in this group are 55.3%, 73.2%, and 63.7%, exceeding the corresponding rates across all revisions. This pattern highlights the role of initial answer quality in revision assessment and motivates conditioning edit acceptance on the current hypothesis.

Table C.3: **Revision outcomes for initial IoU $\geq 0.9$.** Percentages use revised queries in this group as the denominator.

| Dataset | Revised | Improved | Degraded | Equal IoU | Degraded (%) |
|---|---|---|---|---|---|
| QVHighlights | 161 | 57 | 89 | 15 | 55.3 |
| Charades-STA | 123 | 32 | 90 | 1 | 73.2 |
| ActivityNet | 355 | 109 | 226 | 20 | 63.7 |

### C.2.1 BOUNDARY OPERATIONS AND CHANGES IN DURATION

Tables C.4 and C.5 characterize revisions by boundary operation and change in interval duration. The operation categories are expansion, contraction, and shift. A shift can also lengthen or shorten an interval, so the duration breakdown groups edits directly by their final length relative to the original.

Table C.4: **Outcomes by boundary operation.** Shares use actual revisions within each dataset as the denominator. Improved, degraded, and equal refer to IoU changes.

| Dataset | Operation | Count | Share (%) | Improved | Degraded | Equal |
|---|---|---|---|---|---|---|
| QVHighlights | Expansion | 78 | 21.4 | 54 | 20 | 4 |
| | Contraction | 208 | 57.0 | 120 | 82 | 6 |
| | Shift | 79 | 21.6 | 26 | 20 | 33 |
| Charades-STA | Expansion | 302 | 51.7 | 178 | 96 | 28 |
| | Contraction | 224 | 38.4 | 112 | 108 | 4 |
| | Shift | 58 | 9.9 | 37 | 4 | 17 |
| ActivityNet | Expansion | 304 | 29.3 | 139 | 143 | 22 |
| | Contraction | 541 | 52.1 | 312 | 206 | 23 |
| | Shift | 194 | 18.7 | 76 | 46 | 72 |

Table C.5: **Changes in interval duration among boundary revisions.** Unchanged duration corresponds to a translation of the interval at fixed length.

| Dataset | Longer | Shorter | Unchanged duration |
|---|---|---|---|
| QVHighlights | 100 | 211 | 54 |
| Charades-STA | 330 | 231 | 23 |
| ActivityNet | 378 | 571 | 90 |

**Operation frequency and net benefit.** QVHighlights and ActivityNet favor contraction by count, while Charades-STA favors expansion. The net IoU contribution combines the frequency of an operation with the magnitude of its gains and losses. On QVHighlights, 54 of 78 expansions improve IoU, and longer intervals contribute approximately four times as much net gain as shorter intervals. On Charades-STA, expansion accounts for nearly all net gain, while contraction is approximately neutral or slightly negative. On ActivityNet, contraction improves more queries than it harms, and longer intervals contribute roughly three quarters of the net IoU gain. Across these datasets, extending temporal coverage is a substantial source of improvement alongside the removal of excess context.

### C.2.2 VARIATION WITH EVENT DURATION AND CONTENT

Table C.6 reports boundary revision rates within each event duration group. Very short events are revised less frequently. For Charades-STA and ActivityNet, rates increase across the duration bins, reaching 33.0% and 32.6% for events lasting 10 to 30 seconds. QVHighlights shows the same broad pattern, with 14.9% for events of at most two seconds and approximately 25% for the pooled 2 to 30 second group.

Table C.6: **Boundary revision rates by event duration (%).** Rates use queries within each duration group. QVHighlights pools the 2 to 30 second range; dashes indicate unavailable entries.

| Reported duration range | QVHighlights | Charades-STA | ActivityNet |
|---|---|---|---|
| $\leq 2$ s | 14.9 | 8.6 | 12.0 |
| 2 to 5 s | — | 17.8 | 18.7 |
| 5 to 10 s | — | 23.3 | 28.8 |
| 10 to 30 s | — | 33.0 | 32.6 |
| 2 to 30 s (pooled) | $\approx 25$ | — | — |

Table C.7: **Boundary revision rates by content group.** Rates are percentages of queries within each listed group.

| Dataset | Content group | Revision rate (%) |
|---|---|---|
| QVHighlights | Static postures / display | 30.8 |
| | Speech / social interaction | 27.7 |
| | Sports / exercise | 26.5 |
| | Movement / entering and leaving | 15.9 |
| Charades-STA | Static postures / display | 23.0 |
| | Eating | 18.0 |
| | Object manipulation | 15.2 |
| ActivityNet | Eating / cooking | 26.2 |
| | Object manipulation | 26.1 |
| | Movement | 25.4 |
| | Sports / exercise | 24.2 |

Revision frequency also varies with event content. In QVHighlights, static postures and display are revised more often than movement or entering and leaving. Static postures and display have the highest rate among the listed Charades-STA groups as well. ActivityNet shows a narrower range across the listed content groups (24.2 to 26.2%) than across duration groups (12.0 to 32.6%). Together, these breakdowns characterize how SeTeR allocates revisions across temporal scales and event categories.

### C.3 Inference environment and runtime

Table C.8 summarizes the inference environment and evaluation time on two GPUs. TimeLens2-8B returns one interval per video; SeTeR additionally generates candidate intervals, verifies the corresponding clips, and selects a final prediction.

Table C.8: **Inference environment and runtime.** Durations are wall clock estimates for the two GPU setup. Stage times give the additional cost over the baseline.

| *Inference environment* | |
|---|---|
| GPUs | 2× NVIDIA A100-SXM4 (80 GB each) |
| Numerical precision | BF16 |
| Attention implementation | FlashAttention 2 |
| Inference batch size | 1 |
| DataLoader workers | 4 |
| *Total evaluation time* | |
| TimeLens2-8B (one interval per video) | 6 h 20 min |
| SeTeR (generation, verification, and selection) | 7 h 23 min |
| Additional time | 1 h 03 min (16.6%) |
| *Additional time by stage* | |
| Additional candidate generation | 28 min |
| Verification of candidate clips | 33 min |
| Final selection | 1 to 2 min |

SeTeR adds 63 minutes to the baseline evaluation time of 380 minutes, corresponding to 16.6% overhead. Additional candidate generation takes 28 minutes and verification takes 33 minutes, accounting for most of the added cost. Final selection takes 1 to 2 minutes, making the learned decision stage a small part of the overall computation.

# D Qualitative Examples

## D.1 Successful single moment refinements

Figure D.1 illustrates three successful refinements of single moment predictions. In QVHighlights, SeTeR contracts an overlong prediction around the writing event. In Charades-STA, it extends a prematurely ending prediction to cover the opening of the pantry door. In ActivityNet Captions, it removes excess trailing coverage from the prediction of wiping a car roof. The tIoU increases from 0.500, 0.333, and 0.556 to 1.000, respectively. These examples show how local evidence supports both removing excess context and recovering omitted parts of an event.

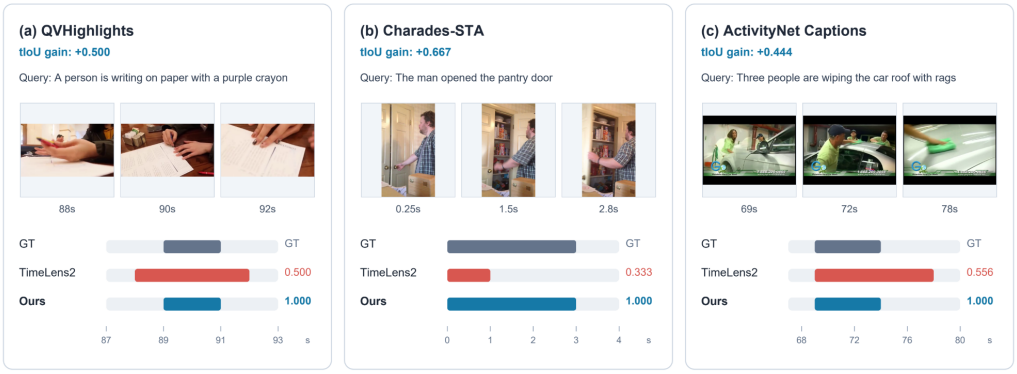


Figure D.1: **Qualitative evolution of temporal predictions.** Examples from (a) QVHighlights, (b) Charades-STA, and (c) ActivityNet Captions. Gray denotes ground truth (GT), red TimeLens2, and blue SeTeR. Gains are per-query tIoU differences on the $[0, 1]$ scale. Refinement corrects overlong spans and premature endings.

Table D.1 lists the queries and tIoU scores for the three examples in Figure D.1.

Table D.1: **Queries and scores for Figure D.1.** Scores are tIoU on the $[0, 1]$ scale.

| Dataset | Query | Original | SeTeR |
|---|---|---|---|
| QVHighlights | A person is writing on paper with a purple crayon | 0.500 | 1.000 |
| Charades-STA | The man opened the pantry door | 0.333 | 1.000 |
| ActivityNet Captions | Three people are wiping the car roof with rags | 0.556 | 1.000 |

## D.2 GROUNDING MULTIPLE MOMENTS ON OMTG

Figure D.2 presents six OMTG examples. SeTeR recovers the correct interval counts in OMTG 4, 182, and 197, and refines boundaries while preserving the count in OMTG 273. In OMTG 319 and 117, overlap improves while the output represents two occurrences with one interval. These cases illustrate the interaction between boundary refinement and the recovery of event structure.

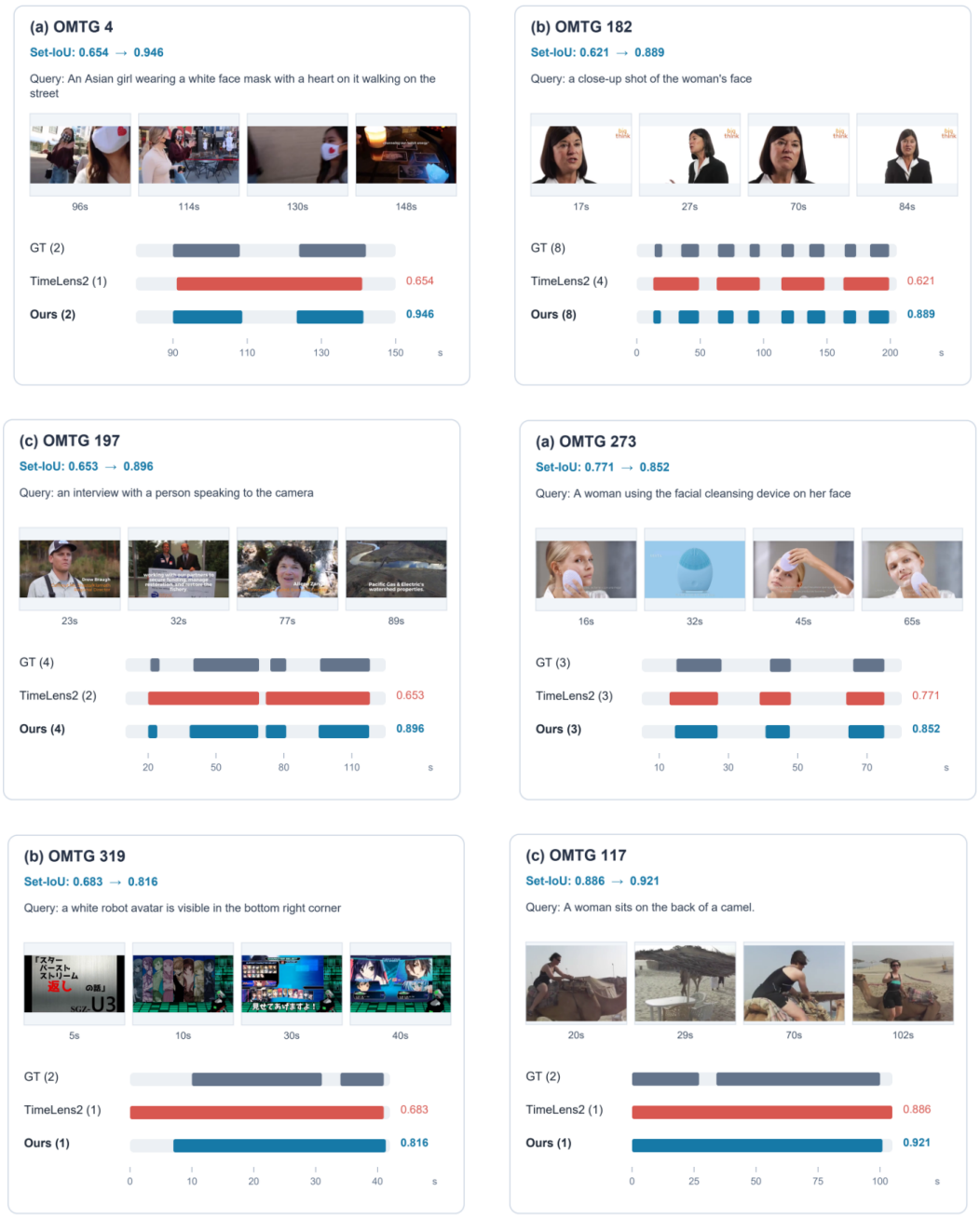


Figure D.2: **Event recovery and boundary refinement on OMTG.** Read left to right, then top to bottom. Gray denotes ground truth, red TimeLens2, and blue SeTeR; parentheses give interval counts. The first three cases recover 2, 8, and 4 intervals. The last two improve Set-IoU while producing one interval for two occurrences. Scores are per-example Set-IoU.

### D.3 Analysis of revision errors

Figure D.3 groups six revision errors across QVHighlights, ActivityNet Captions, and Charades-STA by boundary operation. Expansion includes irrelevant time, contraction removes relevant content, and shifts misalign the prediction. The cases identify concrete cues for improving edit assessment: coverage of the queried event, exclusion of surrounding activity, and alignment of event transitions.

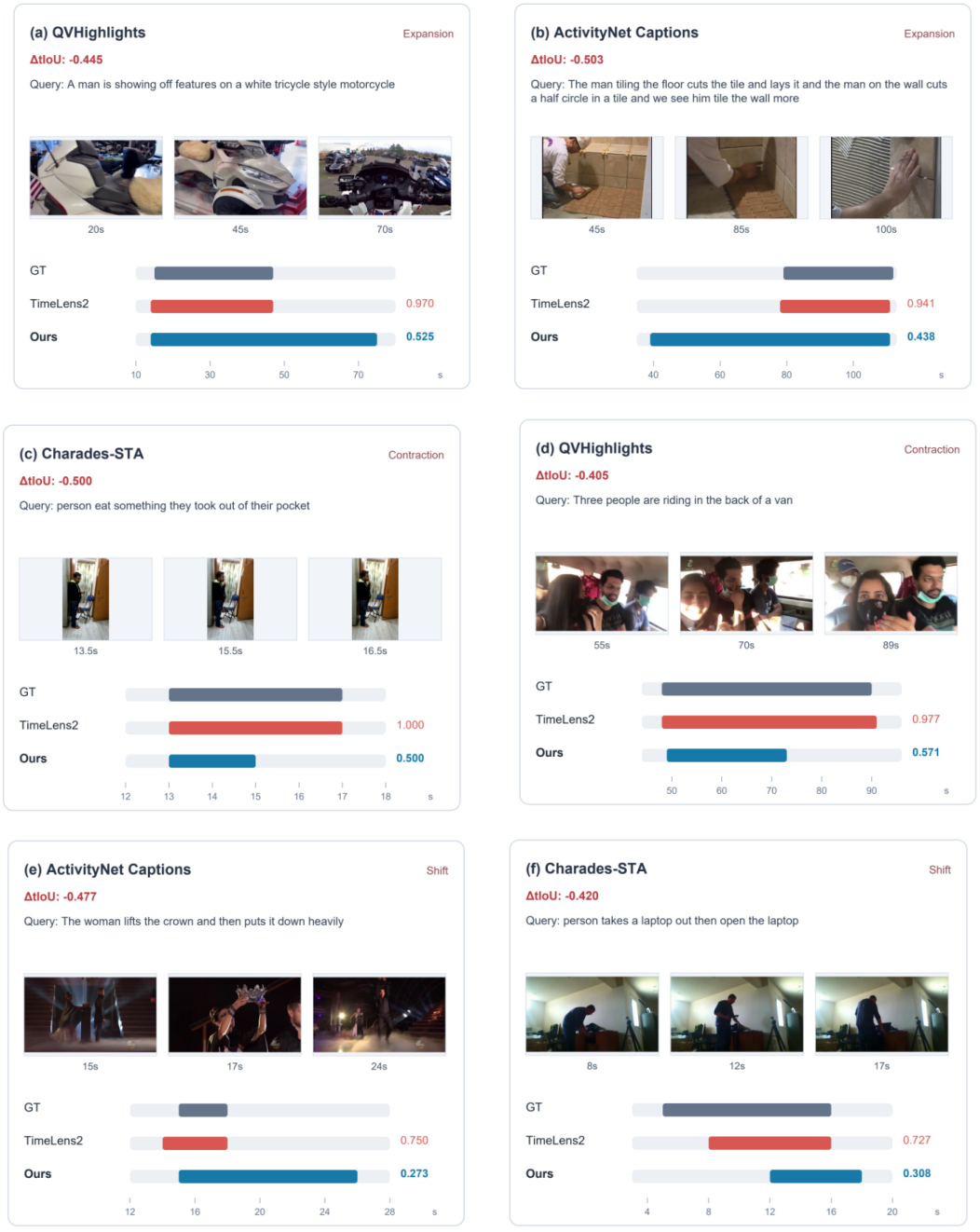


Figure D.3: **Revision errors by boundary operation.** Panels (a,b) show expansion, (c,d) contraction, and (e,f) temporal shifts. Gray denotes ground truth, red TimeLens2, and blue SeTeR. The displayed tIoU and $\Delta$ values quantify the change in overlap for each example.