TSL:LEARNING THE TEMPORAL STRUCTURE OF EVIDENCE VIA STATE FLOW FOR TEMPORAL VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Video temporal understanding integrates visual and linguistic information, requiring models to answer accurately while capturing when events occur and how they relate over time. Semantically related evidence segments form complete evidence chains with temporal structure, yet existing methods that directly predict evidence interval sets inadequately model temporal relations and gaps between segments, leading to missed evidence segments, erroneous splitting, and boundary localization errors. To address these challenges, we propose a method for learning temporal evidence structure through state flow, explicitly modeling the onset, continuation, and termination of query-relevant evidence along the timeline as a learnable evidence-state transition process that organizes local evidence segments into structurally complete chains. Our theoretical analysis establishes a reversible correspondence between evidence interval sets and canonical evidence state–offset representations. To learn this representation, we design a Temporal Structure Learner (TSL) that injects information about temporal evidence structure into video representations between the decoder layers of a multimodal large language model. Through supervised learning, the model predicts legal evidence-state chains and refines the endpoints of predicted segments using offsets, enabling discrete state chains and continuous boundary offsets to jointly determine complete evidence chains. Experimental results demonstrate that our method substantially outperforms existing approaches to video temporal understanding.

1 INTRODUCTION

Multimodal video understanding integrates visual and linguistic information to interpret events in videos and answer natural-language questions (Lin et al., 2024; Li et al., 2024; Song et al., 2024). Video temporal understanding further concerns when events occur and how they relate over time (Krishna et al., 2017; Xiao et al., 2021; Liu et al., 2024), requiring models to answer accurately while localizing the temporal intervals containing supporting evidence (Di & Xie, 2024; Lu et al., 2025), thereby providing a verifiable evidence base for reliable video understanding.

Early studies localize evidence intervals by directly learning relationships between queries and video segments: VSLNet (Zhang et al., 2020a) predicts segment start and end positions, whereas Moment-DETR (Lei et al., 2021) directly predicts sets of intervals. Research subsequently expanded from segment retrieval to video question answering with temporal evidence localization (Di & Xie, 2024; Gupta et al., 2025). NExT-GQA (Xiao et al., 2024) learns evidence–question correspondence through Gaussian masking and cross-modal self-supervision, using the learned visual representations to predict answers; GroundFormer (Seo et al., 2026) incorporates question information into video features and links answer selection to evidence. In recent work, MUSEG (Luo et al., 2026) learns multi-segment localization through phased rewards, while TimeLens (Zhang et al., 2026a) investigates data refinement, timestamp encoding, and reinforcement learning with verifiable rewards.

Despite progress in video temporal grounding, the evidence segments supporting an answer may occur at different times and form complete, temporally structured evidence chains through semantic connections. Methods that directly predict sets of evidence intervals do not adequately model

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.

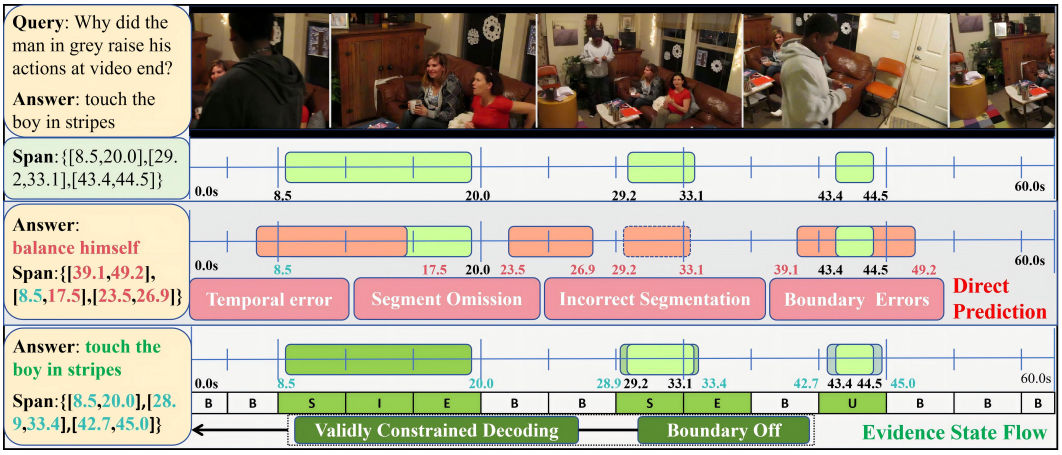

Figure 1: Existing methods that directly predict sets of evidence intervals fail to adequately model temporal relations and gaps between segments, leading to missed evidence segments, erroneous splitting, and boundary localization errors. We propose a method for learning temporal evidence structure through state flow, explicitly modeling the onset, continuation, and termination of query-relevant evidence along the timeline as a learnable evidence-state transition process, thereby organizing local evidence segments into structurally complete evidence chains.

temporal relationships and gaps between segments, which can lead to omitted evidence segments, erroneous splitting, and inaccurate boundaries.

To address these challenges, we propose a method for learning temporal evidence structure through state flow, explicitly modeling the onset, continuation, and termination of query-relevant evidence along the timeline as a learnable state-transition process, thereby organizing local evidence segments into structurally complete evidence chains. Specifically, we assign each video temporal unit one of five states: background, the start, interior, or end of an evidence segment, or an evidence segment contained within a single unit. We impose legal-transition constraints on sequence initiation, termination, and transitions between states to ensure the structural completeness and consistency of evidence-state chains. Our theoretical analysis shows that correctly encoded state chains and exact continuous offsets can losslessly reconstruct the temporal coverage of annotated evidence, establishing a reversible correspondence between evidence intervals and canonical evidence state–offset representations. To learn this representation, we design a Temporal Structure Learner (TSL) that, conditioned on query semantics, injects information about temporal evidence structure into video representations between the decoder layers of a multimodal large language model. Explicit labels supervise the model in learning the evidence role of each temporal unit, while continuous within-unit offsets provide supervision for localizing evidence-segment endpoints. Finally, decoding under legality constraints organizes local evidence roles into legal evidence-state chains, and the predicted offsets determine evidence-segment boundaries in continuous time, enabling discrete state chains and continuous boundary localization to jointly determine complete evidence chains. Our contributions are summarized as follows:

- We propose a method for learning temporal evidence structure through state flow, explicitly modeling the onset, continuation, and termination of query-relevant evidence along the timeline as a learnable evidence-state transition process, thereby organizing local evidence segments into structurally complete evidence chains.
- Our theoretical analysis establishes a reversible correspondence between evidence interval sets and canonical evidence state–offset representations. Based on this result, we design a Temporal Structure Learner (TSL) that uses supervised learning to inject information about temporal evidence structure into video representations between the decoder layers of a multimodal large language model, enabling discrete state chains and continuous boundary offsets to jointly determine complete evidence chains.

- Extensive experiments demonstrate the effectiveness of our method for video temporal understanding and show that it substantially outperforms existing methods.

2 Related Work

2.1 Multimodal Video Temporal Understanding

Multimodal video temporal understanding integrates visual and linguistic information to interpret video content, capture when events occur and how they relate over time, and localize evidence intervals relevant to language queries. Early grounding methods evolved from candidate-segment matching (Gao et al., 2017; Zhang et al., 2020b) to direct span prediction (Zhang et al., 2020a). TimeChat (Ren et al., 2024) integrates timestamp-aware representations and temporal instruction tuning into multimodal large language models. For video temporal understanding with evidence grounding, NExT-GQA (Xiao et al., 2024) learns evidence alignment through Gaussian masking and cross-modal self-supervision, while GroundFormer (Seo et al., 2026) further conditions video features on questions before localization and couples answer selection with evidence. Among recent temporal grounding methods, MUSEG (Luo et al., 2026) learns multi-segment grounding with phased rewards, while TimeLens (Zhang et al., 2026a) investigates data refinement, timestamp encoding, and reinforcement learning with verifiable rewards. Building on this line of research, we explicitly organize local evidence states into complete evidence chains and refine their continuous boundaries.

2.2 Flow in Multimodal Video Understanding

Flow has been used at several levels of multimodal understanding, including information propagation and temporal dynamics. For video temporal grounding, Temporal Grounding Bridge (TGB) (Wang et al., 2024) projects optical flow into compact temporal features for multi-span localization and extrapolation to longer videos. Treating video flow as a time series, T3T (Song et al., 2025) combines smoothing based on Brownian bridges, temporal differences, and question-conditioned fusion to capture temporal consistency and variation. Map the Flow (Kim et al., 2026) further analyzes information propagation across model layers following cross-frame interactions. Recent research also employs flow to formulate continuous prediction and organize evidence use. FlowRVS (Wang et al., 2026) applies flow matching to video object segmentation, learning a continuous transformation from video representations to target masks. EFlow (Zhang et al., 2026b) organizes long-video reasoning into temporal evidence localization, grounded reasoning, and confidence-triggered reflection, revisiting the video when the selected evidence is deemed insufficient. Our evidence state flow organizes discrete evidence roles along the video timeline into complete evidence state chains for individual segments, thereby enabling precise video temporal understanding.

3 Method

This section presents our proposed method for learning temporal evidence structure through state flow. Section 3.1 formalizes the problem and defines evidence states for temporal units along with their legal transition constraints. Section 3.2 provides the theoretical analysis, establishing a reversible correspondence between evidence intervals and canonical state–offset representations. Section 3.3 introduces the Temporal Structure Learner (TSL), which incorporates evidence roles and their temporal dependencies into video representations between the decoder layers of a multimodal large language model to support evidence interval prediction and answer generation. Finally, Section 3.4 combines four types of loss to jointly optimize the temporal organization of evidence and the model's ability to answer using the selected evidence.

3.1 Problem Formulation and State Modeling

Given a video V of duration $T > 0$ and a natural language query Q, video temporal understanding requires predicting an answer $\hat{A}$ and localizing a finite set of temporal intervals $\hat{\mathcal{I}}$ corresponding to evidence segments that support the answer: $f_\theta(V, Q) = (\hat{A}, \hat{\mathcal{I}}),\ \hat{A} \in \mathcal{A}(Q)$, where θ denotes the

model parameters and $\mathcal{A}(Q)$ is the admissible answer space.The predicted evidence intervals are represented as $\hat{\mathcal{I}} = \{[\hat{s}_i, \hat{e}_i)\}_{i=1}^{\hat{K}}$, with $0 \le \hat{s}_i < \hat{e}_i \le T$, where $\hat{K} = |\hat{\mathcal{I}}|$ is the number of predicted intervals, and $\hat{s}_i$ and $\hat{e}_i$ denote the start and end times of interval i, respectively.

To explicitly model the onset, continuation, and termination of query-relevant evidence over video time as a state-transition process that can be supervised, we assign each video temporal unit one of five states: background, evidence-segment beginning, interior, ending, or a complete evidence segment contained within a single unit, and enforce legal transitions between these states. We begin with the following definitions.

Definition 1 (Temporal units and temporal partition) *Represent the evidence interval set associated with (V, Q) by its support $S = \bigcup_{k=1}^{K} J_k$, where $J_k = [s_k, e_k)$ are its maximal connected components in temporal order, satisfying $0 \le s_k < e_k \le T$ and $e_k < s_{k+1}$ for $k < K$. The temporal partition of the video is defined as*

$$\mathcal{P} = \{C_j\}_{j=1}^{N}, \qquad C_j = [t_{j-1}, t_j), 0 = t_0 < \cdots < t_N = T, \qquad N \ge 1. \tag{1}$$

Here, C_j is a temporal unit with duration $\Delta_j = t_j - t_{j-1} > 0$; unit durations may differ.

Definition 2 (Admissible temporal partition) *Given the evidence support S and a temporal partition $\mathcal{P}$, let ν_j denote the number of evidence components that overlap unit C_j. The partition $\mathcal{P}$ is admissible for S if and only if each unit overlaps at most one component:*

$$\mathcal{P} \text{ is admissible for } S \iff \max_{1 \le j \le N} \nu_j \le 1. \tag{2}$$

Here, $\nu_j = \sum_{k=1}^{K} \mathbf{1}\{|C_j \cap J_k| > 0\}, j = 1, \ldots, N$; $|\cdot|$ denotes temporal length, and $\mathbf{1}\{\cdot\}$ is the indicator function.

Under an admissible partition, each unit either contains no evidence or overlaps a unique evidence segment, thereby giving its evidence state a well-defined meaning. If any unit satisfies $\nu_j > 1$, the current partition is inadmissible and that unit requires further refinement. Recursively bisect every such unit according to $[a, b) \longmapsto \big\{[a, (a+b)/2),\ [(a+b)/2, b)\big\}$. Each split replaces the parent unit with two child units, each of half its duration. Recompute the overlaps between the child units and evidence segments until each leaf unit overlaps at most one segment; order these leaves chronologically to form the final partition and reindex them as $C_1, \ldots, C_N$. For finite evidence support with positive gaps, this bisection procedure terminates after finitely many splits.

On the final admissible partition, assign each unit an evidence state $z_j \in \mathcal{Y} = \{\mathrm{B}, \mathrm{S}, \mathrm{I}, \mathrm{E}, \mathrm{U}\}$, with $z_j = \mathrm{B}$ when $\nu_j = 0$. For each evidence segment J_k, define its first and last overlapping units as $p_k = \min\{j : |C_j \cap J_k| > 0\}$ and $q_k = \max\{j : |C_j \cap J_k| > 0\}$. The state sequence over these units is defined as

$$z_{p_k:q_k} = \begin{cases} \mathrm{U}, & p_k = q_k, \\ \mathrm{S}\,\mathrm{I}^{q_k - p_k - 1}\,\mathrm{E}, & p_k < q_k, \end{cases} \tag{3}$$

Here, I^m denotes a sequence of m consecutive interior states, with $m = 0$ corresponding to the empty sequence. A temporal unit is labeled B if it does not intersect any evidence segment. If an evidence segment is contained within a single temporal unit, that unit is labeled U. If an evidence segment spans multiple temporal units, the first and last units are labeled S and E, respectively, while all intervening units are labeled I.

Definition 3 (Legal transitions and complete evidence chains) *An evidence state chain $z = (z_1, \ldots, z_N)$ evolves along video time. Its initial and terminal states satisfy $z_1 \in \{\mathrm{B}, \mathrm{S}, \mathrm{U}\}$ and $z_N \in \{\mathrm{B}, \mathrm{E}, \mathrm{U}\}$, respectively, and transitions between adjacent states follow $\{\mathrm{B}, \mathrm{E}, \mathrm{U}\} \longrightarrow \{\mathrm{B}, \mathrm{S}, \mathrm{U}\}$ and $\{\mathrm{S}, \mathrm{I}\} \longrightarrow \{\mathrm{I}, \mathrm{E}\}$; each arrow indicates that any state in the left-hand set may transition to any state in the right-hand set. Equivalently, the set of legal evidence state chains is*

$$\mathcal{G}_N = \{z \in \mathcal{Y}^N : z \text{ matches } (\mathrm{B} \mid \mathrm{U} \mid \mathrm{SI}^*\mathrm{E})^*\}, \tag{4}$$

Here, $\mid$ denotes alternatives and $*$ denotes zero or more repetitions. The legal transition constraints govern the initial and terminal states of the sequence and the transitions between states, ensuring the structural completeness and consistency of evidence state chains.

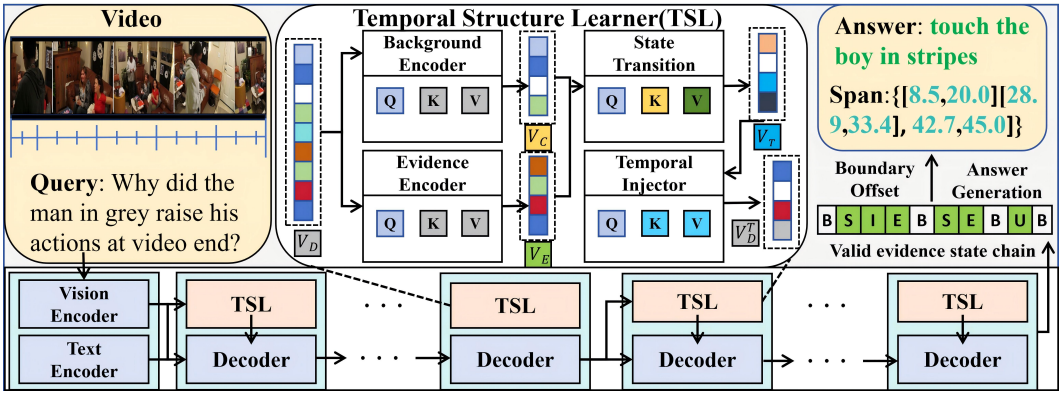

Figure 2: Conditioned on query semantics, the Temporal Structure Learner (TSL) injects information about temporal evidence structure into video representations between the decoder layers of a multimodal large language model. Decoding under legality constraints organizes local evidence roles into legal evidence-state chains, which are combined with predicted offsets to determine evidence-segment boundaries in continuous time, enabling discrete state chains and continuous boundary localization to jointly determine complete evidence chains.

3.2 Theoretical Analysis

To establish the relationship between evidence intervals and the proposed evidence-state chains, we present the following theoretical proposition:

Proposition 1 (Lossless reconstruction and canonical bijection) *For a fixed finite partition $\mathcal{P}$, a finite interval support S has a feasible code $c \in \mathcal{F}_{\mathcal{P}}$ satisfying $\mathrm{Dec}_{\mathcal{P}}(c) = S$ if and only if $\mathcal{P}$ is admissible for S. Moreover, $\mathrm{Enc}_{\mathcal{P}} : \mathcal{S}_{\mathcal{P}} \to \mathcal{C}_{\mathcal{P}}$ and $\mathrm{Dec}_{\mathcal{P}} : \mathcal{C}_{\mathcal{P}} \to \mathcal{S}_{\mathcal{P}}$ are mutual inverses:*

$$\mathrm{Dec}_{\mathcal{P}}(\mathrm{Enc}_{\mathcal{P}}(S)) = S, \qquad S \in \mathcal{S}_{\mathcal{P}}, \tag{5}$$

$$\mathrm{Enc}_{\mathcal{P}}(\mathrm{Dec}_{\mathcal{P}}(c)) = c, \qquad c \in \mathcal{C}_{\mathcal{P}}. \tag{6}$$

Here, $c = (z, \delta)$ consists of a legal evidence state path z and the continuous endpoint offsets δ of its complete chains. The set $\mathcal{S}_{\mathcal{P}}$ contains the finite evidence supports for which $\mathcal{P}$ is admissible; $\mathcal{F}_{\mathcal{P}}$ contains feasible codes satisfying the path and offset constraints, and $\mathcal{C}_{\mathcal{P}}$ is its canonical subset, in which the decoded intervals of distinct chains are separated by positive gaps. The maps $\mathrm{Enc}_{\mathcal{P}}$ and $\mathrm{Dec}_{\mathcal{P}}$ denote encoding and decoding, respectively. The definitions and proof are provided in Appendices C and B.1.

The above proposition establishes representational sufficiency for evidence state flow: under an admissible partition, evidence-state chains and continuous offsets jointly preserve the segment organization and boundary information required to reconstruct evidence intervals. Evidence localization can therefore be decomposed into constrained state-path learning and continuous boundary regression. Learning the state–offset representation enables the joint determination of each temporal unit's segment membership and the continuous start and end positions of evidence segments, thereby organizing local temporal judgments into structurally complete evidence chains.

3.3 Temporal Structure Learner (TSL)

To learn canonical state–offset representations, we design the Temporal Structure Learner (TSL), comprising a background encoder, an evidence encoder, a state transition module, and a temporal injector, as illustrated in Figure 2. TSL is inserted between the decoder layers of a frozen multimodal large language model to learn temporal evidence structure conditioned on query semantics and incorporate it into intermediate video representations. State and boundary-offset heads then predict the evidence role of each temporal unit and the continuous positions of segment endpoints, respectively. Decoding under legal-path constraints and refining the endpoints organize local evidence roles into

complete state chains and determine their evidence intervals. These intervals subsequently select the video evidence used to answer the query.

The four TSL components are independently parameterized Transformer blocks based on cross-attention. Given the multimodal hidden states $X^i \in \mathbb{R}^{M\times H}$ output by decoder layer i, let $\mathcal{V}$ index the video-token positions and extract $V_D^i = X_{\mathcal{V}}^i \in \mathbb{R}^{M_v\times H}$. We set $Q^i = X^i$ as the attention query sequence containing both video and query semantics. Denote the Transformer block for component m by $\mathcal{B}_m^i(Q, K, V)$, whose three arguments specify the cross-attention query, key, and value inputs. The shared computations for cross-attention, residual connections, normalization, and feed-forward networks are specified in Appendix **??**.

(1) Background encoder. The background encoder uses Q^i as queries and the video representations V_D^i as keys and values to aggregate query-relevant video context, providing a contextual reference for identifying evidence roles:

$$V_C^i = \mathcal{B}_C^i(Q^i, V_D^i, V_D^i). \tag{7}$$

(2) Evidence encoder. The evidence encoder reads the same video representations through independently learned parameters to extract query-relevant candidate evidence:

$$V_E^i = \mathcal{B}_E^i(Q^i, V_D^i, V_D^i). \tag{8}$$

The resulting $V_E^i \in \mathbb{R}^{M\times H}$ provides evidence features for subsequent contextual organization.

(3) State transition module. The state transition module uses Q^i as queries, the background representations V_C^i as keys, and the evidence representations V_E^i as values, guiding evidence aggregation through query–context matching:

$$V_T^i = \mathcal{B}_T^i(Q^i, V_C^i, V_E^i). \tag{9}$$

The resulting $V_T^i \in \mathbb{R}^{M\times H}$ integrates the corresponding contextual and evidence information into features for predicting evidence roles and their transitions.

(4) Temporal injector. The temporal injector uses Q^i as queries and reads structural information from V_T^i as both keys and values. Selecting its outputs at video-token positions yields an update aligned with the original video representations:

$$V_D^{i,\mathrm{T}} = \left[\mathcal{B}_{\mathrm{inj}}^i(Q^i, V_T^i, V_T^i)\right]_{\mathcal{V}}. \tag{10}$$

We add this update to the video representations through a residual connection and write the result back into the full multimodal sequence before the next decoder layer:

$$\begin{aligned} \widetilde{X}_{\mathcal{V}}^i &= V_D^i + V_D^{i,\mathrm{T}}, & \widetilde{X}_{\overline{\mathcal{V}}}^i &= X_{\overline{\mathcal{V}}}^i, \\ X^{i+1} &= \mathrm{Decoder}_{i+1}(\widetilde{X}^i), & V_D^{i+1} &= X_{\mathcal{V}}^{i+1}. \end{aligned} \tag{11}$$

Here $\overline{\mathcal{V}}$ indexes non-video tokens. This process progressively incorporates learned temporal evidence structure into video representations, supporting subsequent evidence-state prediction, continuous boundary regression, and answer generation.

We aggregate the video-token outputs of the final decoder layer within each temporal unit to obtain representations R for canonical state–offset prediction. The state prediction head estimates each unit's evidence-state probabilities from R; together with adjacent-state transition scores, these predictions are decoded into a legal evidence-state path under the constraints in Section 3.1. The first and last temporal units of each complete chain define a coarse evidence interval; the boundary-offset module reads the corresponding endpoint-unit representations and predicts continuous offsets to refine the interval's start and end times. After boundary refinement, we merge touching intervals to obtain the final evidence interval set $\hat{\mathcal{I}}$. The model then reads the corresponding evidence E from the original video according to $\hat{\mathcal{I}}$ and generates the answer autoregressively conditioned on E and query Q, making the localized evidence a direct input to answer generation.

3.4 TRAINING OBJECTIVES

We train TSL and the prediction heads with the backbone frozen, encoding evidence roles, transitions, and boundaries in the video representations R of Section 3.3. For $x = (V, Q)$, let S^*, A^* denote the annotated support and answer, and $z^* \in \mathcal{G}_N$ the target state path on an admissible partition (Section 3.1).

(1) Evidence-state supervision. Unit-wise cross-entropy trains representations to distinguish the five evidence roles, with $p^{\mathrm{loc}}_{\theta,j} = \mathrm{softmax}(\ell_{\theta,j})$:

$$\mathcal{L}_{\mathrm{state}} = -\frac{1}{N}\sum_{j=1}^{N} \log p^{\mathrm{loc}}_{\theta,j}(z^*_j). \tag{12}$$

(2) Evidence state-flow reconstruction. To supervise how local roles form complete chains, we use state scores ℓ and adjacent-transition scores ξ computed from R (Appendix **??**) to define a distribution over legal paths:

$$s_\theta(x, z) = \sum_{j=1}^{N} \ell_{\theta,j,z_j} + \sum_{j=1}^{N-1} \xi_{\theta,j,z_j,z_{j+1}},$$
$$\pi_\theta(z \mid x) = \frac{\exp s_\theta(x, z)}{\sum_{\tilde{z}\in\mathcal{G}_N} \exp s_\theta(x, \tilde{z})}, \qquad z \in \mathcal{G}_N. \tag{13}$$

The state–offset decomposition in Proposition 1 motivates joint path and endpoint supervision:

$$\mathcal{L}_{\mathrm{path}} = -\frac{1}{N} \log \pi_\theta(z^* \mid x), \tag{14}$$

$$\mathcal{L}_{\mathrm{end}} = \frac{1}{T}\sum_{k=1}^{K^*} \left(|\hat{s}_k - s^*_k| + |\hat{e}_k - e^*_k|\right), \tag{15}$$

$$\mathcal{L}_{\mathrm{flow}} = \mathcal{L}_{\mathrm{path}} + \eta_b \mathcal{L}_{\mathrm{end}}, \qquad \eta_b > 0. \tag{16}$$

Here $(\hat{s}_k, \hat{e}_k)$ are decoded on the K^* target chains using predicted offsets, and (s^*_k, e^*_k) are their annotated endpoints. The path term trains neighboring representations to favor the target chains among legal alternatives; endpoint regression learns their within-unit boundaries.

(3) Autoregressive answer generation. Given evidence E^* read from S^*, we supervise answer generation with autoregressive cross-entropy:

$$\mathcal{L}_{\mathrm{ans}} = -\frac{1}{L^*}\sum_{t=1}^{L^*} \log p_\theta(a^*_t \mid a^*_{<t}, Q, E^*). \tag{17}$$

Here $A^* = (a^*_1, \ldots, a^*_{L^*})$ includes the end-of-sequence token, and $a^*_{<t}$ denotes the preceding answer prefix.

(4) Evidence-enhanced answer reinforcement learning. To optimize answering from the model's own selections, we sample a legal path and endpoint latents $u = (z, w) \sim \rho_\theta(\cdot \mid x)$, decode $S(u)$, reread n observations $E \sim K_n(\cdot \mid S(u))$, and generate $A \sim p_\theta(\cdot \mid Q, E)$. With union overlap $q(u) = \mathrm{IoU}(S(u), S^*)$ and binary answer correctness $c_Q(A, A^*)$, define

$$r(u, A) = q(u)\,[\eta_r + (1 - \eta_r)c_Q(A, A^*)], \quad 0 < \eta_r < 1,$$
$$\mathcal{L}_{\mathrm{RL}} = -\mathbb{E}_{u,E,A}[r(u, A)]. \tag{18}$$

We set $\mathrm{IoU}(\emptyset, \emptyset) = 1$. Overlap rewards localization even for incorrect answers; correctness adds an overlap-weighted reward. Joint policy gradients update evidence selection and answering; sampling and gradient calculations appear in Appendix **??**.

We optimize all four objectives simultaneously with positive weights:

$$\mathcal{L}_{\mathrm{total}} = \mathbb{E}_{\mathrm{data}}[\lambda_s \mathcal{L}_{\mathrm{state}} + \lambda_f \mathcal{L}_{\mathrm{flow}} + \lambda_a \mathcal{L}_{\mathrm{ans}} + \lambda_r \mathcal{L}_{\mathrm{RL}}]\,. \tag{19}$$

Supervised learning preserves annotation guidance, while RL trains on the answering utility of current evidence predictions.

Table 1: Video temporal understanding results on GroundVideoQA. Bold, underlining, and italics denote the highest, second-highest, and third-highest scores in each column, respectively.

Model	R@0.3	R@0.5	R@0.7	mIoU	ACC@VQA	ACC@GQA
Qwen3-VL-8B (2025)	58.6	49.7	36.9	45.6	*82.8*	49.5
MiMo-VL-7B (2025)	57.5	44.1	24.6	39.5	75.8	44.9
Time-R1-7B (2025)	60.6	48.8	36.1	45.1	77.3	47.9
TRACE (2025)	42.1	31.2	21.4	33.4	74.0	32.1
TimeSuite (2025)	44.8	29.0	22.6	31.2	75.8	30.2
MUSEG (2026)	57.6	46.0	35.1	47.9	79.4	49.3
TimeChat-Captioner (2026)	59.8	45.9	37.5	<u>53.3</u>	74.2	*52.8*
TimeLens-7B (2026)	*62.7*	*51.8*	*37.8*	48.7	**84.0**	<u>53.5</u>
TimeLens-8B (2026)	<u>63.4</u>	<u>55.0</u>	<u>42.3</u>	*50.9*	80.3	51.3
TSL-8B (ours)	**66.5**	**57.1**	**52.2**	**54.6**	<u>83.7</u>	**56.2**

Table 2: Video temporal understanding results on NExT-GQA. Bold, underlining, and italics denote the highest, second-highest, and third-highest scores in each column, respectively.

Model	R@0.3	R@0.5	R@0.7	mIoU	ACC@VQA	ACC@GQA
Qwen3-VL-8B (2025)	37.6	21.5	9.4	29.9	73.7	30.6
MiMo-VL-7B (2025)	37.9	20.9	*10.5*	30.8	67.9	28.6
Time-R1-7B (2025)	39.2	19.9	8.3	30.7	55.7	25.7
TRACE (2025)	39.4	20.4	9.8	28.2	<u>77.5</u>	29.2
TimeSuite (2025)	36.0	22.7	8.9	31.3	69.4	32.2
MUSEG (2026)	40.0	20.8	10.1	29.2	73.5	33.8
TimeChat-Captioner (2026)	40.1	*23.3*	9.1	*32.2*	72.7	<u>37.5</u>
TimeLens-7B (2026)	*42.1*	22.1	9.6	30.5	*76.8*	35.7
TimeLens-8B (2026)	<u>46.3</u>	<u>28.9</u>	<u>14.1</u>	<u>35.6</u>	71.7	*36.6*
TSL-8B (ours)	**56.9**	**29.5**	**14.4**	**45.9**	**81.6**	**58.8**

4 EXPERIMENT

4.1 EXPERIMENTAL SETUP

Datasets and baselines. We evaluate video temporal understanding on groundVideoQA (Wang et al., 2025a) and NExT-GQA (Xiao et al., 2024) in terms of temporal evidence localization and question answering, and further assess temporal localization on ActivityNet-TimeLens (Zhang et al., 2026a). The baselines include TimeLens-8B and TimeLens-7B (Zhang et al., 2026a), MUSEG (Luo et al., 2026), TimeChat-Captioner (Yao et al., 2026), Qwen3-VL-8B (Bai et al., 2025), MiMo-VL-7B (Team et al., 2025), Time-R1-7B (Wang et al., 2025b), TRACE (Guo et al., 2025), and TimeSuite (Zeng et al., 2025).

Evaluation metrics. On groundVideoQA and NExT-GQA, we use Recall@τ (abbreviated as R@τ) and mIoU to evaluate evidence localization, and ACC@VQA and ACC@GQA to assess answer accuracy and joint localization and answering performance, respectively. On ActivityNet-TimeLens, we use R1@τ and mIoU to evaluate temporal localization, with $\tau \in \{0.3, 0.5, 0.7\}$ for both Recall and R1. All metrics are reported as percentages, with higher values indicating better performance; detailed definitions and evaluation protocols are provided in Appendix **??**.

4.2 MAIN RESULTS

Video temporal understanding. TSL-8B achieves the best results across all temporal localization metrics on groundVideoQA and NExT-GQA (Tables 1 and 2). On groundVideoQA, Recall@0.7 exceeds the strongest baseline by 9.9 percentage points, while Recall@0.3 and mIoU improve together on NExT-GQA. These gains in evidence recall and temporal coverage alignment demonstrate the effectiveness of combining constrained state transitions with continuous boundary offsets for video

Table 3: Video temporal localization results on ActivityNet-TimeLens. Bold, underlining, and italics denote the highest, second-highest, and third-highest scores in each column, respectively.

Model	R1@0.3	R1@0.5	R1@0.7	mIoU
Qwen3-VL-8B (2025)	62.1	51.2	*34.4*	46.8
MiMo-VL-7B (2025)	49.3	38.7	22.4	35.5
Time-R1-7B (2025)	44.8	31.0	19.0	33.1
TRACE (2025)	43.4	33.9	22.0	32.7
TimeSuite (2025)	27.1	17.5	8.6	19.8
MUSEG (2026)	*65.8*	48.2	34.0	43.5
TimeChat-Captioner (2026)	59.8	<u>53.7</u>	30.7	*48.2*
TimeLens-7B (2026)	62.8	51.0	32.6	46.2
TimeLens-8B (2026)	**68.9**	**58.4**	<u>40.6</u>	<u>53.2</u>
TSL-8B (ours)	<u>68.4</u>	*52.4*	**41.2**	**57.4**

Table 4: Ablation results for TSL-8B on NExT-GQA. The best score in each column is shown in bold

Configuration	R@0.3	R@0.5	R@0.7	mIoU	ACC@VQA	ACC@GQA
$\mathcal{L}_{\text{state}}$	53.4	17.4	10.5	42.3	77.2	36.9
$\mathcal{L}_{\text{state}} + \mathcal{L}_{\text{flow}}$	55.8	26.4	13.2	42.8	79.2	37.3
$\mathcal{L}_{\text{state}} + \mathcal{L}_{\text{RL}}$	54.9	27.1	13.9	40.4	80.1	53.2
$\mathcal{L}_{\text{state}} + \mathcal{L}_{\text{flow}} + \mathcal{L}_{\text{RL}}$ (Ours)	**56.9**	**29.5**	**14.4**	**45.9**	**81.6**	**58.8**

temporal localization. TSL-8B also achieves the highest ACC@GQA on both question-answering datasets, reflecting improved joint performance in answer correctness and evidence hits and supporting the design of generating answers from evidence selected according to predicted intervals.

Video temporal localization. We further evaluate TSL-8B's video temporal localization capability on ActivityNet-TimeLens. As shown in Table 3, TSL-8B achieves the highest mIoU and R1@0.7, with mIoU exceeding the strongest baseline by 4.2 percentage points. These results show that TSL improves temporal coverage alignment between predicted and annotated segments and increases localization success under a strict overlap criterion, demonstrating its effectiveness in video temporal localization.

4.3 Ablation Studies

Table 4 shows that, with evidence-state supervision, the answer-generation loss, and all other settings held fixed, adding $\mathcal{L}_{\text{flow}}$ improves evidence recall at every evaluated IoU threshold, indicating that joint supervision of complete state paths and continuous endpoints enhances evidence recovery. Adding $\mathcal{L}_{\text{RL}}$ improves answer accuracy and ACC@GQA, highlighting the benefit of feedback on answers generated from predicted evidence for joint localization and answering. The combined configuration achieves the best results on all six metrics, with mIoU and ACC@GQA exceeding the RL configuration without state-flow reconstruction by 5.5 and 5.6 percentage points, respectively. These findings demonstrate the complementary roles of structural reconstruction and answer feedback, as well as the effectiveness of the joint objective in improving evidence alignment and answer generation.

5 Conclusion

We learn temporal evidence structure through state flow to construct complete evidence chains. Experiments demonstrate improved joint performance in localization and answering, while ablation studies verify the complementary roles of structural reconstruction and answer feedback. By combining discrete state chains with continuous boundary localization, our method offers a new perspective on linking temporal localization with answer generation through evidence structure.

AI USE STATEMENT

In this work, we used generative AI tools solely to polish the language of the manuscript drafted by the authors, including correcting grammar, refining wording, and improving transitions between sentences to enhance clarity, fluency, and readability. We did not use these tools for research ideation, method design, theoretical derivations, data processing, code development, experimental work, or analysis of results. We have reviewed and verified all text polished with AI assistance to ensure that it accurately conveys the authors' intended meaning and remains consistent with the technical content and conclusions of the study. We take full responsibility for the final content of the paper, including the text polished with AI assistance and the scholarly claims it contains.

REFERENCES

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, 2025. URL https://arxiv.org/abs/2511.21631.

Shangzhe Di and Weidi Xie. Grounded Question-Answering in Long Egocentric Videos . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12934–12943, Los Alamitos, CA, USA, June 2024. IEEE Computer Society. doi: 10.1109/CVPR52733.2024.01229. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01229.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5277–5285, 2017. doi: 10.1109/ICCV.2017.563.

Yongxin Guo, Jingyu Liu, Mingda Li, Qingbin Liu, Xi Chen, and Xiaoying Tang. TRACE: Temporal grounding video LLM via causal event modeling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=14fFV0chUS.

Ayush Gupta, Anirban Roy, Rama Chellappa, Nathaniel D Bastian, Alvaro Velasquez, and Susmit Jha. Toga: Temporally grounded open-ended video qa with weak supervision. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23593–23603. IEEE, 2025.

Minji Kim, Taekyung Kim, and Bohyung Han. Map the flow: Revealing hidden pathways of information in videoLLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=QCB0HN61TU.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 706–715, 2017. doi: 10.1109/ICCV.2017.83.

Jie Lei, Tamara L. Berg, and Mohit Bansal. Qvhighlights: detecting moments and highlights in videos via natural language queries. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Lou, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22195–22206, 2024. doi: 10.1109/CVPR52733.2024.02095.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.342. URL https://aclanthology.org/2024.emnlp-main.342/.

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. TempCompass: Do video LLMs really understand videos? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8731–8772, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.517. URL https://aclanthology.org/2024.findings-acl.517/.

Yujie Lu, Yale Song, William Wang, Lorenzo Torresani, and Tushar Nagarajan. VITED : Video Temporal Evidence Distillation . In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8501–8511, Los Alamitos, CA, USA, June 2025. IEEE Computer Society. doi: 10.1109/CVPR52734.2025.00795. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52734.2025.00795.

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 Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35549–35561, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1644. URL https://aclanthology.org/2026.acl-long.1644/.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14313–14323, 2024. doi: 10.1109/CVPR52733.2024.01357.

Jinhwan Seo, Kyubeom Han, Jumin Lee, Junhyug Noh, and Sung eui Yoon. What you ask is what you ground: Bridging question intent to temporal evidence for grounded videoqa, 2026. URL https://arxiv.org/abs/2608.15708.

Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, Yan Lu, Jenq-Neng Hwang, and Gaoang Wang. Moviechat: From dense token to sparse memory for long video understanding. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18221–18232, 2024. doi: 10.1109/CVPR52733.2024.01725.

Zijie Song, Zhenzhen Hu, Yixiao Ma, Jia Li, and Richang Hong. Video flow as time series: Discovering temporal consistency and variability for videoqa. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2025. doi: 10.1109/icme59968.2025.11209469. URL http://dx.doi.org/10.1109/ICME59968.2025.11209469.

Core Team, Zihao Yue, Zhenru Lin, Yifan Song, Weikun Wang, Shuhuai Ren, Shuhao Gu, Shicheng Li, Peidian Li, Liang Zhao, Lei Li, Kainan Bao, Hao Tian, Hailin Zhang, Gang Wang, Dawei Zhu, Cici, Chenhong He, Bowen Ye, Bowen Shen, Zihan Zhang, Zihan Jiang, Zhixian Zheng, Zhichao Song, Zhenbo Luo, Yue Yu, Yudong Wang, Yuanyuan Tian, Yu Tu, Yihan Yan, Yi Huang, Xu Wang, Xinzhe Xu, Xingchen Song, Xing Zhang, Xing Yong, Xin Zhang, Xiangwei Deng, Wenyu Yang, Wenhan Ma, Weiwei Lv, Weiji Zhuang, Wei Liu, Sirui Deng, Shuo Liu, Shimao Chen, Shihua Yu, Shaohui Liu, Shande Wang, Rui Ma, Qiantong Wang, Peng Wang, Nuo Chen, Menghang Zhu, Kangyang Zhou, Kang Zhou, Kai Fang, Jun Shi, Jinhao Dong, Jiebao Xiao, Jiaming Xu, Huaqiu Liu, Hongshen Xu, Heng Qu, Haochen Zhao, Hanglong Lv, Guoan Wang, Duo Zhang, Dong Zhang, Di Zhang, Chong Ma, Chang Liu, Can Cai, and Bingquan Xia. Mimo-vl technical report, 2025. URL https://arxiv.org/abs/2506.03569.

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 Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 959–975, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.50. URL https://aclanthology.org/2025.findings-emnlp.50/.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=gJ05Gm5VxQ.

Yuxuan Wang, Yueqian Wang, Pengfei Wu, Jianxin Liang, Dongyan Zhao, Yang Liu, and Zilong Zheng. Efficient temporal extrapolation of multimodal large language models with temporal grounding bridge. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9972–9987, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.556. URL https://aclanthology.org/2024.emnlp-main.556/.

Zanyi Wang, Dengyang Jiang, Liuzhuozheng Li, Sizhe Dang, Chengzu Li, Harry Yang, Guang Dai, Mengmeng Wang, and Jingdong Wang. Deforming videos to masks: Flow matching for referring video segmentation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3KaIcArMAB.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9772–9781, 2021. doi: 10.1109/CVPR46437.2021.00965.

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

Linli Yao, Yuancheng Wei, Yaojie Zhang, Lei Li, Xinlong Chen, Feifan Song, Ziyue Wang, Kun Ouyang, Yuanxin Liu, Lingpeng Kong, Qi Liu, Pengfei Wan, Kun Gai, Yuanxing Zhang, and Xu Sun. Timechat-captioner: Scripting multi-scene videos with time-aware and structural audio-visual captions. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=TL787dxmIM.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=nAVejJURqZ.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. Span-based localizing network for natural language video localization. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6543–6554, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.585. URL https://aclanthology.org/2020.acl-main.585/.

Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, and Limin Wang. Timelens: Rethinking video temporal grounding with multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10419–10429, 2026a.

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, pp. 12870–12877, 2020b.

Wenhao Zhang, Kuanwei Lin, Xuyi Yang, Wei Gao, and Ge Li. Eflow: Learning evidence flow for long-video reasoning with adaptive reflection, 2026b. URL https://arxiv.org/abs/2607.00867.

A Supplementary Experimental Details

A.1 Implementation Details

Experiments are conducted on six NVIDIA GeForce RTX 4090 GPUs using PyTorch 2.7.1 and Adam. Following Section 3.4, the multimodal backbone is frozen, while TSL and the state and boundary-offset prediction heads are optimized jointly. For the provisional default configuration, we set the learning rate to 10^{-4} and train for three epochs, with one video–query pair per GPU and gradient accumulation over four steps, giving an effective batch size of $6 \times 1 \times 4 = 24$. Adam uses $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and zero weight decay; the learning rate is constant, the gradient ℓ_2 norm is clipped to 1.0, and the random seed is 42. These provisional hyperparameters require verification against the configurations used to obtain the reported results.

A.2 Dataset Statistics

Grounded-VideoQA contains 17,096 videos, 17,214 question–answer pairs, and 17,214 raw evidence interval annotations. Since the NExT-GQA training split does not provide evidence interval annotations, we combine its validation and test splits to form our evaluation set, comprising 1,557 videos, 8,911 question–answer pairs, and 10,531 evidence intervals. ActivityNet-TimeLens contains 1,455 videos, 4,500 queries, and 4,500 target evidence intervals.

A.3 Evaluation Metrics and Protocols

Let the evaluation set contain n queries. For the i-th query, denote the predicted and annotated interval collections by $\hat{\mathcal{I}}_i$ and $\mathcal{I}_i$, respectively, where K_i is the number of annotated intervals. All metrics are reported as percentages.

Temporal intersection over union (tIoU). For positive-length intervals $J = [s, e)$ and $J' = [s', e')$, tIoU is computed by dividing their intersection duration $d(J, J')$ by their union duration:

$$\begin{aligned} d(J, J') &= \max\{0, \min(e, e') - \max(s, s')\}, \\ \mathrm{tIoU}(J, J') &= \frac{d(J, J')}{(e - s) + (e' - s') - d(J, J')}. \end{aligned} \tag{20}$$

Recall@τ. On Grounded-VideoQA and NExT-GQA, reversed endpoints are reordered and zero-duration intervals are removed before predictions are processed in descending confidence order. Each prediction selects the annotation with the highest tIoU for the same query and counts as a successful match only if this tIoU is at least τ and the selected annotation has not already been matched; otherwise, no alternative annotation is considered. Ties are resolved by timestamp order. Let $M_i(\tau)$ denote the number of successful matches for query i. Then

$$\mathrm{Recall}\,@\tau = 100\,\frac{\sum_{i=1}^{n} M_i(\tau)}{\sum_{i=1}^{n} K_i}, \qquad \tau \in \{0.3, 0.5, 0.7\}. \tag{21}$$

Each annotated interval participates in matching separately, and recall is set to zero when the denominator is zero.

mIoU for interval collections. On these two datasets, we form the unions of the predicted and annotated intervals, denoted by $\hat{S}_i$ and S_i, respectively, and average their IoU over queries:

$$\mathrm{mIoU} = \frac{100}{n} \sum_{i=1}^{n} \frac{|\hat{S}_i \cap S_i|}{|\hat{S}_i \cup S_i|}. \tag{22}$$

Here, $|\cdot|$ denotes temporal duration, with overlapping portions counted only once. The IoU for a query is set to one when both unions are empty and to zero when exactly one is empty.

R1@τ and mIoU on ActivityNet-TimeLens. Only the first predicted interval in the output list is evaluated against the target interval, and their tIoU is denoted by v_i. If the prediction list is empty or the predicted interval has nonpositive duration, we set $v_i = 0$. The metrics are computed as

$$\mathrm{R1}\ @\tau = \frac{100}{n} \sum_{i=1}^{n} \mathbf{1}[v_i \geq \tau], \tag{23}$$

$$\mathrm{mIoU} = \frac{100}{n} \sum_{i=1}^{n} v_i. \tag{24}$$

Here, $\tau \in \{0.3, 0.5, 0.7\}$, and $\mathbf{1}[\cdot]$ denotes the indicator function. Candidate ordering is independent of the annotations, and queries with unsuccessful predictions remain included in the denominator.

ACC@VQA and ACC@GQA. Predicted and annotated answers are normalized by lowercasing, trimming leading and trailing whitespace, and collapsing consecutive whitespace. We set $c_i = 1$ if the normalized answers match exactly and $c_i = 0$ otherwise. We set $h_i = 1$ if at least one predicted–annotated interval pair has tIoU strictly greater than 0.3, and $h_i = 0$ otherwise. For the subset $\mathcal{D}_A$ of queries with annotated answers, let n_A denote its size. The accuracies are computed as

$$\mathrm{ACC@VQA} = \frac{100}{n_A} \sum_{i \in \mathcal{D}_A} c_i, \tag{25}$$

$$\mathrm{ACC@GQA} = \frac{100}{n_A} \sum_{i \in \mathcal{D}_A} c_i h_i. \tag{26}$$

Evidence hits are determined using unmerged interval pairs. If either interval collection is empty, $h_i = 0$; when $n_A = 0$, neither accuracy is reported.

A.4 Baseline Models

The nine baseline configurations in Section 4.2 are summarized below.

- **TimeLens-8B** (Zhang et al., 2026a) adapts Qwen3-VL-8B-Instruct for temporal grounding using TimeLens-100K, interleaved textual timestamps, and reinforcement learning with verifiable rewards without generating reasoning traces.
- **TimeLens-7B** (Zhang et al., 2026a) applies the same TimeLens grounding recipe to Qwen2.5-VL-7B-Instruct.
- **MUSEG** (Luo et al., 2026) uses reinforcement learning with phased rewards to align a query with multiple relevant video segments through timestamp-aware grounding.
- **TimeChat-Captioner** (Yao et al., 2026) is a 7B audiovisual captioning model trained through supervised fine-tuning and group relative policy optimization to generate structured descriptions with explicit timestamps.
- **Qwen3-VL-8B** (Bai et al., 2025) combines interleaved multimodal rotary position embeddings, multilevel visual features, and textual timestamp alignment in a general-purpose vision–language model.
- **MiMo-VL-7B** (Team et al., 2025) is a vision–language model developed through four-stage multimodal pretraining for general visual understanding and multimodal reasoning.
- **Time-R1-7B** (Wang et al., 2025b) is a temporal grounding model post-trained through reasoning-guided reinforcement learning with verifiable rewards.
- **TRACE** (Guo et al., 2025) models video outputs as causal event sequences that interleave timestamps, saliency scores, and textual captions using dedicated prediction heads.
- **TimeSuite** (Zeng et al., 2025) denotes VideoChat-T, which extends VideoChat to long videos through token shuffling, temporal adaptive position encoding, and grounded instruction tuning.

B Theoretical Proofs and Further Analysis

The following proofs use the definitions in Appendix C and exact real arithmetic for continuous offsets.

B.1 Proof of Proposition 1

Necessity. For any feasible code, distinct chains occupy disjoint unit blocks. Writing $\eta^s_{p_k} = \alpha_k$ and $\eta^e_{q_k} = \beta_k$ for active offsets, its decoded support S_c satisfies

$$C_j \cap S_c = \begin{cases} \emptyset, & z_j = \mathrm{B}, \\ [t_{j-1} + \Delta_j \eta^s_j, t_j), & z_j = \mathrm{S}, \\ C_j, & z_j = \mathrm{I}, \\ [t_{j-1}, t_j - \Delta_j \eta^e_j), & z_j = \mathrm{E}, \\ [t_{j-1} + \Delta_j \eta^s_j, t_j - \Delta_j \eta^e_j), & z_j = \mathrm{U}. \end{cases} \tag{27}$$

Feasibility makes each nonempty intersection a single positive-length interval. A unit overlapping two maximal components would instead contain two pieces separated by a positive gap, contradicting this local form. Thus $\mathcal{P}$ is admissible for every feasible decoding.

Sufficiency and exact reconstruction. For an admissible support S, each component meets consecutive units $p_k, \ldots, q_k$, and different components have disjoint unit blocks. Their states therefore form a legal path. Positive overlap gives $t_{p_k-1} \le s_k < t_{p_k}$ and $t_{q_k-1} < e_k \le t_{q_k}$, so the offsets in equation 42 lie in $[0, 1)$. For a single-unit component, $\alpha_k + \beta_k = 1 - (e_k - s_k)/\Delta_{p_k} < 1$. The encoding is feasible, and substitution into equation 41 recovers (s_k, e_k) exactly. Since maximal components have positive gaps, the code is canonical and $\mathrm{Dec}_{\mathcal{P}}(\mathrm{Enc}_{\mathcal{P}}(S)) = S$.

Inverse and uniqueness. For a canonical code c, its decoded chains are exactly the maximal components of S_c. Equation equation 27 identifies their first and last positively intersected units as the original p_k, q_k, so re-encoding recovers the state path. Each endpoint has a unique normalized offset because $\Delta_j > 0$; hence $\mathrm{Enc}_{\mathcal{P}}(S_c) = c$. The empty support satisfies both identities by definition. This proves the equivalence and the bijection in Proposition 1.

Canonicalization and continuous offsets. On cells $[0, 1)$ and $[1, 2)$, zero-offset paths UU and SE both decode to $[0, 2)$, but only SE is canonical. For any feasible code, Proposition 1 therefore yields the support-preserving, idempotent normalization

$$\begin{aligned} \mathcal{N}_{\mathcal{P}} &= \mathrm{Enc}_{\mathcal{P}} \circ \mathrm{Dec}_{\mathcal{P}}, \\ \mathrm{Dec}_{\mathcal{P}}(\mathcal{N}_{\mathcal{P}}(c)) &= \mathrm{Dec}_{\mathcal{P}}(c), \\ \mathcal{N}_{\mathcal{P}}(\mathcal{N}_{\mathcal{P}}(c)) &= \mathcal{N}_{\mathcal{P}}(c). \end{aligned} \tag{28}$$

Indeed, normalization returns the unique canonical code of the decoded support. Merging is performed after continuous decoding so that positive gaps between adjacent chains are retained. Offsets are essential: within $[0, 1)$, both $[1/4, 1/2)$ and $[1/2, 3/4)$ have state U but different endpoints.

B.2 Finite Refinement and Support Invariance

Proposition 2 (Finite refinement) *For $K \ge 2$ maximal components, let $g = \min_{k<K}(s_{k+1} - e_k) > 0$. Starting from N_0 units with maximum width W, recursively bisecting ambiguous units reaches an admissible partition at depth at most*

$$D = \max\{0, \lceil \log_2(W/g) \rceil\}, \tag{29}$$

with at most $N_0 2^D$ leaves. For $K \le 1$, no split is needed. Every refinement $\mathcal{P}'$ of an admissible $\mathcal{P}$ remains admissible, and

$$\mathrm{Dec}_{\mathcal{P}'}(\mathrm{Enc}_{\mathcal{P}'}(S)) = \mathrm{Dec}_{\mathcal{P}}(\mathrm{Enc}_{\mathcal{P}}(S)) = S. \tag{30}$$

Proof. A unit $[a, b)$ overlapping two components $k < \ell$ with positive length must satisfy $a < e_k$ and $b > s_\ell$, giving $b - a > s_\ell - e_k \geq g$. At depth D, every descendant has width at most $W2^{-D} \leq g$ and is therefore unambiguous. Each initial unit has at most 2^D leaves. A child of an admissible unit cannot overlap more components than its parent, proving preservation of admissibility. Proposition 1 then gives support invariance. States and offsets must be recomputed on the refined partition; for example, $[1/4, 3/4)$ changes from U to SE when $[0, 1)$ is bisected. The refinement rule tests the specified support and thus provides an annotation-based target construction.

B.3 Legal Paths and Exact Decoding

Construct a layered acyclic graph with nodes (j, a), $a \in \mathcal{Y}$, using the transitions in Definition 3; source and sink edges impose the initial and terminal constraints. A nonnegative unit flow satisfies

$$\sum_{e \in \mathrm{out}(v)} f_e - \sum_{e \in \mathrm{in}(v)} f_e = \mathbf{1}[v = \mathrm{src}] - \mathbf{1}[v = \mathrm{snk}]. \tag{31}$$

Proposition 3 (Flow decomposition and legal decoding) *Every feasible unit flow is a convex combination of legal path indicators. Integral unit flows correspond exactly to paths in $\mathcal{G}_N$, each consisting only of background units and complete evidence chains. Maximizing a linear flow score with finite coefficients admits an integral optimum.*

Proof. Follow positive-flow edges from the source. Conservation supplies an outgoing edge at each internal node, and acyclicity ensures arrival at the sink. Subtract the minimum flow along this path; at least one positive edge disappears. Repeating terminates in finitely many steps. Once the source flow is zero, conservation forces zero residual flow successively through all layers. Hence $f = \sum_r \lambda_r \mathbf{1}_{P_r}$ with $\lambda_r > 0$ and $\sum_r \lambda_r = 1$. An integral unit flow must follow one path, while every legal path indicator is feasible. Along a path, each S is followed only by I states until an E; terminal constraints prohibit unfinished chains. The remaining blocks are B and U, giving unique complete-chain parsing. Finally, a linear flow score is an average of path scores and cannot exceed their maximum, which is attained by a path indicator.

For node scores ℓ and allowed-transition scores ξ, the maximizing path is obtained by backtracking

$$\begin{aligned} v_1(a) &= \begin{cases} \ell_{1,a}, & a \in \{\mathrm{B}, \mathrm{S}, \mathrm{U}\}, \\ -\infty, & \text{otherwise}, \end{cases} \\ v_j(a) &= \ell_{j,a} + \max_{b:\, b \to a} \{v_{j-1}(b) + \xi_{j-1,b,a}\}, \\ v_{\mathrm{opt}} &= \max_{a \in \{\mathrm{B},\mathrm{E},\mathrm{U}\}} v_N(a). \end{aligned} \tag{32}$$

The recurrence uses $j = 2, \ldots, N$ and takes $O(N|\mathcal{Y}|^2)$ time. Replacing both maximizations by log-sum-exp computes the log-partition function. Thus exact decoding and normalization use the same legal support.

B.4 State and Boundary Error Bounds

On a partition admissible for S^*, let $c^* = (z^*, \delta^*) = \mathrm{Enc}_{\mathcal{P}}(S^*)$ and let a feasible prediction $\hat{c} = (\hat{z}, \hat{\delta})$ decode to $\hat{S}$. Use η_j^s, η_j^e for active offsets as in equation 27; hats and stars distinguish predictions and targets.

Proposition 4 (Coverage and IoU error bounds) *Define*

$$\begin{aligned} B_{\mathrm{role}} &= \sum_{j:\hat{z}_j \neq z_j^*} \Delta_j, \\ B_{\mathrm{offset}} &= \sum_{j:\hat{z}_j = z_j^* \in \{\mathrm{S},\mathrm{U}\}} \Delta_j |\hat{\eta}_j^s - \eta_j^{s,*}| \\ &\quad + \sum_{j:\hat{z}_j = z_j^* \in \{\mathrm{E},\mathrm{U}\}} \Delta_j |\hat{\eta}_j^e - \eta_j^{e,*}|. \\ |\hat{S} \triangle S^*| &\leq B_{\mathrm{sb}} := B_{\mathrm{role}} + B_{\mathrm{offset}}. \end{aligned} \tag{33}$$

If $\hat{z} = z^*$, then, for chain endpoints before merging,

$$B_{\mathrm{sb}} = B_{\mathrm{abs}} := \sum_k \big(|\hat{s}_k - s_k^*| + |\hat{e}_k - e_k^*|\big)$$
$$= \sum_k \Big(\Delta_{p_k}|\hat{\alpha}_k - \alpha_k^*| + \Delta_{q_k}|\hat{\beta}_k - \beta_k^*|\Big). \tag{34}$$

For $|S^*| > 0$, the support IoU satisfies

$$1 - \mathrm{IoU}(\hat{S}, S^*) = \frac{|\hat{S} \triangle S^*|}{|\hat{S} \cup S^*|} \le \min\left\{1, \frac{B_{\mathrm{sb}}}{|S^*|}\right\}. \tag{35}$$

Proof. The cellwise symmetric differences sum to $|\hat{S} \triangle S^*|$. A mismatched-state cell contributes at most Δ_j. Matching B or I states contribute zero; matching S or E states contribute the respective endpoint displacement. For matching U states, use $|[a, b) \triangle [a', b')| \le |a - a'| + |b - b'|$, which follows by expressing interval indicators as differences of half-line indicators and applying the triangle inequality. Summing proves the coverage bound without requiring equal chain counts. If the paths agree, their unique decompositions pair the chains, and subtracting equation 41 gives the endpoint identities above. Finally, the union is the disjoint union of the intersection and symmetric difference, and $|\hat{S} \cup S^*| \ge |S^*|$, proving the IoU bound.

B.5 Evidence Reading and Answer Stability

Fix V, Q, A^*, θ and $n \ge 1$. For $|S| > 0$, assume $\mathrm{Read}_n(V, S)$ samples n times independently and uniformly from S, then sorts them and forms answer inputs through a common observation map. This map receives only the sampled video content, original timestamps, and fixed metadata $(T, \mathcal{P})$; the support affects its inputs only through the sampled times. Define

$$F_\theta(S) = \mathbb{E}_{E \sim \mathrm{Read}_n(V,S)} p_\theta(A^* \mid Q, E),$$
$$L_A^*(\theta) = \mathbb{E}_{E^* \sim \mathrm{Read}_n(V,S^*)}[-\log p_\theta(A^* \mid Q, E^*)]. \tag{36}$$

Proposition 5 (Stability under a common reading protocol) *For $|S|, |S^*| > 0$, let $d = 1 - |S \cap S^*| / \max(|S|, |S^*|)$. Then*

$$|F_\theta(S) - F_\theta(S^*)| \le 1 - (1 - d)^n \le \min\left\{1, \frac{n|S \triangle S^*|}{|S^*|}\right\}. \tag{37}$$

Under Proposition 4, if $|\hat{S}|, |S^*| > 0$ and $L_A^*(\theta) < \infty$, then

$$F_\theta(\hat{S}) \ge \max\left\{0, \exp[-L_A^*(\theta)] - \min\left(1, \frac{nB_{\mathrm{sb}}}{|S^*|}\right)\right\}. \tag{38}$$

Proof. Write $a = |S|$, $b = |S^*|$, and $c = |S \cap S^*|$. The uniform time densities have common mass $c/\max(a, b)$, so their total variation distance is d. The product densities for n independent samples have common mass $(c/\max(a, b))^n$, giving

$$\mathrm{TV}\big(\mathrm{Unif}(S)^{\otimes n}, \mathrm{Unif}(S^*)^{\otimes n}\big) = 1 - (1 - d)^n. \tag{39}$$

A common observation map cannot increase total variation, and the expectation difference of any $[0, 1]$-valued function is bounded by that distance. Applying this to $p_\theta(A^* \mid Q, E)$ proves the first inequality. Moreover, $1 - (1 - d)^n \le \min\{1, nd\}$ and $d \le (a + b - 2c)/b = |S \triangle S^*|/|S^*|$, which gives the second inequality. Jensen's inequality yields $F_\theta(S^*) \ge \exp[-L_A^*(\theta)]$. Combining this with the stability bound, Proposition 4, and $F_\theta(\hat{S}) \ge 0$ proves the final claim.

Here L_A^* uses the complete-answer log-likelihood: for the fixed target length L^* in equation 17, $L_A^* = L^* \, \mathbb{E}[\mathcal{L}_{\mathrm{ans}}]$. The result bounds expected answer probability under the stated sampling protocol; it may be vacuous and does not guarantee greedy-answer accuracy.

C Supplementary Definitions

This appendix specifies the support, coding, and model notation used in Section 3. We retain the temporal partition $\mathcal{P} = \{C_j = [t_{j-1}, t_j)\}_{j=1}^N$ and unit durations $\Delta_j = t_j - t_{j-1} > 0$ from Section 3.1.

C.1 Evidence Support and Endpoint Ownership

The support $S = \bigcup_{J \in \mathcal{I}} J$ records the temporal coverage of a finite interval set $\mathcal{I}$. Merging overlapping or touching intervals gives its maximal components $J_k = [s_k, e_k)$, ordered so that $e_k < s_{k+1}$. Reconstruction of S preserves this coverage; annotation multiplicity and boundaries inside a merged component are not part of the representation. Let $\mathcal{S}_{\mathcal{P}}$ denote the finite interval supports for which $\mathcal{P}$ is admissible under Definition 2, including $\emptyset$.

For component J_k, the first and last units with positive-length overlap are p_k and q_k. Their endpoint ownership is characterized by

$$t_{p_k-1} \le s_k < t_{p_k}, \qquad t_{q_k-1} < e_k \le t_{q_k}. \tag{40}$$

Thus, at an internal partition boundary t_j, a start is assigned to C_{j+1} and an end to C_j. This convention uniquely assigns both endpoints, including those exactly on partition boundaries.

C.2 State–Offset Codes and Their Mappings

By the grammar in equation 4, a legal path $z \in \mathcal{G}_N$ decomposes uniquely into background units and complete U or SI*E chains. Index its $K(z)$ chains chronologically, with first and last units p_k, q_k.

Definition 4 (Feasible code) *A code $c = (z, \delta) \in \mathcal{F}_{\mathcal{P}}$ consists of a legal path and its active offsets $\delta = ((\alpha_k, \beta_k))_{k=1}^{K(z)}$, where $0 \le \alpha_k, \beta_k < 1$ and $\alpha_k + \beta_k < 1$ when $p_k = q_k$. Its decoding is*

$$\bar{s}_k = t_{p_k-1} + \Delta_{p_k}\alpha_k, \qquad \bar{e}_k = t_{q_k} - \Delta_{q_k}\beta_k,$$
$$\mathrm{Dec}_{\mathcal{P}}(c) = \bigcup_{k=1}^{K(z)} [\bar{s}_k, \bar{e}_k). \tag{41}$$

The all-background path has no active offsets and decodes to $\emptyset$.

Here α_k and β_k measure inward distances from the left edge of C_{p_k} and the right edge of C_{q_k}, normalized by the respective unit durations. The constraints retain positive overlap with both endpoint units; for U, the joint constraint ensures $\bar{s}_k < \bar{e}_k$. Only active offsets belong to the code. Equivalently, unit-indexed notation uses $\eta^s_{p_k} = \alpha_k$ and $\eta^e_{q_k} = \beta_k$, active at S/U and E/U states, respectively.

Definition 5 (Canonical code) *A feasible code belongs to $\mathcal{C}_{\mathcal{P}}$ if $\bar{e}_k < \bar{s}_{k+1}$ for all consecutive chains. Its decoded chains are therefore the maximal components of its support.*

Feasible chains may touch at unit boundaries. Canonicalization merges touching decoded intervals and re-encodes the resulting maximal components, retaining every positive gap.

Encoding. For $S \in \mathcal{S}_{\mathcal{P}}$ with maximal components $[s_k, e_k)$, assign the states in equation 3 and define

$$\alpha_k = \frac{s_k - t_{p_k-1}}{\Delta_{p_k}}, \qquad \beta_k = \frac{t_{q_k} - e_k}{\Delta_{q_k}}. \tag{42}$$

This defines $\mathrm{Enc}_{\mathcal{P}}(S) = (z, \delta)$. The empty support is encoded by the all-background path with no active offsets. Proposition 1 establishes the inverse relationship on canonical codes; its proof appears in Appendix B.1.

C.3 Model and Training Notation

Queries and representations. Q denotes the natural language query, whereas $Q^i = X^i \in \mathbb{R}^{M \times H}$ is the multimodal attention-query sequence in TSL. Here M is the prefix length, H is the hidden width, and $\mathcal{V}$ indexes the M_v video tokens, giving $V^i_D = X^i_{\mathcal{V}} \in \mathbb{R}^{M_v \times H}$. The temporal representation $R = [h_1; \ldots; h_N] \in \mathbb{R}^{N \times H}$ contains learned unit features; the symbolic code $c = (z, \delta)$ is predicted from these features through the state and boundary heads. Detailed computations are given in Appendix **??**.

Local scores and legal-path probabilities. For $x = (V, Q)$, $\ell_{\theta,j,a}$ scores state a at unit j, and $\xi_{\theta,j,a,b}$ scores the transition $a \to b$ between units j and $j+1$. The local distribution $p^{\mathrm{loc}}_{\theta,j} = \mathrm{softmax}(\ell_{\theta,j})$ supervises individual states. The legal-path distribution $\pi_\theta(z \mid x)$ in equation 13 additionally incorporates transition scores and path constraints. Its state marginal is

$$\gamma_j(a) = \sum_{\substack{z \in \mathcal{G}_N \\ z_j = a}} \pi_\theta(z \mid x), \tag{43}$$

which generally differs from $p^{\mathrm{loc}}_{\theta,j}(a)$.

Targets, actions, and observations. Stars denote annotated targets and hats denote predictions. In $\mathcal{L}_{\mathrm{end}}$, predicted endpoints are decoded on the target chains z^*, so each predicted–target pair refers to the same chain. For reinforcement learning, $u = (z, w)$ contains a legal path and the vector $w \in \mathbb{R}^{2K(z)}$ of active endpoint latents. The feasible map h_z in equation **??** gives $\delta = h_z(w)$ and $S(u) = \mathrm{Dec}_{\mathcal{P}}(z, \delta)$; the all-background path has an empty latent vector. The law $\rho_\theta(u \mid x)$ governs this joint action. The evidence E consists of observations read from $S(u)$, with conditional law $K_n(\cdot \mid S(u))$, and the answer is generated from $p_\theta(A \mid Q, E)$. Thus z, δ, and E denote the discrete path, continuous offsets, and video observations, respectively.

D CASE STUDY

Figures 3 and 4 present four qualitative examples. Each case displays video frames and a query together with evidence spans, an evidence state chain, refined temporal boundaries, and an answer.

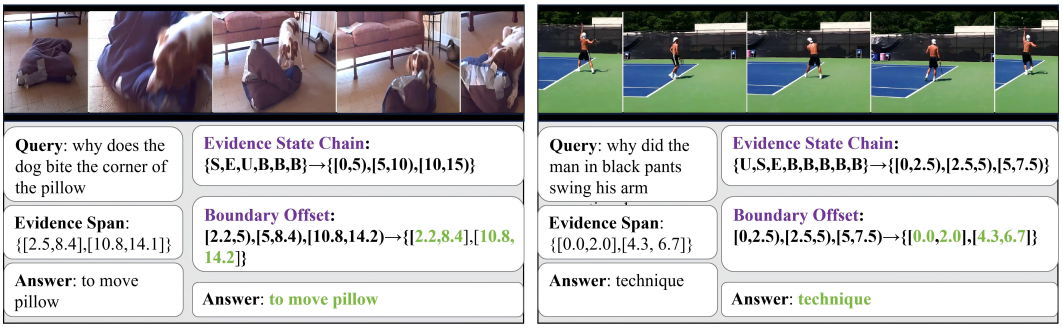

Figure 3: Cases involving a dog interacting with a pillow (left) and a tennis player (right). Both cases combine evidence spanning two units (SE) with evidence contained in a single unit (U). Boundary offsets refine the endpoints of the resulting intervals.

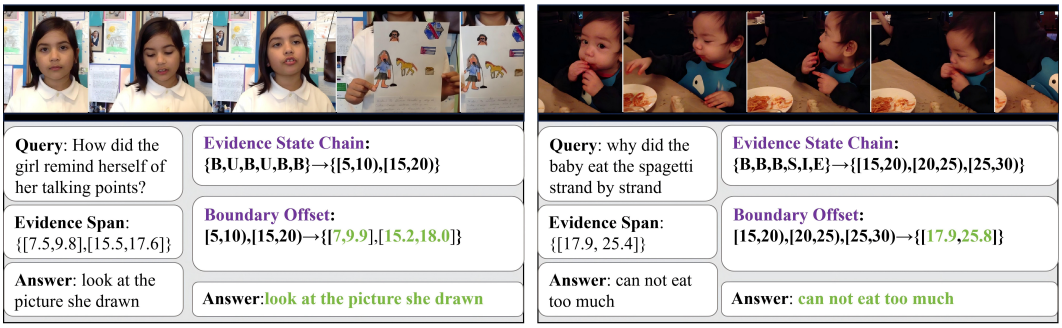

Figure 4: Cases involving a girl referring to a drawing (left) and a child eating spaghetti (right). The left case contains two separate U segments, while the right case forms one SIE chain spanning three units. Boundary offsets determine the continuous endpoints in both cases.