

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** (**C**ausal **E**vent **R**easoning and **A**ttribution), a framework direction for online surveillance analysis that prioritizes three requirements: causal fidelity, evidential grounding, and runtime efficiency. This manuscript presents an initial formalization with a structured objective, an online output contract, and an evaluation protocol that jointly considers reasoning quality and deployment-time feasibility.

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 [5,9]. Detection-focused systems emphasize recall but provide limited causal structure [10,1,12,2,13]. Caption-focused systems can describe scenes fluently but often blur causality and agency [4,3,7]. Offline reasoning approaches are expressive but difficult to deploy with strict real-time constraints [6,7].

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 includes a first method-level formalization and protocol-level experiment design. Next iterations will expand concrete module implementations and larger-scale empirical validation.

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 Related Work

Surveillance Anomaly Detection. Prior surveillance literature has made strong progress on anomaly detection under weak supervision, including benchmark construction and learning objectives that improve anomaly recall [5,8,11,10,1]. More recent adaptations of vision-language models improve semantic coverage for anomaly understanding [12,2,13]. However, most systems still optimize anomaly scoring or descriptive diagnosis, not explicit cause-outcome attribution with verifiable evidence links.

Video-Language Reasoning. General video-language systems provide richer temporal descriptions and dialogue-style interaction [4,3,6,7]. These advances are valuable for open-ended understanding, but they are not primarily designed around causal direction consistency, evidence attachment, or deployment-time abstention behavior required in online surveillance settings.

Position of CERA. CERA is not another captioning or anomaly-score variant. It targets the missing intersection between surveillance deployment constraints and causal attribution requirements by enforcing structured outputs, evidence checks, and runtime-feasibility objectives in one contract.

3 Method

3.1 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.

3.2 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:

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

3.3 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.

3.4 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.

3.5 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.

3.6 Failure Taxonomy

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

- **Causal Structure Violation**: cause-effect direction or relation type is wrong.
- **Evidence Support Failure**: claim is plausible but unsupported by linked evidence.
- **Runtime Budget Breach**: 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.

3.7 Module-Level Design

We decompose CERA into four cooperative modules:

$$\hat{\mathbf{y}}_k = \mathcal{M}_{ctrl}(\mathcal{M}_{evd}(\mathcal{M}_{attr}(\mathcal{M}_{evt}(\mathbf{X}_{t_s^k:t_e^k}))))). \quad (10)$$

This decomposition keeps responsibilities explicit and aligns each module with one or more CERA requirements.

Event Proposal Module (\mathcal{M}_{evt}). The event proposal stage generates candidate temporal windows and tracked entities from the stream. Its goal is high recall under bounded latency, producing a compact event context for downstream reasoning instead of processing the full stream at maximum cost.

Attribution Construction Module (\mathcal{M}_{attr}). Given event context, CERA builds a directed causal interaction graph

$$\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k^c), \quad (11)$$

where nodes represent entities/outcomes and directed edges represent candidate causal relations. Final attribution tuples are decoded from validated graph edges, enforcing direction consistency by construction.

Evidence Alignment Module (\mathcal{M}_{evd}). For each candidate tuple, CERA links supporting evidence spans in the same event window. Each claim receives an evidence support score $q_{k,j}$ and is retained only if $q_{k,j} \geq \tau_{evd}$. This stage prevents fluent but ungrounded attributions from passing as valid outputs.

Budget-Aware Control Module (\mathcal{M}_{ctrl}). The control module manages compute allocation and output policy under runtime budgets. It applies adaptive depth, selective decoding, and abstention triggers to keep latency within L_{max} while preserving causal and evidential quality. When budget pressure is high, the controller prioritizes reliable partial outputs over unstable full attributions.

3.8 Current Scope

This manuscript now defines CERA at task, objective, and module-design levels. The next stage is implementation-level specification and empirical validation of each module under shared deployment constraints.

4 Experiments

4.1 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.

As an initial benchmark substrate, we plan to start from established surveillance anomaly datasets [8,11] and extend evaluation with attribution- and evidence-level annotations.

4.2 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.

4.3 Protocol Principles

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

5 Conclusion

This paper established **CERA** as a practical objective for real-time surveillance understanding: produce causal event reports that are structurally correct, evidentially grounded, and operationally feasible. We formalized this objective through (1) a structured task definition, (2) a utility that balances causal quality, grounding quality, and runtime fitness, (3) an online contract with abstention and explicit status outputs, and (4) a concrete module-level design for event proposal, attribution construction, evidence alignment, and budget-aware control. We also defined an initial evaluation protocol aligned with these requirements.

The next validation stage is empirical. We plan to implement and ablate the CERA modules, build annotated benchmarks for attribution and evidence linkage, and run controlled comparisons against detection- and caption-centric baselines under matched runtime settings. This progression is intended to test whether causal reasoning quality can improve without violating deployment latency budgets.

References

1. Feng, J.C., Hong, F.T., Zheng, W.S.: MIST: Multiple Instance Self-Training Framework for Video Anomaly Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) [1](#), [2](#)
2. Gu, Z., Zhu, B., Zhu, G., Chen, Y., Tang, M., Wang, J.: AnomalyGPT: Detecting Industrial Anomalies using Large Vision-Language Models. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2024) [1](#), [2](#)
3. Lin, B., Zhu, B., Ye, Y., Ning, M., Jin, P., Yuan, L.: Video-LLaVA: Learning United Visual Representation by Alignment Before Projection. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024) [1](#), [2](#)

4. Maaz, M., Rasheed, H., Khan, S., Khan, F.S.: Video-ChatGPT: Towards Detailed Video Understanding via Large Vision and Language Models. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL) (2024) [1](#), [2](#)
5. Nayak, R., Pati, U.C., Das, S.K.: A Survey on Deep Learning Based Video Anomaly Detection. *IEEE Transactions on Circuits and Systems for Video Technology* (2021) [1](#), [2](#)
6. Ren, S., Yao, L., Li, S., Sun, X., Hou, L.: TimeChat: A Time-sensitive Multi-modal Large Language Model for Long Video Understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024) [1](#), [2](#)
7. Song, E., Chai, W., Wang, G., Zhang, Y., Zhou, H., Wu, F., Guo, X., Ye, T., Lu, Y., Hwang, J.N., Gao, G.: MovieChat: From Dense Token to Sparse Memory for Long Video Understanding. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024) [1](#), [2](#)
8. Sultani, W., Chen, C., Shah, M.: Real-world Anomaly Detection in Surveillance Videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) [2](#), [5](#)
9. Tian, Y., Zhang, X., Werghi, N., Abdullah, A., et al.: A Survey of Video Analytics for Smart Cities. *IEEE Access* (2020) [1](#)
10. Tian, Y., Pang, G., Chen, Y., Singh, R., Verjans, J.W., Carneiro, G.: Weakly-supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) [1](#), [2](#)
11. Wu, P., Liu, J., Shi, Y., Sun, Y., Shao, F., Wu, Z., Yang, Z.: Not Only Look, But Also Listen: Learning Multimodal Violence Detection Under Weak Supervision. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020) [2](#), [5](#)
12. Wu, P., Zhou, X., Pang, G., Sun, Y., Liu, J., Wang, P., Zhang, Y.: VADCLIP: Adapting Vision-Language Models for Weakly Supervised Video Anomaly Detection. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2024) [1](#), [2](#)
13. Zhang, H., Xu, X., Wang, X., Zeng, J., Li, C., Chen, X.: Holmes-VAD: Towards Unbiased and Explainable Video Anomaly Detection via Multi-modal LLM. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025) [1](#), [2](#)