

CERA: Causal Event Reasoning and Attribution for Real-Time Surveillance

Anonymous Authors

Anonymous Institution
anonymous@eccv2026.org

Abstract. Real-time surveillance systems need more than anomaly flags or descriptive captions. Operational response requires causal accounts that connect events, agents, and outcomes under strict latency constraints. We introduce **CERA** (Causal Event Reasoning and Attribution), a framework direction for online surveillance analysis that prioritizes three requirements: causal fidelity, evidential grounding, and runtime efficiency. This draft establishes the problem framing and design requirements as the foundation for method and evaluation development.

Keywords: Causal Event Reasoning, Attribution, Surveillance, Efficient Inference

1 Introduction

Why Causality Matters in Surveillance. Modern surveillance models can detect anomalies or generate captions, but practical response requires more: a causal account of what happened, what triggered it, and which entities were responsible. In safety-critical settings, this distinction affects intervention priority, accountability, and prevention planning.

Problem Setting. We consider online surveillance streams where decisions must be both timely and explainable. For each event segment, the system should produce (1) an event statement, (2) an attribution statement linking causes to outcomes, and (3) supporting evidence references. We refer to this objective as *causal event reasoning and attribution*.

Current Gap. Existing pipelines are often optimized for a single objective. Detection-focused systems emphasize recall but provide limited causal structure. Caption-focused systems can describe scenes fluently but often blur causality and agency. Offline reasoning approaches are expressive but difficult to deploy with strict real-time constraints.

CERA. We introduce **CERA**, a framework direction for real-time causal event reasoning and attribution. CERA is organized around three constraints: *causal fidelity* (correct cause-effect structure), *evidential grounding* (traceable support in observed frames), and *runtime efficiency* (practical throughput for continuous streams). This paper starts by fixing these requirements as first-class design targets.

Scope of This Draft. This manuscript is built from a clean-room start. The current version focuses on defining the CERA problem statement and system requirements before expanding to full method specification and large-scale experiments.

Contributions.

- We define a real-time surveillance task for causal event reasoning and attribution with explicit outputs for event, cause, and evidence.
- We propose CERA as a framework direction centered on causal fidelity, evidential grounding, and runtime efficiency.
- We establish a starting evaluation perspective that jointly considers causal quality and deployment-time performance.

2 Method

2.1 Task Formulation

Let a surveillance stream be $\mathcal{V} = \{\mathbf{X}_1, \mathbf{X}_2, \dots\}$, where \mathbf{X}_t is the frame at time t . The system identifies event windows $\mathcal{W} = \{[t_s^k, t_e^k]\}_{k=1}^K$ and outputs one structured report per window.

For each window k , CERA predicts

$$\hat{\mathbf{y}}_k = \{\hat{e}_k, \hat{\mathcal{A}}_k, \hat{\mathcal{E}}_k\}, \quad (1)$$

where \hat{e}_k is the event statement, $\hat{\mathcal{A}}_k$ is a set of causal attribution tuples, and $\hat{\mathcal{E}}_k$ is an evidence index set.

Each attribution tuple is represented as

$$\hat{a}_{k,j} = (\hat{c}_{k,j}, \hat{r}_{k,j}, \hat{o}_{k,j}), \quad (2)$$

where $\hat{c}_{k,j}$ is a candidate cause, $\hat{o}_{k,j}$ is an outcome, and $\hat{r}_{k,j}$ is the causal relation type. The evidence set $\hat{\mathcal{E}}_k$ maps each tuple to supporting temporal spans or frame references.

2.2 CERA Design Requirements

CERA is designed around three joint requirements.

Causal Fidelity. Predictions should preserve cause-effect direction and avoid descriptive but non-causal statements. Operationally, this means the attribution tuples should match reference causal structure, not only lexical overlap.

Evidential Grounding. Every attribution claim should be linked to observable evidence in the same event window. Ungrounded claims are treated as low-trust outputs regardless of language fluency.

Runtime Efficiency. Inference must satisfy deployment latency constraints under continuous streams. Given an end-to-end latency L_{e2e} and deployment budget L_{max} , CERA should operate in the feasible region:

$$L_{e2e} \leq L_{max}. \quad (3)$$

2.3 Structured Objective

We model CERA as maximizing a utility that balances causal quality, evidence quality, and runtime feasibility:

$$\mathcal{U} = \lambda_c S_c + \lambda_e S_e + \lambda_r S_r, \quad (4)$$

where S_c measures causal fidelity, S_e measures evidence grounding quality, and S_r measures runtime fitness.

For runtime fitness, we use a normalized score:

$$S_r = \min \left(1, \frac{L_{max}}{L_{e2e}} \right). \quad (5)$$

This formulation encourages quality gains only when latency remains practical.

2.4 Online Inference Contract

Given an event window $\mathbf{X}_{t_s^k:t_e^k}$, CERA predicts:

$$\hat{\mathbf{y}}_k = \mathcal{F}_\theta(\mathbf{X}_{t_s^k:t_e^k}), \quad (6)$$

with confidence score p_k . To reduce high-confidence hallucination risk in safety contexts, CERA uses abstention:

$$\hat{\mathbf{y}}_k = \begin{cases} \mathcal{F}_\theta(\mathbf{X}_{t_s^k:t_e^k}) & \text{if } p_k \geq \tau_{conf}, \\ \emptyset & \text{otherwise.} \end{cases} \quad (7)$$

This contract makes failure modes explicit and prevents forcing a causal narrative when evidence is weak.

2.5 Output Schema and Validation

For deployment integration, CERA exposes a fixed output schema:

$$\hat{\mathbf{y}}_k = \{\hat{e}_k, \hat{\mathcal{A}}_k, \hat{\mathcal{E}}_k, \hat{s}_k\}, \quad (8)$$

where \hat{s}_k is a quality status flag (`valid`, `abstain`, or `insufficient_evidence`).

Before returning `valid`, CERA applies two consistency checks:

$$\text{CausalCheck}(\hat{\mathcal{A}}_k) = 1, \quad \text{EvidenceCheck}(\hat{\mathcal{A}}_k, \hat{\mathcal{E}}_k) = 1. \quad (9)$$

The first check verifies direction-consistent tuples and relation admissibility. The second check verifies whether each accepted tuple has linked observable support. If either check fails, CERA downgrades output status to `insufficient_evidence` and suppresses final attribution claims.

2.6 Error Taxonomy

To keep failure analysis actionable, we partition output errors into three types:

- **Type-I (Structure Error)**: cause-effect direction or relation type is wrong.
- **Type-II (Grounding Error)**: claim is plausible but unsupported by linked evidence.
- **Type-III (Latency Error)**: quality is acceptable but latency violates deployment budget.

This taxonomy maps directly to the three CERA requirements and guides where to improve: reasoning logic, evidence alignment, or system optimization.

2.7 Current Scope

This version defines CERA at the task and objective level. Next iterations will specify the concrete module design (event proposal, attribution construction, evidence alignment, and budget-aware control) and their implementation details.

3 Experiments

3.1 Evaluation Targets

We evaluate CERA along three axes aligned with the method objective:

- **Causal Quality**: correctness of predicted attribution tuples.
- **Evidence Quality**: correctness and completeness of linked evidence references.
- **Runtime**: end-to-end latency and effective throughput under fixed hardware.

3.2 Planned Metrics

Let $\hat{\mathcal{A}}$ and \mathcal{A}^* be predicted and reference attribution tuples, and let $\hat{\mathcal{E}}$ and \mathcal{E}^* be predicted and reference evidence. We plan to report:

- tuple-level precision/recall/F1 for causal attribution,
- evidence precision/recall/F1 for grounding quality,
- end-to-end latency (ms), FPS, and stream capacity.

3.3 Protocol Principles

To keep comparisons meaningful, all compared systems will use the same runtime protocol: hardware, input resolution, decoding length, and reporting policy. Final claims will be tied to both quality metrics and latency feasibility under deployment budgets.

4 Conclusion

TODO. Summarize findings and next validation steps.

References