VISUAL EVIDENCE-BASED REASONING FOR UNDERSTANDING MULTI-OBJECT MULTI-STATE TRANSITIONS IN VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding object states in videos requires not only recognizing physical properties but also linking state changes to specific actions and conditions. Recognizing states and actions does not ensure correct object–state associations or causal explanations: Even when models correctly identify states, the explanations they provide may not align with the actual video content despite potentially conforming to common sense. Models therefore need to reason from visual evidence to connect each state change to the specific operations and conditions affecting the corresponding object. To address this, we propose a graph-guided visual evidence-based reasoning framework for understanding multi-object multi-state transitions in videos. A state graph is built based on timestamped segment descriptions and visual observations through iterative relation verification and state completion by leveraging a Vision-Language Model (VLM). In the graph, nodes represent objects and their states at specific stages, and edges represent object state transitions, multi-object combinations, or single-object separations. This structure links each state change to the corresponding object, relevant actions, and visual observations supporting these associations. For each transition, the VLM infers potential causes from object and contextual information in connected nodes and edges, accounting for implicit actions whose effects are difficult to observe locally. The timestamps retained in the graph are used to locate visual evidence and verify the consistency between state differences across nodes and actions represented by edges, thereby correcting erroneous object–state associations and revising or rejecting unsupported causal explanations. To support this reasoning framework, we construct VERMOST, a multi-object multi-state dataset, which comprises 1,827 third-person cooking videos and 75,977 annotated state transition events. Annotations include objects, pre- and post-transition states, a temporal interval, relevant actions, and specific causes along with their types. VERMOST links states and transitions to video-supported causes, covering explicit actions, implicit actions, and their joint effects. Experiments on VERMOST show that our framework outperforms the baselines.

1 INTRODUCTION

Understanding object states in videos requires identifying what changes and explaining why it changes. Multiple objects may undergo transformations, and each can have several coexisting states: an egg mixture can be raw, liquid, and well mixed. Recent work advances state-change localization, multi-label state recognition, multi-object state prediction, and natural-language descriptions of changes (Xue et al., 2024; Tateno et al., 2025; Zameni et al., 2025; Nguyen et al., 2024). These capabilities provide a basis for understanding state transitions, but a recognized change must also be associated with the correct object and a specific cause supported by visual evidence.

Figure 1 illustrates incorrect object–state association and causal attribution. A salient action may occur close to a change without causing it, while the actual cause may originate in an earlier event and leave weak visual cues. Different actions can also explain different properties within one transition: stirring explains mixing, whereas heat transferred from hot cream explains chocolate melting.

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.

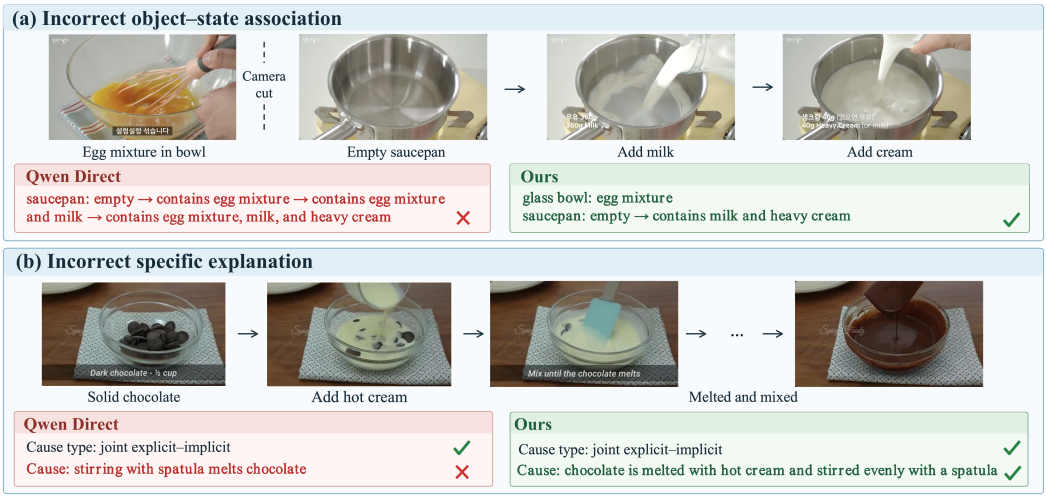

Figure 1: Association and attribution errors in multi-object multi-state transition understanding. (a) An egg mixture from an earlier event is incorrectly included in a saucepan's subsequent content states. (b) Melting is attributed to stirring without understanding that the heat transferred by hot cream is the cause of the state change. Understanding these transitions requires correct object–state associations and a specific cause supported by visual evidence.

Identifying the causes of state changes is particularly difficult when the relevant actions lack direct visual cues. Compared with actions such as cutting and stirring, the effects of heating, cooling, and evaporation may be harder to observe directly, and a substantial delay may separate the initiating operation from the observable state change. Models may therefore overlook these effects and attribute the change to a more salient action. To evaluate these different conditions, we refer to actions with directly visible manipulation cues, such as cutting and stirring, as *explicit actions*, and actions whose contributions may require temporal context and operating conditions to identify, such as heating, cooling, and evaporation, as *implicit actions*. When both types contribute to the target transition, we refer to *joint explicit–implicit actions*. This categorization enables separate evaluation of transition identification and causal attribution for each type, particularly changes involving implicit actions. Understanding these changes requires more than recognizing salient actions in the current event: the target object's pre- and post-transition states must be connected to relevant operations and conditions, including visual cues from earlier events.

To establish these associations, we propose a graph-guided visual evidence-based reasoning framework for understanding multi-object multi-state transitions in videos. Given a video, the framework identifies multiple objects and their state transitions and provides a specific cause supported by visual evidence for each transition. A state graph connects observations of objects across transformations, combinations, and separations. The VLM uses timestamped descriptions and sampled visual observations to verify object correspondence and complete state descriptions. This organization helps prevent states belonging to different objects from being compared as a single transition, while retaining information about related objects, earlier operations, and relevant conditions. It provides more comprehensive observational context for causal reasoning, particularly for implicit contributions difficult to identify from the current event alone.

For each transition, the framework separates changed properties from persistent ones, specifying what its cause must explain. Graph-associated observations supply relevant operations and conditions for generating candidate causes, considering explicit, implicit, and joint contributions when supported. The VLM compares these candidates with the state difference and recorded observations: an explanation must account for the changed properties, and its specific operations and conditions must agree with the observations. Local verification checks the predicted event's object, pre- and post-transition states, interval, and cause together, revising inconsistent predictions or rejecting unsupported events. Global verification then checks for duplicate transitions and inconsistent state sequences. Through this reasoning and verification, the framework links each identified object state transition to a specific cause supported by visual evidence.

To support this task, we construct the Video Evidence for Reasoning about Multi-Object State Transitions (VERMOST) dataset, comprising 1,827 third-person cooking videos and 75,977 object-level state-transition events. Its event-level annotations support joint evaluation of transition detection, state understanding, and causal attribution across the three action types. Our contributions are:

- We propose a graph-guided visual evidence-based reasoning framework that improves event precision and specific causal attribution by verifying object–state associations and explanations of target state differences.
- We introduce VERMOST, with linked event-level annotations of objects, coexisting states, temporal intervals, and causes supported by visual evidence across the three action types.
- Experiments on VERMOST demonstrate the effectiveness of our framework in state-transition detection and causal attribution compared with the baselines. They also highlight the value of VERMOST in revealing attribution errors that persist even when transitions are detected, particularly those involving implicit and joint explicit–implicit actions.

2 Related Work

Object state understanding captures object changes during activities, providing information beyond object identities and action labels. Early studies typically assume a single object state change per video, represented as an initial state, a state-modifying action, and a resulting state. Their temporal ordering supports joint state and action recognition (Alayrac et al., 2017; Souček et al., 2022; 2024), establishing action–state associations as a basis for understanding object changes.

Subsequent work expanded this formulation in two directions: improving generalization to unseen objects through shared state representations (Xue et al., 2024), and broadening state and object coverage to coexisting properties and multiple-object evolution (Tateno et al., 2025; Zameni et al., 2025). Natural-language descriptions further express states and action effects beyond predefined labels (Nguyen et al., 2024).

To support these richer settings, graph-based approaches encode transferable state–action relationships or connect object observations across transformations (Zang et al., 2026; Sun et al., 2026), while knowledge-guided methods improve zero-shot recognition by assessing knowledge reliability under visual and temporal constraints (Qi & Wu, 2026). These advances improve state representations and associations, but recognizing a change does not establish its specific cause. Our work jointly examines transition identification and whether visual evidence supports the specific operations and conditions proposed to explain each change.

Explaining why events occur requires reasoning about their relationships. Video question answering evaluates this ability through temporal and causal questions (Yi* et al., 2020; Xiao et al., 2021; Li et al., 2022). Subsequent benchmarks extend evaluation to compositions of object–action relations and dependencies between actions and their effects (Grunde-McLaughlin et al., 2021; Jia et al., 2022), connecting event reasoning with object state understanding.

Further developments address evidence access and reasoning reliability. Cross-time benchmarks examine whether models can connect observations across a video (Chen et al., 2024), while tree- and graph-based methods organize observations to retrieve question-relevant information (Wang et al., 2025; Malik et al., 2026). Complementary evaluations inspect intermediate reasoning and stepwise causal dependencies, rather than only final-answer accuracy (Gandhi et al., 2022; Li et al., 2026). These directions emphasize examining both the information used and the reasoning leading to an answer.

However, questions provide targets or relations to investigate, so correct answers do not establish that a model can independently discover and explain relevant transitions. Our task requires identifying multiple objects and their state transitions without reference objects, states, or intervals, and providing a specific cause supported by visual evidence for each transition. VERMOST jointly evaluates these abilities across explicit, implicit, and joint explicit–implicit actions. Our framework supports this task by linking state differences with relevant observations for causal reasoning and verification.

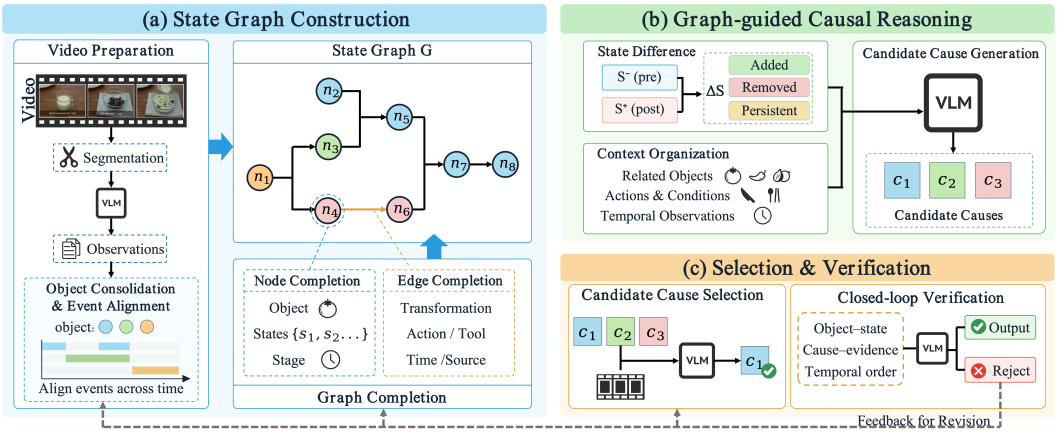

Figure 2: Overview of the graph-guided visual evidence-based reasoning framework. State graph construction connects object observations across transitions. State differences and related observations guide candidate cause reasoning. Local and global consistency checks refine transition predictions and their explanations.

3 METHOD

Understanding object state transitions requires maintaining object correspondence and identifying the operations and conditions responsible for each change. These requirements are difficult to satisfy locally: an object's appearance may change during transformation, while evidence of its cause may occur earlier than the visible result. A model may therefore associate states with the wrong object or explain a change using a salient but unrelated action. To address these difficulties, we propose a graph-guided visual evidence-based reasoning framework. Given a video V, our task is to identify multiple objects and their state transitions and provide a specific cause supported by visual evidence for each transition. The output is a set of object-level state-transition events $\mathcal{Y} = \{y_i\}_{i=1}^{N}$, where

$$\begin{aligned} y_i &= (o_i, \mathcal{S}_i^-, \mathcal{S}_i^+, \tau_i, a_i, u_i, c_i, z_i), \\ \tau_i &= [t_i^s, t_i^e]. \end{aligned} \tag{1}$$

Here, o_i denotes the target object; $\mathcal{S}_i^-$ and $\mathcal{S}_i^+$ are its pre- and post-transition state sets; τ_i is its interval in the original video; a_i and u_i denote associated actions and tools; and c_i and z_i denote the specific cause and its type. State sets represent coexisting properties. Cause types comprise explicit actions, implicit actions, and joint explicit–implicit actions. Reference objects, states, actions, and event intervals are not supplied during inference.

The framework comprises three stages (Figure 2). First, state graph construction establishes object correspondence and organizes states for comparison across a transition. Graph-guided candidate cause reasoning then identifies changed properties and connects them to relevant observations, including earlier operations with later visible effects. Finally, cause selection and consistency verification examine whether an explanation accounts for the changes and agrees with the recorded observations, while revisiting its underlying object, state, and temporal predictions. The same pretrained Qwen3.5-9B (Qwen Team, 2026) performs all stages through role-specific prompts without parameter updates. Appendix A provides the information flow, prompt summaries, and a worked example.

3.1 STATE GRAPH CONSTRUCTION

Nearby observations may concern different objects, while ingredients can change appearance or enter a mixture. We therefore use a state graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ to maintain correspondence through these transformations. Each node $n_k = (o_k, \mathcal{S}_k, \sigma_k)$ records an object, its coexisting states, and its stage interval; links represent single-object transformations, multi-object combinations, or single-object separations.

The graph combines timestamped observations $\mathcal{Q} = \{q_j\}_{j=1}^{M}$, where $q_j = (\sigma_j, d_j)$ contains an interval and a description of objects, states, actions, and tools, with sampled visual evidence. Visual checks examine object associations, combinations, separations, and temporal order; state completion supplies the corresponding pre- and post-transition properties. Repeated observations of the same transition are consolidated; distinct changes are retained. The graph supports state comparison and preserves source observations for cause reasoning.

3.2 Graph-Guided Candidate Cause Reasoning

Not every property changes. After state completion, we normalize and match pre- and post-transition state expressions to identify changes requiring explanation. Let $\mathcal{M}_i$ contain matched pairs describing the same persistent property. The removed, added, and persistent states are

$$\begin{aligned} \Delta_i^- &= \{s \in \mathcal{S}_i^- \mid s \text{ is unmatched in } \mathcal{M}_i\}, \\ \Delta_i^+ &= \{s \in \mathcal{S}_i^+ \mid s \text{ is unmatched in } \mathcal{M}_i\}, \\ \mathcal{P}_i &= \{s^+ \mid (s^-, s^+) \in \mathcal{M}_i\}. \end{aligned} \tag{2}$$

A cause must account for (Δ_i^-, Δ_i^+); persistent properties in $\mathcal{P}_i$ are not new effects to explain. Appendix A.3 illustrates this distinction.

To identify a change's source, the graph connects the target transition to context $\mathcal{H}_i$ containing related objects, operations, tools, and conditions from transformation records and temporal observations. Earlier operations help explain implicit actions with later visible effects. For example, earlier observations of adding hot cream can identify the heat source for melting, whereas concurrent stirring alone cannot explain melting.

Given the state difference and $\mathcal{H}_i$, the VLM proposes candidates $\mathcal{C}_i = \{c_{i,m}\}_{m=1}^{K_i}$. The prompt considers explicit, implicit, and joint contributions, generating two or three observation-supported alternatives. Each candidate specifies operations, conditions, and supporting observations. Candidates may explain different parts of the target change; they are alternatives, not required explanations for each cause type.

3.3 Cause Selection and Consistency Verification

A plausible cause may name operations absent from the video or explain only part of the state difference. The VLM therefore compares candidate causes with the target pre/post states and timestamped observation descriptions, checking both changed-property coverage and agreement with the observed operations and conditions. It selects or revises a specific cause supported by visual evidence, c_i, and assigns its type z_i.

Selecting a cause is insufficient if the transition itself contains an incorrect object–state association or interval. Local consistency verification therefore checks the event's object, states, interval, and cause together against associated and neighboring observations. Conflicts prompt reconsideration and revision of relevant predictions, or rejection of unsupported events. Individually plausible events may still duplicate transitions or describe conflicting state sequences, so global verification compares retained events along the full-video timeline. Appendix A.4 details these checks and their feedback.

The final output comprises revised object-level state-transition events with specific causes supported by visual evidence. Maintaining object correspondence, linking changed properties to relevant observations, and jointly checking transitions and explanations addresses what changed and why. Earlier observations support reasoning about implicit contributions difficult to identify locally. Appendix A.5 provides an example.

4 VERMOST Dataset

Existing datasets offer limited support for jointly evaluating multi-object state transitions and their specific causes. To address this gap, we introduce VERMOST, which links coexisting pre- and post-transition states to a specific cause supported by visual evidence, covering explicit, implicit, and joint explicit–implicit actions.

Table 1: Comparison of dataset scale and state–cause annotations.

Dataset	MRA	Multi-object	Multi-state	Open vocab.	Pre/post sets	Cause expl.	Cause types
ChangeIt (Souček et al., 2022)	667 V	×	×	×	×	×	×
HowToChange (Xue et al., 2024)	5,424 C	×	×	×	×	×	×
MOST (Tateno et al., 2025)	61 V	×	✓	×	×	×	×
MOSCATO (Zameni et al., 2025)	111 V	✓	✓	×	×	×	×
OSCaR (Nguyen et al., 2024)	500 C	△	△	✓	△	△	×
ProcObject-10K (Guo & Kong, 2025)	10,522 QA	△	△	✓	△	△	×
VERMOST	**75,977 E**	✓	✓	✓	✓	✓	✓

MRA: manually reviewed annotations; Multi-object means recording states separately for multiple objects; multi-state means allowing properties to coexist for one object; Pre/post sets denotes explicitly paired state sets for the same transition; Open vocab.: open vocabulary refers to free-text object and state expressions, rather than generalization to unseen objects alone; V/C/E/QA: videos, clips, object-level transition events, and question–answer pairs; Cause expl.: cause explanations; Cause types distinguish explicit, implicit, and joint explicit–implicit contributions; ✓: dedicated annotation; △: information conveyed in descriptions or QA without equivalent dedicated structured annotation; ×: no dedicated annotation of the specified feature.

4.1 Data Construction and Annotation

We collect third-person cooking videos from YouTube, where ingredients change form, combine, and become intermediate products. After removing duplicate, corrupted, and finished-product-only videos, human reviewers inspect the source videos and neighboring events to correct objects, states, causes, and temporal boundaries, add missing events, and consolidate duplicates. Each reviewed event links a target object, coexisting pre- and post-transition state sets, an interval, relevant actions and tools, and a specific cause with its type. Further details on event definitions, cause types, and the annotation and review procedure are provided in Appendix B.

4.2 Comparison with Existing Datasets

Table 1 compares VERMOST with existing datasets. VERMOST provides manually reviewed annotations covering 1,827 videos and 75,977 events, exceeding the manually annotated counts reported for the other datasets in the table. The review covers the objects, pre- and post-transition states, temporal intervals, and specific causes of all 75,977 object-level state-transition events, enabling evaluation of both what changed and why it changed. Cause annotations distinguish explicit, implicit, and joint explicit–implicit contributions, supporting separate analysis of transitions whose causes are difficult to identify from local observations alone.

4.3 Dataset Statistics and Annotation Example

The left panel of Figure 3 shows 60,953 explicit events (80.23%), 2,677 implicit events (3.52%), and 12,347 joint events (16.25%). The latter two groups comprise 19.77% of events, enabling evaluation of transitions involving implicit actions. This class imbalance also motivates reporting cause-type macro-F1, which gives equal weight to the three classes.

Videos total 205.38 hours, with mean and median durations of 6.74 and 6.02 minutes and an average of 41.59 object-level transitions per video. Multiple transitions over several minutes require maintaining object correspondence and connecting changes to relevant observations across time.

The right panel illustrates how VERMOST separately records changes to different objects within the same video interval. Each object-level event pairs pre- and post-transition states with a specific cause and its type, allowing evaluation of whether a model associates each change with the correct object and explains the actions responsible for it.

4.4 Evaluation Protocol

Within each video, predictions are matched one-to-one with reference events using temporal overlap and object correspondence. Event precision, recall, and F1 measure prediction reliability and transition coverage. Pre/post state F1 evaluates the states before and after a transition, while state-

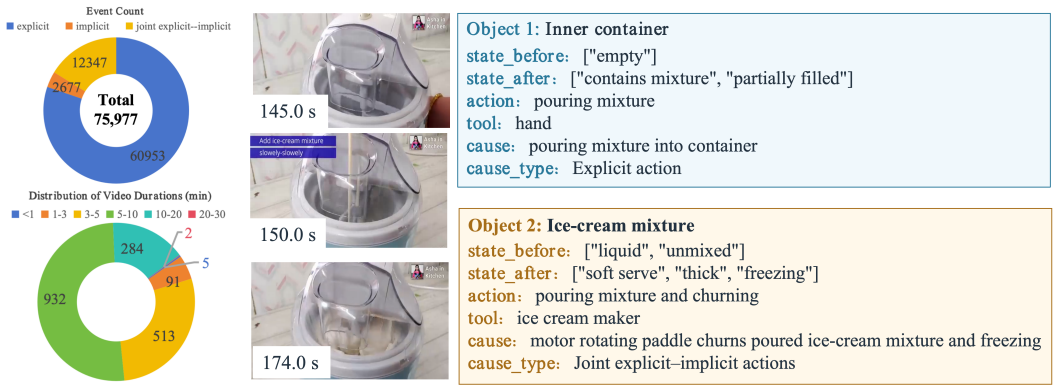

Figure 3: Overview of VERMOST. Left: event counts by cause type and the distribution of video durations. Right: an annotation example showing the state transitions of two objects within the same video interval.

difference F1 evaluates added and removed properties. Cause-type macro-F1 weights the three classes equally. State and cause metrics are computed on matched events and interpreted alongside detection performance.

We also report a 0–10 Cause Score measuring coverage of specific operations, conditions, and causal relationships in the reference explanation, with penalties for contradictions. A VLM evaluates these aspects without video input, and a fixed rule computes the score. Details of event matching, Cause Score computation, and consistency diagnostics are provided in Appendix C.

5 Experiments

All stages use the same pretrained Qwen3.5-9B checkpoint (Qwen Team, 2026) and inference settings, with role-specific prompts and no parameter updates. We evaluate on the full VERMOST dataset without providing reference objects, states, actions, or event intervals during inference. We use the event, state, and cause metrics defined in Section 4.4.

5.1 Comparison with Existing Methods

To evaluate the framework's effectiveness in identifying object state transitions and explaining their causes, we compare it with Qwen2.5-VL-7B (Bai et al., 2025), Video-R1-7B (Feng et al., 2025), VideoLLaMA2-7B (Cheng et al., 2024), VideoLLaMA3-7B (Zhang et al., 2025), and VideoChat-R1.5 (Yan et al., 2026), together with adapted X-CLIP (Ni et al., 2022) and OSCaR + Qwen (Nguyen et al., 2024). We also compare with direct prediction using Qwen3.5-9B, denoted *Qwen Direct*, to examine the benefit of the framework with the same pretrained backbone. Baseline adaptations are described in Appendix D.

Table 2 shows that our framework achieves the highest event precision and F1, all three state F1 scores, cause-type macro-F1, and specific cause score. These results indicate more reliable transition predictions and more accurate state descriptions and causal attribution, consistent with the framework's design of organizing object–state associations and checking explanations against state differences and related observations. However, event recall remains below Qwen Direct, possibly because some true transitions are missed during graph construction or rejected during verification due to incomplete supporting observations. Overall, these results support jointly identifying object state transitions and explaining their specific causes, while highlighting missed transitions as an area for improvement.

5.2 Transitions Involving Implicit Actions

To evaluate how the framework explains state changes involving implicit actions, we compare methods on implicit and joint explicit–implicit transitions. Table 3 reports recall, weighted type F1, and

Table 2: Overall comparison on VERMOST. Detection, state, and cause-type metrics are reported in percentages.

Method	Event Detection			State Understanding			Causal Attribution	
	P ↑	R ↑	F1 ↑	Pre F1 ↑	Post F1 ↑	Δ F1 ↑	Type mF1 ↑	Cause Score ↑
Qwen2.5-VL-7B	34.67	31.07	32.77	5.56	3.56	4.40	32.60	2.47
Video-R1-7B	36.19	32.50	34.25	1.21	0.81	0.98	35.86	2.61
VideoLLaMA2-7B	6.82	20.70	10.26	13.55	8.59	10.08	35.53	1.95
VideoLLaMA3-7B	20.12	6.34	9.64	22.36	13.88	17.37	33.60	1.92
VideoChat-R1.5	22.80	36.43	28.05	7.21	5.59	6.11	45.00	2.54
X-CLIP	4.30	7.89	5.57	1.83	2.52	1.72	33.85	N/A
OSCaR + Qwen	40.52	51.81	45.47	6.11	3.90	4.78	46.90	3.71
Qwen Direct	41.55	**57.63**	48.29	64.18	64.91	62.87	47.28	7.17
Ours	**78.06**	54.20	**63.98**	**68.34**	**67.40**	**63.24**	**65.09**	**8.21**

Table 3: Summary performance on transitions involving implicit actions, including implicit and joint explicit–implicit causes.

Method	Recall ↑	Weighted Type F1 ↑	Weighted Cause Score ↑
Qwen2.5-VL-7B	34.66	6.96	1.65
Video-R1-7B	37.05	15.55	1.40
VideoLLaMA2-7B	22.88	11.97	1.81
VideoLLaMA3-7B	6.95	2.46	1.85
VideoChat-R1.5	38.18	24.82	1.07
X-CLIP	5.73	2.74	N/A
OSCaR + Qwen	32.90	23.67	4.00
Qwen Direct	42.38	22.16	5.20
Ours	**43.69**	**61.05**	**7.88**

weighted Cause Score. Our framework achieves the highest values on all three metrics, with particularly pronounced improvements in cause-type recognition and specific explanation quality over Qwen Direct. These results support the effectiveness of the framework in understanding transitions involving implicit actions. By linking the target change to earlier observations and related objects, the state graph makes relevant operations and conditions available for causal reasoning. Comparing candidate causes against these observations and the changed properties helps the model explain implicit contributions beyond the most salient visible action.

5.3 Consistency between State Changes and Causes

To examine whether correctly identifying changes or cause types also leads to adequate specific explanations, we compare Qwen Direct, OSCaR + Qwen, and our framework on events matched and successfully scored by all three methods. Evaluating the same events reduces differences due to detecting different transition sets. We define three binary indicators: $I_{\Delta} = 1$ when the prediction fully covers the reference state difference; $I_{\text{type}} = 1$ when the binary cause type is correctly identified (explicit versus implicit or joint explicit–implicit); and $I_{\text{cause}} = 1$ when the specific cause score is at least 5. We report $P(I_{\text{cause}} = 1 \mid I_{\Delta} = 1)$ and $P(I_{\text{cause}} = 1 \mid I_{\text{type}} = 1)$ to assess whether models can explain the specific operations and conditions responsible for a change once its reference state difference is fully covered or its cause type is correctly identified. Appendix C.3 details these indicators.

Table 4 shows that our framework achieves the highest complete change coverage and explanation pass rate on the events matched by all three methods. When the binary cause type is correctly identified, its conditional explanation pass rate also exceeds both baselines, supporting the benefit of checking specific explanations against state differences and related observations. However, for all three methods, neither complete change coverage nor a correct cause type guarantees a passing explanation. These results show that recognizing the relevant changes does not necessarily yield an adequate explanation of their specific causes, highlighting the value of jointly evaluating state understanding and causal explanation.

Table 4: State–cause consistency on events matched and successfully scored by all three methods. All values are percentages; higher is better. Best and second-best results for each metric are shown in bold and underlined, respectively.

Metric	Qwen Direct	OSCaR + Qwen	Ours
$P(I_{\Delta} = 1) \uparrow$	<u>49.19</u>	1.62	**61.08**
$P(I_{\text{cause}} = 1) \uparrow$	<u>79.46</u>	37.30	**87.57**
$P(I_{\text{cause}} = 1 \mid I_{\Delta} = 1) \uparrow$	<u>96.70</u>	66.67	**97.35**
$P(I_{\text{cause}} = 1 \mid I_{\text{type}} = 1) \uparrow$	<u>80.11</u>	37.65	**89.14**

Table 5: Ablation study of the reasoning framework.

Variant	Event Detection			State	Cause Score		
	P ↑	R ↑	F1 ↑	Δ F1 ↑	Overall ↑	Implicit ↑	Joint ↑
w/o State Graph	42.05	**58.64**	<u>48.98</u>	<u>52.88</u>	5.31	5.90	5.14
w/o Structured Context	<u>64.34</u>	31.91	42.66	42.99	<u>6.70</u>	7.60	4.78
w/o Candidate Reasoning	63.92	31.82	42.49	42.56	4.32	3.79	2.81
w/o Closed-loop Verification	61.10	36.55	45.74	43.56	6.53	<u>7.61</u>	<u>5.23</u>
Full Framework	**78.06**	<u>54.20</u>	**63.98**	**63.24**	**8.21**	**7.88**	**7.88**

5.4 Ablation Study

To examine how the framework's components contribute to transition identification and causal explanation, we remove the following components and compare each variant with the full framework: (1) *State Graph*, removing the state graph construction stage; (2) *Structured Context*, removing the organized graph-associated context supplied to cause reasoning; (3) *Candidate Reasoning*, removing the stage that generates alternative causes; and (4) *Closed-loop Verification*, removing consistency checking and prediction revision. All variants use the same backbone and evaluation protocol; detailed settings are provided in Appendix D.3.

Table 5 shows that the full framework achieves the best event F1, state-difference F1, and specific cause scores, indicating that each component contributes to jointly identifying transitions and explaining their causes. However, recall increases when the state graph is removed. This suggests that graph construction helps distinguish genuine object state transitions from incorrect associations, but may also miss some true transitions. Removing candidate cause reasoning yields the lowest specific cause scores, highlighting the importance of considering alternative explanations and assessing whether they explain the target change; missing explanations also contribute to this decline. Removing structured context particularly reduces joint-cause scores, further supporting the use of related operations and conditions to explain their combined effects. Finally, removing verification lowers both precision and recall, suggesting that its revisions may help recover correct event correspondences as well as remove unsupported predictions. Together, these results support the complementary roles of the framework's components in understanding object state transitions and their specific causes.

6 Conclusion

We study visual evidence-based reasoning for understanding multi-object multi-state transitions in videos, requiring models to identify transitions and provide a specific cause supported by visual evidence for each transition. Our framework connects object–state relations, state differences, and causal context, then verifies the resulting events and explanations. VERMOST supports joint evaluation of these tasks through event-level state, temporal, and cause annotations. Experiments show that our framework improves state-transition detection and causal attribution, with particularly notable gains for transitions involving implicit actions. However, some valid transitions may be omitted during graph construction or rejected during verification when supporting observations are incomplete. Future work will investigate how to recover these transitions by incorporating additional relevant observations while maintaining prediction reliability.

AI USE STATEMENT

We used VLMs to assist with video processing and candidate annotation generation. Human reviewers made the final decisions on accepting, correcting, or removing annotations. VLMs are also part of the reasoning framework, and Qwen3.5-9B provides reference-based specific cause scores using a fixed rubric. These scores are model-based assessments rather than human judgments.

Generative AI tools assisted with manuscript organization, bilingual drafting, language editing, and LaTeX formatting. The authors are responsible for checking the final technical content, reference annotations, experimental claims, and AI-assisted artifacts.

REPRODUCIBILITY STATEMENT

We plan to publicly release VERMOST and the code for reproducing our framework and evaluation. Implementation details, prompts, and evaluation protocols are provided in the appendix.

REFERENCES

Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2127–2136, 2017.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. 2025. URL https://arxiv.org/abs/2502.13923.

Jr-Jen Chen, Yu-Chien Liao, Hsi-Che Lin, Yu-Chu Yu, Yen-Chun Chen, and Yu-Chiang F Wang. Rextime: A benchmark suite for reasoning-across-time in videos. volume 37, pp. 28662–28673, 2024.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. URL https://arxiv.org/abs/2406.07476.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 99114–99137. Curran Associates, Inc., 2025. doi: 10.52202/085713-3310. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/8eb3976840e08b80dda9667562574246-Paper-Conference.pdf.

Mona Gandhi, Mustafa Omer Gul, Eva Prakash, Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Measuring compositional consistency for video question answering. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5036–5045. IEEE, 2022.

Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11287–11297, June 2021.

Wenliang Guo and Yu Kong. Procobject-10k: Benchmarking object-centric procedural understanding in instructional videos. *arXiv preprint arXiv:2512.03479*, 2025.

Baoxiong Jia, Ting Lei, Song-Chun Zhu, and Siyuan Huang. Egotaskqa: Understanding human tasks in egocentric videos. volume 35, pp. 3343–3360, 2022.

Jiangtong Li, Li Niu, and Liqing Zhang. From representation to reasoning: Towards both evidence and commonsense reasoning for video question-answering. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21241–21250. IEEE, 2022.

Xuchen Li, Xuzhao Li, Shiyu Hu, Kaiqi Huang, and Wentao Zhang. Causalstep: A benchmark for explicit stepwise causal reasoning in videos. 40(8):6530–6538, 2026.

Sameer Malik, Ayush Singh, Moyuru Yamada, and Dishank Aggarwal. Ravu: Retrieval augmented video understanding with compositional reasoning over graph. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2869–2878. IEEE, 2026.

Nguyen Nguyen, Jing Bi, Ali Vosoughi, Yapeng Tian, Pooyan Fazli, and Chenliang Xu. Oscar: Object state captioning and state change representation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3565–3576, 2024.

Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *European conference on computer vision*, pp. 1–18. Springer, 2022.

Yayun Qi and Xinxiao Wu. What to trust? a trust-aware knowledge-guided method for zero-shot object state understanding in videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8484–8492, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Look for the change: Learning object states and state-modifying actions from untrimmed web videos. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13936–13946. IEEE, 2022.

Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Multi-task learning of object states and state-modifying actions from web videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(7):5114–5130, 2024.

Yihong Sun, Xinyu Yang, Jennifer Sun, and Bharath Hariharan. Tracking and understanding object transformations. volume 38, pp. 81039–81065, 2026.

Masatoshi Tateno, Takuma Yagi, Ryosuke Furuta, and Yoichi Sato. Learning multiple object states from actions via large language models. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 9555–9565. IEEE, 2025.

Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3272–3282. IEEE, 2025.

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. IEEE, 2021.

Zihui Xue, Kumar Ashutosh, and Kristen Grauman. Learning object state changes in videos: An open-world perspective. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18493–18503. IEEE, 2024.

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. volume 38, pp. 119152–119184, 2026.

Kexin Yi*, Chuang Gan*, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HkxYzANYDB`.

Parnian Zameni, Yuhan Shen, and Ehsan Elhamifar. Moscato: Predicting multiple object state change through actions. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11600–11611. IEEE, 2025.

Yuan Zang, Zitian Tang, Junho Cho, Jaewook Yoo, and Chen Sun. Sage: A unified framework for generalizable object state recognition with state-action graph embedding. volume 38, pp. 59742–59764, 2026.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

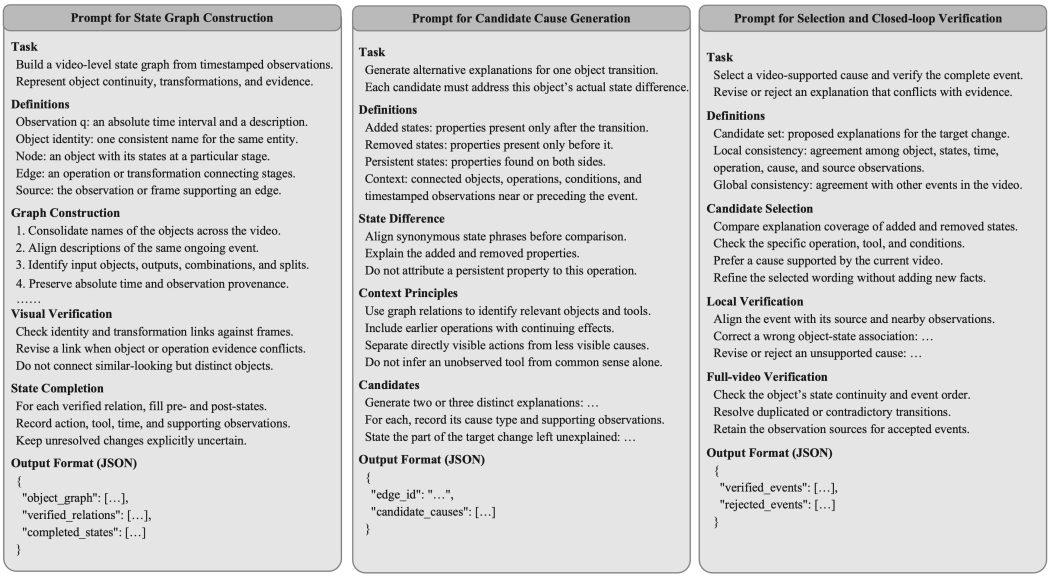

Figure 4: Module-level prompt summaries for the three reasoning stages. Ellipses indicate omitted content.

A Reasoning Framework Details

A.1 Model Roles and Information Flow

All three stages use the same pretrained Qwen3.5-9B checkpoint with role-specific prompts. Outputs from earlier stages become inputs to later stages; no model parameters are updated. Table 6 summarizes the information flow rather than verbatim prompts.

Table 6: Inputs and outputs of the three reasoning stages.

Stage	Inputs	Outputs
State graph construction	Timestamped descriptions and sampled visual observations	Object–state nodes, transformation relations, and source references
Candidate cause reasoning	Target states, state differences, and related observations	Alternative causes specifying operations, conditions, and supporting observations
Selection and verification	Candidates, event fields, associated observations, and the timeline	Selected or revised causes and events; local/global consistency decisions

A.2 Module-Level Prompt Summaries

Figure 4 summarizes the instructions for state graph construction, candidate cause generation, and selection and verification. These module-level summaries combine instructions from separate calls and omit examples and implementation details; they are not verbatim prompts or single-call implementations.

A.3 State Comparison

State differences are computed after state completion and normalization. For example, a transition from {whole, solid} to {halved, solid} removes *whole*, adds *halved*, and preserves *solid*. A cause must explain the division rather than the persistent property. Completion supports this comparison, but an omitted description does not by itself establish that a property disappeared. Remaining omissions can therefore affect state-difference accuracy.

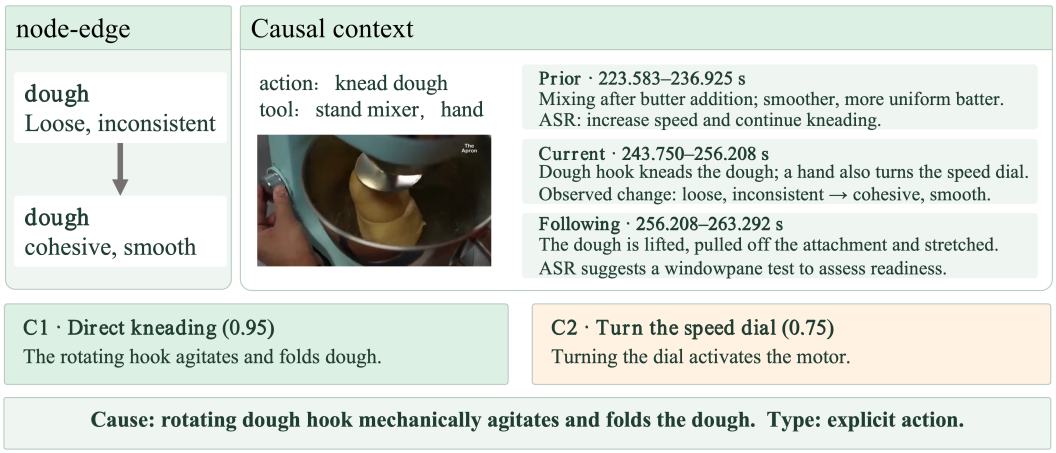

Figure 5: Example of visual evidence-based cause reasoning. The state graph supplies the target change, timestamped observations provide context, and candidate comparison yields a specific cause associated with the observed tool–object interaction. The image's label "action" denotes an explicit action.

A.4 Evidence Checks and Feedback

Graph construction uses sampled visual observations together with timestamped descriptions. Candidate reasoning and cause selection use the recorded descriptions; their agreement checks consistency, rather than independently confirming every statement against video frames. Feedback changes predicted records, not model parameters; local and global checks address within-event and between-event errors, respectively.

A.5 Example of Visual Evidence-based Cause Reasoning

Figure 5 illustrates how the framework derives a specific cause supported by visual evidence. The graph associates the dough's pre/post states, identifying the change from loose and inconsistent to cohesive and smooth. Timestamped earlier, current, and subsequent observations provide the context for explaining this difference.

The VLM considers kneading by the dough hook and adjustment of the mixer speed. It compares the candidates with the target change and observed operations: mechanical agitation and folding describe how the hook acts on the dough, whereas dial adjustment alone does not explain the transformation. The selected explanation names the tool, action, and affected object, followed by consistency checking of the event. Speech transcripts provide supplementary context; visual observations support the tool–object interaction. This illustrates how state differences define the explanatory target and related observations support a specific cause. The selected operation is an explicit action under our taxonomy.

B Dataset Annotation Details

B.1 Event Semantics and Cause Types

An event describes one target object's transition and associates its pre/post state sets, original-video interval, actions, tools, and specific cause. State sets contain coexisting properties: *raw*, *liquid*, and *well mixed* may describe the same egg mixture at one stage. The transition interval need not contain all causal observations, which can occur earlier. Combination and separation relations connect participating objects across events; they do not require every event to have several target objects.

Cause types concern contributions to the target change, not merely simultaneously visible actions. Cutting and stirring are explicit manipulations. Heating, cooling, and evaporation are implicit ac-

tions whose contributions can require earlier observations and operating conditions. Visible heating equipment can support an implicit explanation without making heat transfer a direct manipulation.

The following examples illustrate the definitions:

- Cutting whole dough into halves is explicit; solidity can persist through the transition.
- Heat transferred from added hot cream can explain chocolate melting, an implicit heating effect; concurrent stirring alone does not explain melting.
- When the target change includes both mixing and melting, stirring and heat transfer explain different changed properties, giving a joint explicit–implicit cause.

These distinctions make the state difference important: the same surrounding operations can have different explanatory roles depending on which properties change.

B.2 Construction and Human Review

We collect third-person cooking videos from YouTube and remove duplicate, corrupted, and finished-product-only videos. Each retained video receives a unique identifier and initial scene intervals with aligned speech transcripts. These intervals support inspection, but are not fixed event boundaries: one scene can contain several changes, and one change can continue across scenes.

Qwen3.5-9B generates initial object, action, tool, state, and cause annotations for human review and temporal boundary refinement. Reviewers inspect target intervals, neighboring events, and original videos, using aligned speech as supplementary context. They correct unsupported fields, add missing events, merge repeated descriptions, and separate distinct transitions. Specific operations and conditions are retained only when supported by the available evidence.

After human review, the reviewed annotations are organized into structured records without adding new inferences. Following expression normalization and completeness checks, these records are exported chronologically and by object. These are two views of the same events, not additional samples. Final reviewed reference annotations are not provided to the prediction framework.

Figure 6 summarizes the prompts for automatic annotation and structured output generation. The left prompt generates initial annotations for human review; the right prompt organizes the reviewed annotations into structured records.

The right-hand prompt in Figure 6 takes human-reviewed annotations as input and organizes them into structured event records. References to video content in this step concern information already recorded and reviewed, rather than new inference from the video. This step preserves the reviewed objects, states, intervals, and causes without adding unsupported content or changing annotation decisions.

C Evaluation Protocol and Consistency Diagnostics

C.1 Matched-Event Evaluation

Predictions are matched one-to-one to reference events within each video using temporal overlap and object correspondence. A match identifies the corresponding event; it does not establish that its states or cause are correct. Each method's state and cause scores are computed on its own matched events. A method detecting fewer, easier transitions can therefore obtain high conditional scores, so precision and recall must be considered alongside them. The analysis of events matched by all compared methods compares the same reference transitions but remains conditional on recovery by all compared methods.

C.2 Specific Cause Score

We use `gold_cause_coverage_0_10_v2`. We use Qwen3.5-9B (Qwen Team, 2026) as the evaluator, with temperature set to 0, applying the same scoring prompt and rule to all methods. Inputs include predicted and reference explanations and their event context, but exclude video frames,

Prompt for Video Observation

Task
Describe a short, automatically segmented video.
Produce observations for later event merging and splitting.

Definitions
A state change modifies an object property.
Position changes and camera cuts alone do not qualify.
Direct action: visible human or tool operation.
Physical cause: heating, cooling, dissolving, or a related condition that changes the object.

Decision Principles
1. Record sequential changes of the same object separately.
2. Group simultaneous changes from one shared cause.
3. Mark an interval with no clear state change.
4. Distinguish action and physical cause when possible.
5. Do not invent an event that occurs after this clip.
6. Use unknown for a cause that cannot be determined.

Required Observations
List objects, actions, tools, and transition times.
Record before and after states and evidence type.
Provide a short reason for a split or merge decision.

Output Format (JSON)

```
{
 "clip_summary": "…",
 "state_change_events": […],
 "simultaneous_event_groups": […],
 "clip_decision": {…}
}
```

Prompt for Structured Dataset Annotation

Task
Annotate object states, state transitions, and their causes for one re-cut video event in English.

Definitions
Object: a concise canonical name for the target entity.
State: a short concrete property; several may coexist.
Cause: a specific action or physical condition with tool.
Evidence type: visible, strongly inferred, unsupported, or unknown, according to the available observations.

Annotation Principles
1. Keep only visible or strongly inferable changes.
2. Do not mark every container change as a core event.
3. Use the original video's absolute time interval.
4. Label explicit, implicit, joint, or unknown cause.
5. Group objects changed by one shared operation.
6. Prefer existing vocabulary terms and record aliases.
7. Flag uncertain tools, causes, or boundaries.

Object-level Record
object, state_before, state_after: …
action, tool, specific cause, cause type: …
time, evidence type, reason: …

Output Format (JSON)

```
{
 "absolute_time_sec": […],
 "object_states": […],
 "state_changes": […],
 "event_group": {…},
 "new_vocab_candidates": {…},
 "needs_manual_check": …
}
```

Figure 6: Prompt summaries for automatic annotation and structuring human-reviewed annotations. Ellipses indicate omitted content.

Table 7: Dimensions and weights for reference-cause coverage.

Dimension	Weight	Content assessed
Objects and states	2	Objects, materials, and relevant pre/post states.
Actions and processes	3	Operations, physical processes, and causal mechanisms.
Roles and relations	3	Acting/affected entities, source/destination, tool–object and action–state correspondences, and temporal/causal relations.
Specific details	2	Tools, containers, locations, conditions, manner, and sequence specified in the reference.

method names, and cause-type labels. Context fields resolve references and check consistency, but do not count as facts expressed in the prediction.

The evaluator splits the reference into minimal meaningful facts and returns coverage in the four dimensions of Table 7. Synonyms, paraphrases, and unambiguous pronouns are accepted; length, lexical overlap, and cause-class agreement do not determine the score. Details absent from the reference are not required. Actions/processes and causal relations receive higher weights because they specify what produces the change and how the entities are connected.

For dimension k, let $r_k \in [0, 1]$ be the returned coverage; a dimension with no applicable reference fact is `null`. Let $\mathcal{A}$ be the non-null dimensions and w_k their weights. The evaluator returns a contradiction penalty $p \in [0, 10]$, and the program computes

$$s = \max\left(0,\ 10\frac{\sum_{k\in\mathcal{A}} w_k r_k}{\sum_{k\in\mathcal{A}} w_k} - p\right). \tag{3}$$

Null dimensions are excluded from both sums. Omissions reduce coverage and are not penalized twice. Additional details earn no bonus and incur no penalty unless they contradict the reference or context. The penalty is 0 for no conflict, 1–2 for minor conflicts, 3–5 for materially wrong

tools/states/roles, and 6–10 for a wrong central action/process, reversed causal direction, or an incompatible cause. The evaluator also returns covered facts, missing facts, incorrect claims, and a rationale; each conflict must identify the predicted claim and reference fact.

The score measures reference agreement, not independent visual verification. Concealing method identity does not eliminate potential evaluator preferences for particular wording or explanation structures. State and cause metrics also overlap because cause coverage includes objects and states. Their joint diagnostics describe output consistency rather than statistical independence.

Figure 7 summarizes the evaluator instructions. The returned coverage and contradiction penalty are converted into the final Cause Score using Equation 3.

Prompt for Cause Explanation Scoring

Task

Score the predicted cause for one matched object-state transition.
Use the reference annotation and the available video evidence.
Return one score on the 0–10 scale with a short explanation.

Definitions

Target transition: the matched object and its pre/post states.
State difference: properties added or removed by this event.
Specific cause: the operation and conditions that explain
the target difference in this particular video.

Scoring Principles

1. Check whether the explanation concerns the correct object.
2. Check coverage of the changed properties.
3. Check the operation, tool, and relevant conditions.
4. Penalize a plausible cause unsupported by observations.
5. Do not award credit for describing only persistent states.
6. Handle missing or uncertain visual evidence: …

Score Anchors

0: …
Intermediate scores: …
10: …

Instructions

Compare the prediction with the matched Gold event: …
Assess cause specificity and video consistency: …
Return a score and a concise justification: …

Output Format (JSON)

```
{
 "cause_score": …,
 "reason": "…"
}
```

Figure 7: Summary of the reference-based cause explanation scoring prompt. The evaluator returns coverage and contradiction assessments rather than directly assigning the final score.

In Figure 7, "the available video evidence" refers to video-derived evidence recorded in the reference explanation and event context, against which the predicted explanation is evaluated. No video frames are provided to the evaluator. The evaluator returns coverage scores for four dimensions and a contradiction penalty; the program then computes the final 0–10 Cause Score. Penalties apply only

to claims that contradict the reference explanation or event context. Omissions reduce coverage and are not penalized again.

C.3 Consistency Diagnostics

The analysis in Section 5.3 uses reference events matched and successfully scored by all three methods; failed scoring requests are excluded. The indicator I_{Δ} measures full coverage of the reference state difference and does not penalize extra predicted states. The indicator I_{type} distinguishes explicit causes from causes involving implicit actions, combining implicit and joint labels only for this binary diagnostic. The indicator I_{cause} uses a Cause Score threshold of 5, the midpoint of the 0–10 scale, as an operational pass criterion rather than a guarantee of factual correctness or completeness. A score below 5 can reflect missing reference facts, contradictions, or both.

The marginal rates use all events in this common set. Each conditional rate uses only events satisfying its condition as the denominator. For $P(I_{\text{cause}} = 1 \mid I_{\Delta} = 1)$, Qwen Direct and our framework have the same number of failures but different denominators; their rate difference does not indicate fewer absolute failures for our framework.

D Baseline Adaptation and Ablation Scope

D.1 X-CLIP Adaptation

X-CLIP (Ni et al., 2022) matches video content to candidate text rather than generating event records. Qwen3.5-9B first describes video segments; extracted object and state expressions are normalized and deduplicated into a candidate vocabulary. X-CLIP matches visual inputs to these prompts and prompts for the three cause types. Temporal comparison of predicted states is used to construct object-level transitions in the common event format. Qwen supplies candidates and X-CLIP performs visual matching, so the baseline evaluates a Qwen-assisted system. Cause fields contain fixed type-dependent templates rather than explanations of specific operations and conditions; the specific cause score is therefore N/A.

D.2 OSCaR + Qwen Adaptation

OSCaR (Nguyen et al., 2024) generates object-centered descriptions rather than complete event records. We obtain object and temporal candidates from automatic segment results produced before human review, without reading final references. Beginning, middle, and end frames are combined into a composite image, which OSCaR receives with the target object to describe its state change.

Qwen then receives the OSCaR description, automatic segment description, automatic speech recognition (ASR) transcript, and object/time information to extract or infer states, actions, tools, causes, and cause types. Its prompt requests `unknown` when a cause is unspecified. This includes semantic inference, not just format conversion. Results therefore characterize OSCaR + Qwen as a complete adapted system. Common output formatting does not establish identical input conditions across baselines.

D.3 Ablation Scope

The variants in Section 5.4 remove functional stages rather than retrain the backbone. Removing State Graph also removes state completion performed within graph construction. Removing Structured Context changes the context supplied to cause reasoning, while removing Candidate Reasoning removes alternative-cause generation. Removing Closed-loop Verification removes consistency checks and prediction revision. These comparisons use the same backbone and evaluation protocol but do not establish equal inference-call or token budgets.