HoloTrace: LLM-based Bidirectional Causal Knowledge Graph for Edge-Cloud Video Anomaly Detection

Hanling Wang
wanghl03@pcl.ac.cn
Pengcheng Laboratory
Shenzhen, Guangdong, China
Shenzhen International Graduate School, Tsinghua University
Shenzhen, Guangdong, China

Qing Li*
liq@pcl.ac.cn
Pengcheng Laboratory
Shenzhen, Guangdong, China

Li Chen
20226670@stu.neu.edu.cn
Northeastern University
Shenyang, Liaoning, China

Haidong Kang
hdkang@stumail.neu.edu.cn
Northeastern University
Shenyang, Liaoning, China

Fei Ma
mafei@gml.ac.cn
Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)
Shenzhen, Guangdong, China

Yong Jiang*
jiangy@sz.tsinghua.edu.cn
Shenzhen International Graduate School, Tsinghua University
Shenzhen, Guangdong, China
Pengcheng Laboratory
Shenzhen, Guangdong, China

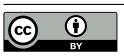
Abstract

Video anomaly detection (VAD) is vital for public safety, yet current approaches struggle with limited generalization, low interpretability, and high resource demands. To address these challenges, we propose HoloTrace, an edge-cloud collaborative VAD system that integrates large language models (LLMs) to construct and update a novel bidirectional causal knowledge graph. At the edge, HoloTrace leverages LLM-based cross-modal understanding and employs Hidden Markov Model (HMM) for bidirectional event reasoning, obtaining anomaly boundaries with low computational overhead. On the cloud side, LLMs are leveraged to dynamically update the Bi-CKG graph with key frames sent from the edge, in order to update causal relationships between events. Additionally, we introduce SVAD, a new large-scale VAD dataset comprising 632 real-world surveillance videos across 10 anomaly types and diverse scenes, with manually labeled frame-level annotations. Experimental results demonstrate that HoloTrace not only achieves the highest accuracy but also enhances interpretability and efficiency, paving the way for more generalizable and explainable video anomaly detection systems.

CCS Concepts

• **Computing methodologies** → **Scene anomaly detection**; **Knowledge representation and reasoning**; **Machine learning**.

*Corresponding author.



MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755185

Keywords

Anomaly detection, Knowledge graph, Large language model

ACM Reference Format:
Hanling Wang, Qing Li, Li Chen, Haidong Kang, Fei Ma, and Yong Jiang. 2025. HoloTrace: LLM-based Bidirectional Causal Knowledge Graph for Edge-Cloud Video Anomaly Detection. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3755185

1 Introduction

Video Anomaly Detection (**VAD**) aims to identify anomaly events in videos and plays a crucial role in applications such as traffic monitoring [39] and public safety [30, 58]. With the growing number of surveillance cameras and increasing video resolutions, traditional centralized VAD approaches face challenges related to limited bandwidth and computational resources. Leveraging heterogeneous edge-cloud resources for low-latency, high-accuracy VAD has become a key focus in industry and academia.

On one hand, the sheer volume of video data makes uploading all data to the cloud for analysis impractical due to bandwidth limitations and high end-to-end latency, which hinders timely anomaly detection. On the other hand, the limited computational and storage capabilities of edge devices prevent them from handling compute-intensive analytics tasks at the data source.

From a methodological perspective, traditional VAD approaches can be categorized into three types based on the availability of annotated data: unsupervised [10, 11, 32, 34], fully supervised [18, 57], and weakly supervised [17, 19, 50]. Unsupervised VAD typically learns only from normal samples, while fully and weakly supervised VAD are trained on videos containing anomaly events with frame-level or video-level labels. Recently, large language models (**LLMs**) have been integrated into VAD systems to enable training-free and interpretable analysis due to their strong capabilities in logical reasoning and zero-shot learning [21, 41, 57].

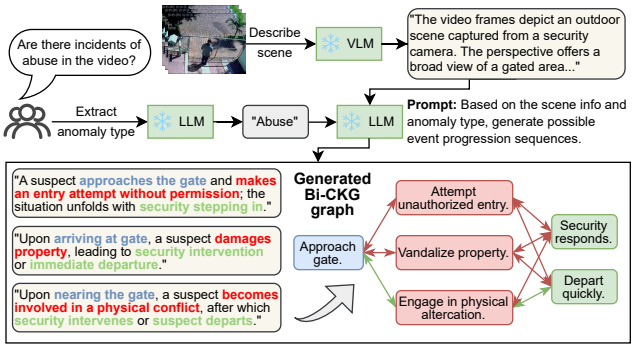


Figure 1: Construction process of a Bi-CKG graph.

Despite the promising performance of the aforementioned systems and methods, several limitations remain: (*i*) *Zero-shot generalization*: Previous approaches can only detect anomalies based on prior knowledge of known anomaly types, making them incapable of identifying unseen anomalies; (*ii*) *Interpretability*: Most existing methods focus solely on prediction accuracy, treating the decision-making process as a black box and lacking the ability to explain the underlying causes of anomalies; (*iii*) *Inference cost*: Prior works prioritize detection accuracy while overlooking critical deployment factors such as latency, computational overhead, and bandwidth consumption; (*iv*) *Lack of high-quality datasets*: Existing VAD-related datasets either contain only video-level annotations or of low scene diversity, limiting the development of accurate VAD models with fine-grained results.

In recent years, Causal Knowledge Graph (**CKG**) has emerged as a model that integrates and stores causal relationships within data along with related knowledge [9]. CKGs exhibit strong causal inference capabilities, not only revealing causal relationships between events but also aiding in the reasoning of anomaly events. Traditional CKGs [23, 35, 54] are typically constructed offline using predefined datasets and contain only unidirectional associations between events. However, we observe that causal inference often involves bidirectional logical relationships. For example, an explosion as a preceding event can lead to thick smoke as a subsequent event, while the presence of thick smoke can also be used to infer the occurrence of an explosion. Moreover, traditional CKGs rely on large-scale real-world datasets, which hinders their rapid construction and dynamic adaptation.

To address these challenges, we propose **HoloTrace**, an edge-cloud collaborative VAD system that integrates an LLM-based bidirectional causal knowledge graph. The core idea of HoloTrace is to assign dedicated nodes to key events in the video and establish bidirectional occurrence probabilities between them, forming a Bidirectional Causal Knowledge Graph (**Bi-CKG**). The construction process of a Bi-CKG graph is shown in Fig. 1.

Specifically, on the edge side, we first utilize the inertial characteristics of the anomaly score from a LLM to determine the preliminary range of anomaly events. Then, we employ a Hidden Markov Model (**HMM**) to infer the event state of each frame based on the constructed Bi-CKG graph. Using forward reasoning, we determine event states in the chronological direction, while backward reasoning infers states in the reverse direction. By integrating the reasoning results from both directions, we derive the final event state for each frame.

On the cloud side, we leverage the internal knowledge of LLMs to construct and update the Bi-CKG graph. During the offline phase, the LLM generates related event chains based on the video scene and initialize a graph for each anomaly type. To establish bidirectional association probabilities between events, we exploit the internal knowledge of LLMs with a carefully designed choice-based prompt. In the online phase, the LLM generates a new graph based on key video frames from the edge and update the original graph, enhancing its representation of scenarios in the video.

In addition, we propose **SVAD**, a new large-scale VAD dataset that includes 10 anomaly types and 632 real-world videos. The videos are collected from public resources and are manually annotated with frame-level anomaly labels. Compared to existing VAD datasets, SVAD provides more fine-grained annotations, greater scene diversity, and is based on real-world scenarios. Our contributions are:

- HoloTrace is the first system to leverage LLMs to construct and update the Bi-CKG graph for the VAD task. By analyzing events in the video and utilizing logical reasoning capabilities, we enhance the accuracy of the VAD task.
- We construct SVAD, a new large-scale VAD dataset that includes 632 real-world videos and frame-level annotations.
- We propose event reasoning based on inertial characteristics of anomaly event and HMM on the edge to comprehensively determine the event state of each frame and provide explainable anomaly detection results.
- We leverage the internal knowledge of LLMs to construct the Bi-CKG graph using choice-based prompt design, and update it with key events from real-world video frames for better event understanding and reasoning.

2 Related Works

2.1 Video Anomaly Detection

Due to the sparsity of anomaly events and the limited availability of training data, VAD methods are generally classified into three types: unsupervised, fully-supervised, and weakly-supervised. Unsupervised VAD is trained solely on normal videos to learn normal patterns. In this category, reconstruction-based methods [6, 43, 53] use generative models, *e.g.*, autoencoders and diffusion models, to reconstruct the original data and compute anomaly scores based on the discrepancy between the original and reconstructed samples. Prediction-based methods [5, 24, 49] predict future frames using previous frames and determine anomaly scores based on the difference between the predicted and actual frames. Some studies [26, 42] also combine reconstruction and prediction approaches to enhance model performance.

In contrast, fully-supervised and weakly-supervised VAD are trained on both normal and anomaly videos. However, fully-supervised VAD uses frame-level annotations, while weakly-supervised VAD relies on video-level annotations. Since most existing large-scale VAD datasets only contain video-level annotations, research on fully-supervised VAD [18, 57] is relatively limited. Most weakly-supervised VAD methods use Convolutional

Neural Networks (**CNNs**) and adopt the Multiple Instance Learning (**MIL**) framework [40] for training. MIL improves anomaly detection accuracy by dividing videos into multiple segments and classifying them based on video-level labels. Recently, researchers have also explored integrating attention mechanisms [29, 44] and Graph Neural Networks (**GNNs**) [3] to enhance the detection accuracy and robustness of weakly-supervised VAD.

2.2 Casual Knowledge Graph

Knowledge graphs aim to leverage logical associations between entities to address problems and have demonstrated effectiveness in tasks such as question answering [15] and information extraction [36]. Traditional knowledge graphs integrate entities and their relationships within a structured graph. Causal knowledge graphs extend this concept by incorporating probabilistic information into the logical rules governing entity interactions. Nesen *et al.* [31] represent objects and relationships within a knowledge graph and compute semantic similarity among video objects to detect anomalies with significant semantic deviations. Chen *et al.* [4] decouple scenes and actions in videos, explicitly modeling their relationships using a knowledge graph to enable human behavior anomaly detection.

2.3 Large Language Model

Pretrained LLMs have demonstrated exceptional performance in natural language processing and have excelled in downstream tasks such as logical reasoning [7] and text comprehension [51]. Building on LLMs, vision-language models (**VLMs**) [25, 45, 48] showcase strong visual understanding capabilities. Recently, LLMs have been explored for intelligent knowledge graph generation and the VAD task. Yun *et al.* [55] leverage LLMs to automatically generate task-specific knowledge graphs and integrate them with multi-modal GNNs for VAD. Yang *et al.* [52] employ LLMs for rule-based reasoning, using inductive rules derived from a small set of normal samples to identify anomalies through a combination of induction and deduction. Zanella *et al.* [56] integrate LLMs and VLMs, utilizing generated textual descriptions of video frames and cross-modal similarity for anomaly scoring.

3 SVAD Dataset

In this section, we introduce **SVAD**, a new large-scale VAD dataset with frame-level annotations. The SVAD dataset contains 632 real-world surveillance videos and covers 10 types of anomaly behaviors, including *abuse*, *arson*, *burglary*, *car accident*, *fighting*, *jumping the red light*, *robbery*, *shooting*, *slip*, and *vandalism*. These anomaly types were selected because they frequently occur in real life and have a significant impact on public safety.

3.1 Video Collection

The videos in SVAD are collected from three public resources. We first selected high-quality surveillance videos from the UCF-Crime dataset [40] and trimmed them to ensure that the content of each video is continuous. UCF-Crime is a large-scale video dataset for anomaly detection, but it only provides video-level labels. Next, we searched for each type of anomaly video on Google [13] and Douyin [8] using corresponding keywords (*e.g.,* "surveillance video abuse", "surveillance video fight"). After manually filtering out videos that did not meet the requirements or were of low quality, we collected a total of 632 videos, amounting to over 9 hours of footage. Among them, 192 videos are from UCF-Crime, 394 videos are from Google search, and 46 videos are from Douyin.

3.2 Frame-level Annotation

Unlike existing large-scale VAD datasets, which only contain video-level annotations, we performed frame-by-frame annotation on all the collected videos. The annotation process consists of three steps: (*i*) Step 1: We manually label the boundaries of anomaly events in each video as a rough estimation. Note that each anomaly video contains only one type of anomaly; (*ii*) Step 2: We selected multiple frames (≥20) near the manually determined boundaries and used GPT-4o [16] to assess whether each frame was abnormal; (*iii*) Step 3: We manually review frames that exhibit inconsistencies between the VLM's assessment and the initial manual judgment, producing the final annotated dataset. We divide the SVAD dataset into a training set and a test set. The training set consists of 69 normal videos and 444 anomaly videos, while the test set contains 18 normal videos and 101 anomaly videos.

3.3 Comparison with Existing Datasets

To highlight the advantages of SVAD, we compare it with other widely used VAD datasets containing frame-level annotations in Table 1. Overall, SVAD offers three key advantages: (*i*) It features diverse, real-world video scenes, enabling the development of more robust and reliable VAD approaches; (*ii*) It consists of full videos, allowing models to leverage temporal context for more accurate anomaly detection; (*iii*) Among datasets with frame-level annotations, SVAD is the largest with the highest number of frames.

Table 1: Comparison of SVAD with other VAD datasets.

Dataset	Real World	Scene Diversity	Type	#Frames	#Videos
UCSD PED1 [47]	✓	low	Image	14,000	70
UCSD PED2 [47]	✓	low	Image	4,560	28
CUHK Avenue [27]	✓	low	Video	30,652	37
Subway Entrance [2]	✓	low	Video	136,524	1
Subway Exit [2]	✓	low	Video	72,401	1
UBnormal [1]	✗	high	Video	236,902	543
ShanghaiTech [28]	✓	middle	Video	317,398	437
SVAD (Ours)	✓	**high**	**Video**	**1,391,566**	**632**

4 HoloTrace: System Design

4.1 Overview

The system overview of HoloTrace is depicted in Fig. 2, which consists of causal event inference and adaptive frame transmission at the edge, as well as dynamic Bi-CKG graph construction and update in the cloud.

Edge Side. Given a video frame, we first compute the cosine similarity between the frame feature and the anomaly-related keyword feature. Based on the computed similarity, the *Inertial Event*

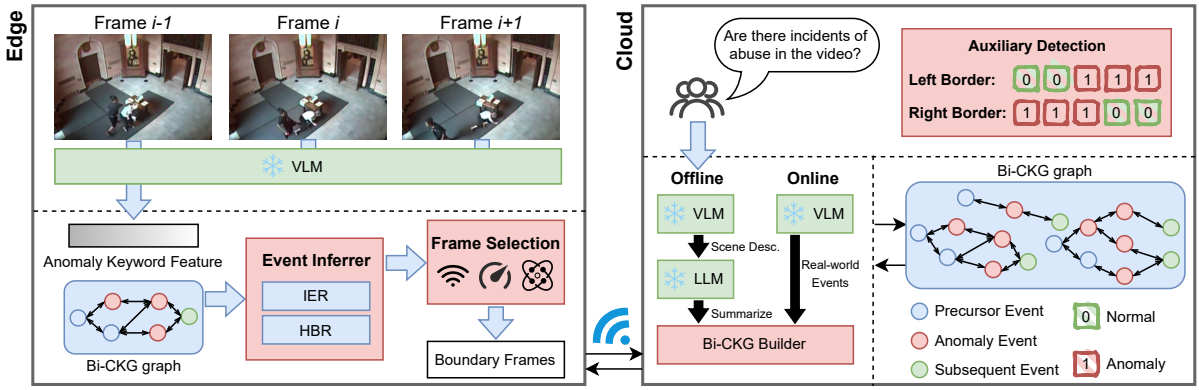


Figure 2: The system overview of HoloTrace.

Reasoning (**IER**) module generates a preliminary boundary for the anomaly event. Subsequently, we calculate the cosine similarity between the frame feature and the features of all events in the constructed Bi-CKG graph at the edge. Based on the causal relationships in the Bi-CKG graph, the *HMM Bidirectional Reasoning* (**HBR**) module refines the preliminary boundaries. After determining the range of the anomaly event at the edge, multiple frames around the boundary of the anomaly event are selected for transmission to the cloud. The frame selection process considers the event variety of the video, bandwidth consumption, and cloud workload. The transmitted frames are used to update the Bi-CKG graph on the cloud and improve the anomaly detection results.

Cloud Side. After receiving the selected frames on the cloud, a VLM is used to determine whether each frame is abnormal or not. The anomaly detection results are then analyzed to identify anomaly boundaries (*i.e.,* a left or right border). If a boundary is missing in the received frames, the edge device is requested to send additional frames until the boundary is found. Otherwise, the descriptions of all frames are summarized into an event description and used by the *Bi-CKG Builder* for online graph updating. The *Bi-CKG Builder* operates in an offline mode and an online mode.

In offline mode, the LLM first extracts anomaly-related keywords from a user query, and the VLM generates scene descriptions based on randomly sampled frames from the edge. Then, based on the anomaly-related keywords and scene descriptions, the LLM generates multiple anomaly event chains. By leveraging the reasoning capabilities of the LLM and choice-based prompts, a knowledge graph with bidirectional probabilities for the anomaly event is constructed. In online mode, the LLM extracts event chains from the summarized event descriptions and updates the existing Bi-CKG graph. By incorporating anomaly events from real-world scenarios, the graph is dynamically updated.

4.2 Dynamic Bi-CKG Building

4.2.1 Bi-CKG Graph Definition. Before introducing the details of Bi-CKG graph, we define the concepts and notations that will be used throughout the paper. Formally, the Bi-CKG graph is defined as a symmetric directed graph G, composed of a set of connected nodes, and each graph corresponds to a specific anomaly type.

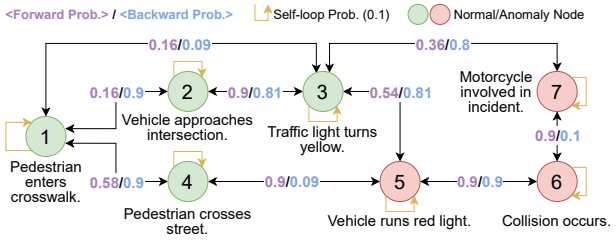


Figure 3: An example Bi-CKG graph G.

Specifically, $G = \{V, R\}$, where V represents the set of nodes, and R is the set of edges. An example Bi-CKG graph is shown in Fig. 3.

In HoloTrace, each node $V_i \in V$ corresponds to a specific event (*e.g.,* node 4 represents the event "Pedestrian crosses street"). Based on its relevance to the anomaly type, a node can be categorized as either a normal node or an anomaly node. For two events V_i and V_j ($i < j$) occurring in chronological order (*i.e.,* V_i happens before V_j), the edge R_{ij} denotes the probability that V_i leads to V_j, referred to as the forward probability. Conversely, the edge R_{ji} represents the probability that V_i has occurred given that V_j is observed, known as the backward probability.

In the Bi-CKG graph, self-loops are allowed, meaning a node can connect to itself (*i.e.,* $i = j$). In this case, the forward and backward probabilities are identical, *i.e.,* the self-loop probability (default 0.1). This probability indicates the likelihood of an event persisting over time. During the construction and updating of the graph, we normalize both the forward and backward probabilities within G to ensure they satisfy Theorem 1 and Theorem 2.

Theorem 1 (Forward Probability Sum). *In a Bi-CKG graph G, for any node i, the sum of all forward probabilities satisfies:*

$$\sum_{j \geq i} R_{ij} = 1.$$

Theorem 2 (Backward Probability Sum). *In a Bi-CKG graph G, for any node j, the sum of all backward probabilities satisfies:*

$$\sum_{i \leq j} R_{ji} = 1.$$

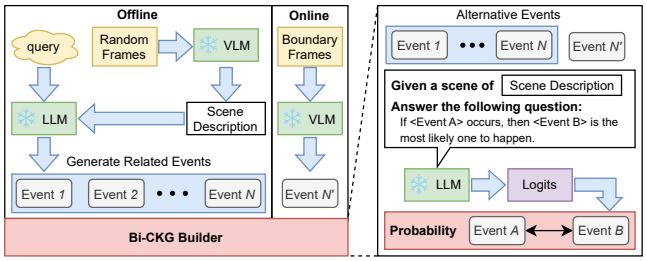


Figure 4: Overview of the Bi-CKG builder.

For ease of discussion, in the remainder of this section, we denote the causal association between two events in a graph as $\langle V_c, V_e, I_f, I_b \rangle$. Here, V_c represents the preceding event, V_e denotes the subsequent event, I_f indicates the forward probability, and I_b represents the backward probability. The preceding event occurs before the subsequent event in a given pair of events.

To construct and update the Bi-CKG graph, we designed the Bi-CKG Builder, an overview of which is shown in Fig. 4. It consists of two phases: an offline phase for graph construction and an online phase for graph updating. We describe each phase in detail below.

4.2.2 Offline Phase. The main goal of the offline phase is to initialize the Bi-CKG graph G using the user query and frames from the edge device. First, given a user query in natural language, an LLM extracts the anomaly-related keyword. For example, for the query "Is there a robbery in the video?", HoloTrace extracts the keyword "Robbery". Next, k ($k = 5$) recent frames are randomly selected to better understand the scene. Specifically, a VLM extracts scene descriptions from each frame, and the LLM then summarizes the descriptions of the k frames to generate an overall description of the video's scene.

Based on the anomaly-related keyword and the overall scene description, the LLM generates all possible anomaly event chains. For example, two generated anomaly event chains for "robbery" could be "Enter store → Threaten cashier → Demand money" and "Enter store → Browse items → Show weapon → Collect cash." These event chains are then used to construct the Bi-CKG graph, where nodes represent extracted events, and edges represent the association relationships between events.

However, the forward and backward probabilities for each edge remain unknown. To address this, we categorize the nodes into two types. For nodes with only one preceding event (*e.g.,* Node 4 in Fig. 3) or one subsequent event (*e.g.,* Node 6 in Fig. 3), Theorem 1 and Theorem 2 allow us to directly set the corresponding backward or forward probability to 0.9, considering that the self-loop probability is 0.1. For the remaining nodes, we design a choice-based prompt that leverages the reasoning capabilities of the LLM to generate the probabilities.

Specifically, we include the overall scene description in the prompt and ask the LLM to answer a question in the format: "If <event> occurs, which event from the following list is most likely to happen?" To determine the forward probability I_f, we replace "<event>" with the preceding event and label all possible subsequent events with numbers for the LLM to choose from. Once the LLM generates a response, we retrieve the logits for all subsequent events and normalize them to obtain the forward probability for each event. Similarly, to determine the backward probability I_b, we replace "<event>" with the subsequent event and label all possible preceding events with numbers for the LLM to choose from. After generating both probabilities, the complete Bi-CKG graph, including forward and backward probabilities, is finalized.

Additionally, for object-based anomalies (*e.g.,* a biker appearing on the pedestrian sidewalk in the UCSD PED2 dataset [47]), constructing an event chain with relevant events may be challenging. In such cases, the event chain can be simply defined as "no <anomaly_object> → <anomaly_object> occurs." All other operations remain the same.

4.2.3 Online Phase. To better align the knowledge in the Bi-CKG graph with the video content, we update the offline Bi-CKG graph using selected video frames from the edge. This section describes the graph updating process, while the frame selection process is clarified in Section §4.4.

Specifically, given a set of selected frames, the VLM first generates a description for each frame. Unlike the scene description in the offline phase, the description generated in the online phase focuses more on the events occurring within the frames. Then, an LLM is used to summarize the event descriptions of all frames and generate an event chain. Based on this event chain, we construct a local Bi-CKG graph G' following the steps outlined in the offline phase. Finally, we integrate this local graph G' into the initial Bi-CKG graph G to produce the updated Bi-CKG graph.

Denote the probability of the event pair $\langle V_c, V_e \rangle$ in the original graph G as $\langle I_f, I_b \rangle$, and the probability of the same event pair in G' as $\langle I'_f, I'_b \rangle$. The probability in G is then updated as follows:

$$\begin{aligned} I_f &\leftarrow \alpha I'_f + (1-\alpha) I_f, \\ I_b &\leftarrow \alpha I'_b + (1-\alpha) I_b, \end{aligned} \tag{1}$$

where α controls the updating ratio. In this step, existing event pairs are updated, and new event pairs are inserted into the original graph G. After the forward and backward probabilities are modified, they are also normalized to ensure that Theorem 1 and Theorem 2 are always satisfied.

4.3 Casual Event Inferring

Due to the limited network resources on the edge device, uploading each frame to the cloud for anomaly detection is impractical. To address this issue, we design an *Event Inferrer* to perform preliminary anomaly analysis on the edge. The Event Inferrer consists of two main components: *Inertial Event Reasoning* (IER) and *HMM Bidirectional Reasoning* (HBR). IER uses the similarity between anomaly keywords and video frames for preliminary anomaly identification, while HBR refines the anomaly detection results by leveraging the Bi-CKG graph.

4.3.1 Inertial Event Reasoning. Anomaly events in a video typically last for a certain period of time. For example, a *slip* may last for a few seconds, while a *fight* may last for several minutes. Generally, the beginning and ending frames of an anomaly event have lower anomaly scores compared to the frames in the middle. In practice, we find that the overall trend of the anomaly score for an anomaly event is to rise initially, remain stable, and then decrease,

which we refer to as the "inertia" of the anomaly event. In this section, we exploit the inertial characteristics of anomaly events for preliminary anomaly detection on the edge.

First, we use the text encoder from the pretrained CLIP [37] to extract features ξ_T for the anomaly-related keywords discussed in Section §4.2.2. Given a video frame, we then use the image encoder from CLIP to extract image features ξ_I. Based on the image feature of a video frame and the text feature of the anomaly-related keyword, we calculate the cosine similarity between them as the anomaly score S, *i.e.,*

$$S = \frac{\xi_T^\top \xi_I}{\|\xi_T\|_2 \cdot \|\xi_I\|_2}, \tag{2}$$

where $S \in [0, 1]$ and a higher S indicates that the frame is more likely to be part of an anomaly event.

To leverage the inertial characteristics of anomaly events, we apply Exponential Moving Average (**EMA**) smoothing to the anomaly score S and normalize it to obtain the smoothed anomaly score S'. Subsequently, we calculate the difference in anomaly scores between S'_n and S'_{n-1}, denoted as $\text{diff} \in [-1, 1]$. To determine the start and end of an anomaly event, we define an ascending window $w_a \in \mathbb{N}^+$, a holding window $w_h \in \mathbb{N}^+$, a descending window $w_d \in \mathbb{N}^+$, and an anomaly threshold $\text{thresh} \in \mathbb{R}$.

If, in a set of video frames, the condition $|\{i \mid \text{diff}_i > 0\}| > w_a$ and $|\{i \mid \text{diff}_i < 0\}| < w_d$ are satisfied, these frames are considered to be in the ascending phase. At the same time, the first ascending frame s is recorded. During the ascending phase, if for all $j \in \{i, i+1, \ldots, i+w_h-1\}$, $S'_j >$ thresh holds, then s is considered the starting frame of an anomaly event.

The procedure for determining the end position of an anomaly event is similar. If, in a set of video frames, the condition $|\{i \mid \text{diff}_i < 0\}| > w_d$ and $|\{i \mid \text{diff}_i > 0\}| < w_a$ are satisfied, these frames are considered to be in the descending phase. At the same time, the first descending frame e is recorded. During the descending phase, if for all $j \in \{i, i+1, \ldots, i+w_h-1\}$, $S'_j <$ thresh holds, then e is considered the ending frame of an anomaly event. Finally, we take $\langle s, e \rangle$ as the initial positions of the anomaly event.

4.3.2 HMM Bidirectional Reasoning. Based on the preliminary start and end positions $\langle s, e \rangle$ of the anomaly event obtained from the previous section, we further leverage the Bi-CKG graph to infer T frames near the boundaries to locate the accurate start and end positions of the anomaly event.

Specifically, the edge device maintains a Bi-CKG graph G from the cloud for the corresponding anomaly type. Then, we extract all the events $V = \{V_1, V_2, \ldots, V_n\}$. Next, a pretrained CLIP model is used to encode the text of each event. Using the same approach as in IER, we calculate the cosine similarity between a given frame and the text features of all events. After that, we obtain the normalized event scores $S_t = \{S_t^1, S_t^2, \ldots, S_t^n\}$, where S_t^n represents the similarity score between event V_n and the t-th video frame.

In Section §4.2.1, we defined the forward probability R_{ij} and backward probability R_{ji} ($i < j$). For the i-th node and the t-th frame, we define a state probability B_t^i, which represents the probability that the t-th frame belongs to the i-th event. Given a set of T frames, we leverage the causal relationships in the Bi-CKG graph to obtain the state probability B_t^i, where $t \in [1, T]$. Specifically, B_t^i is calculated via forward reasoning and backward reasoning.

In forward reasoning, we define the forward state probability α_t^i, which represents the probability that the t-th frame belongs to i-th event, based on the forward probability R_{ij}. The update process is

$$\alpha_{t+1}^j = [\sum_{i=1}^{j} \alpha_t^i R_{ij}] S_{t+1}^j, \quad j = 1, 2, \ldots, N \tag{3}$$

where $\alpha_1^i = B_1^i \cdot B_1^i$.

In backward reasoning, similarly, we define the backward state probability β_t^i, which represents the probability that the t-th frame belongs to the i-th event, based on the backward probability R_{ji}. The update process is

$$\beta_t^i = [\sum_{j=i}^{N} \beta_{t+1}^j R_{ji}] S_t^i, \quad j = 1, 2, \ldots, N \tag{4}$$

where $\beta_T^j = B_T^j \cdot B_T^j$.

By integrating the results from both forward and backward reasoning, the refined state probability B_t^i, which represents the probability that the t-th frame belongs to the i-th event, is defined as

$$B_t^i = \alpha_t^i \cdot \beta_t^i \tag{5}$$

For the t-th frame, we determine its associated event as the one with the largest state probability, *i.e.,* $\{i' \mid B_t^{i'} = \max_i B_t^i\}$. Based on the normal and anomaly nodes in the Bi-CKG graph G, we refine the preliminary anomaly boundaries from $\langle s, e \rangle$ to $\langle s', e' \rangle$. We refer to the refined detection result as *HoloTrace-base.*

4.4 Adaptive Frame Transmission

To update the Bi-CKG graph with real-world events in the video and further improve anomaly detection accuracy using cloud computing resources, we transmit frames of anomaly events from the edge to the cloud. Transmitting more frames to the cloud for analysis helps build a Bi-CKG graph that better fits real-world scenarios and improves anomaly detection accuracy. However, this also incurs higher bandwidth overhead and cloud resource consumption. To strike a balance between performance and resource consumption, we design an adaptive frame transmission mechanism.

4.4.1 Frame Selection on the Edge. For the boundary s' and e' of an anomaly event, we first select γ frames before and after the boundary as the initial selection range $\{0, 1, \ldots, 2\gamma\}$. We then refine this initial selection range to determine a start position r_1 and an end position r_2, where $0 \le r_1 < \gamma \le r_2 \le 2\gamma$. The frames between r_1 and r_2 are transmitted to the cloud for analysis. In this process, we send multiple frames around the beginning and ending of each anomaly event, as these frames are the most critical for Bi-CKG graph updating and anomaly event detection.

To improve the performance of anomaly detection, we define the event coverage of a series of frames $\{r_1, \ldots, r_2\}$ as C, which represents the variety of events in multiple frames. The larger the event coverage, the more fully an anomaly event is described. Specifically, the event coverage $C(r_1, r_2)$ is defined as:

$$C(r_1, r_2) = \max_{t_1 \in [r_1, r_2]} B_{t_1} - \min_{t_2 \in [r_1, r_2]} B_{t_2}, \tag{6}$$

where $B_t = \max_i B_t^i$. To avoid excessive consumption of network bandwidth and cloud resources, we define a bandwidth usage factor $W(r_1, r_2)$ and a cloud load factor $L(r_1, r_2)$ as:

$$W(r_1, r_2) = (1 + \frac{W_c}{W_m}) \cdot (r_2 - r_1 + 1),$$
$$L(r_1, r_2) = (1 + \frac{L_c}{L_m}) \cdot (r_2 - r_1 + 1). \tag{7}$$

where W_c denotes the current bandwidth usage, W_m denotes the maximum available bandwidth, L_c denotes the current cloud workload, and L_m denotes the maximum cloud workload.

Our goal is to improve event coverage while minimizing the consumption of bandwidth and cloud resources. In summary, the frame selection problem at the edge can be formulated as:

$$\max_{r_1, r_2} \{\lambda_1 C(r_1, r_2) - \lambda_2 W(r_1, r_2) - \lambda_3 L(r_1, r_2)\},$$
$$\text{s.t. Eq.(6)} - \text{Eq.(7)} \tag{8}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the weighting factors. By leveraging linear programming algorithms, we determine the frames $\{r_1, \ldots, r_2\}$ to be sent to the cloud for Bi-CKG graph updating.

4.4.2 Auxiliary Detection on the Cloud. Apart from being used for Bi-CKG graph updating, the sent frames are also utilized for auxiliary anomaly detection on the cloud. For each anomaly event, we define a *left border* and a *right border*. Specifically, if five consecutive frames follow the pattern "⟨normal, normal, anomaly, anomaly, anomaly⟩", these five frames are referred to as the left border of the anomaly event, with the third frame serving as the start of the anomaly event. Similarly, if five consecutive frames follow the pattern "⟨anomaly, anomaly, anomaly, normal, normal⟩", these five frames are referred to as the right border of the anomaly event, with the third frame determined as the end of the anomaly event.

Given a set of frames $\{r_1, \ldots, r_2\}$ sent from the edge, we use a VLM on the cloud to generate descriptions for each frame and then determine whether they are abnormal using an LLM. Based on the anomaly detection results on the cloud, we scan for a left border and a right border. If either the left or right border is missing from the sent frames, we further request the edge to send additional frames to locate them. Once the target frames are identified, the newly sent frames will be used for both Bi-CKG graph updating and auxiliary detection.

5 Evaluation

5.1 Experimental Setup

5.1.1 Dataset. We conducted comparative experiments on two datasets. **UCSD PED2**[20] is a frame-by-frame labeled dataset captured by fixed cameras on a sidewalk. It contains 28 video clips, with 16 clips in the training set and 12 clips in the test set. **SVAD** is a large-scale, frame-by-frame labeled surveillance video dataset constructed by us, containing 513 training videos and 119 test videos.

5.1.2 Evaluation Metric. To evaluate the anomaly detection performance of HoloTrace, we use the area under the frame-level ROC curve (*i.e.,* AUC) as the primary evaluation metric. Additionally, we also report the area under the frame-level precision-recall curve (*i.e.,* AP). Higher AUC and AP values indicate better performance.

5.1.3 Implementation. We use Cogvlm2 [14] as the VLM module and GPT-4o (version 2024-08-06) [16] as the LLM module. Image features are extracted using a frozen CLIP (ViT-L/14), and text features are extracted using a frozen CLIP text encoder. For the edge device, we use a Jetson Xavier NX (Ubuntu 18.04) [33] equipped with 8GB of RAM. For the cloud device, we use a workstation with seven TITAN RTX GPUs (Ubuntu 18.04).

5.2 Overall Performance

We compare HoloTrace with state-of-the-art methods, including unsupervised methods [38, 46] and training-free methods [12, 22, 37, 52, 56]. To ensure a fair comparison, we re-implement all LLM-based methods to use the same LLM configuration as HoloTrace. The results are shown in Table 2.

From the results, we observe that HoloTrace achieves superior performance compared to existing approaches. For example, on the UCSD PED2 dataset, HoloTrace outperforms SD-MAE (the best-performing unsupervised method) by 0.1% AUC and surpasses AnomalyRuler (the best training-free method) by 1.4% AUC. It is worth noting that SD-MAE is specifically trained on the corresponding dataset, whereas HoloTrace operates in a training-free manner, making SD-MAE inherently more advantageous. AnomalyRuler achieves a 0.4% higher AP than HoloTrace, primarily due to its use of stricter rule-based prompts for frame-level anomaly detection, enabling finer discrimination at the cost of higher computational complexity. On the SVAD dataset, HoloTrace achieves 81.4% AUC and 76.6% AP, outperforming all other baselines.

Table 2: Comparison of different VAD approaches.

Method	Backbone	UCSD PED2		SVAD	
		AUC (%)	AP (%)	AUC (%)	AP (%)
Wang et al. [46]	PWC-Net	94.2	N/A	N/A	N/A
SD-MAE[38]	CvT	95.4	N/A	N/A	N/A
ZS CLIP[37]	ViT	69.3	88.1	69.4	60.1
ZS IMAGEBIND[12]	ViT	76.2	90.6	65.7	56.4
LLaVA-v1.5[22]	ViT	65.6	87.0	60.8	53.0
AnomalyRuler[52]	ViT	93.9	**98.5**	N/A	N/A
HoloTrace-base (Ours)	ViT	93.2	97.5	78.0	70.5
HoloTrace (Ours)	ViT	**95.5**	98.1	**81.4**	**76.6**

5.3 Ablation study

5.3.1 Effectiveness of each proposed module. We evaluate different variants of our proposed HoloTrace to verify the effectiveness of the three modules: IER, HBR, and CAD. Table 3 shows the AUC and AP of all HoloTrace variants. When CAD is enabled (Row 2), HoloTrace shows improvements of 3.26% AUC and 1.17% AP on the UCSD PED2 dataset, and improvements of 14.81% AUC and 14.21% AP on the SVAD dataset. In Row 3, enabling HBR, *i.e.,* leveraging the association between events, leads to an improvement of 3.72% in AUC and 1.56% in AP on the UCSD PED2 dataset, and a substantial gain of 17.86% in AUC and 18.21% in AP on the SVAD dataset, compared to when HBR is disabled. Finally, enabling all three modules yields the best performance, achieving an AUC of 95.46% and AP of 98.07% on UCSD PED2, and an AUC of 81.36% and AP of 76.64% on SVAD.

Table 3: Ablation study of each module.

Module			UCSD PED2		SVAD	
IER	HBR	CAD	AUC (%)	AP (%)	AUC (%)	AP (%)
✓	✗	✗	89.44	95.91	60.16	52.28
✓	✗	✓	92.70	97.08	74.97	66.49
✓	✓	✗	93.16	97.47	78.02	70.49
✓	✓	✓	**95.46**	**98.07**	**81.36**	**76.64**

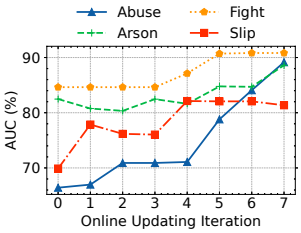


Figure 5: Impact of Bi-CKG graph updating.

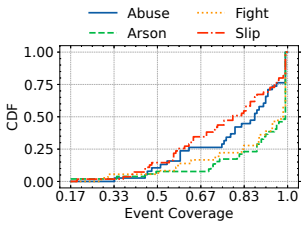


Figure 6: Distribution of event coverage in videos.

5.3.2 Effectiveness of Bidirectional Reasoning Mechanism. To investigate the impact of the bidirectional reasoning mechanism in Bi-CKG graph, we test HBR using only forward reasoning, only backward reasoning, and bidirectional reasoning on UCSD PED2 and SVAD. The results are shown in Table 4. It can be observed that enabling only forward reasoning decreases the AUC and AP on the UCSD PED2 dataset, while improving performance on the SVAD dataset. This is because UCSD PED2 primarily includes object-centric anomalies (e.g., bikers on pedestrian walkways), where context from preceding events contributes little to detection. In contrast, SVAD includes more action-centric anomalies (*e.g.,* slipping), which benefit significantly from temporal cues and features of prior events, making forward reasoning more effective. In comparison, enabling only backward reasoning improves the accuracy, as backward reasoning helps to identify anomalies by considering the events leading up to a particular frame. Finally, enabling both forward and backward reasoning for HBR achieves the best AUC and AP on both datasets.

Table 4: Ablation study of bidirectional reasoning.

Direction		UCSD PED2		SVAD	
Forward	Backward	AUC (%)	AP (%)	AUC (%)	AP (%)
✗	✗	89.44	95.91	60.16	52.28
✓	✗	86.42	94.56	61.86	53.88
✗	✓	91.70	96.93	66.88	60.94
✓	✓	**93.16**	**97.47**	**78.02**	**70.49**

5.3.3 Effectiveness of Online Bi-CKG Graph Updating. Fig. 5 shows the AUC under different iterations of online Bi-CKG graph updating for four types of anomaly videos. It can be observed that the AUC increases with more iterations of Bi-CKG graph updating. This indicates that online updating of the graph using real-world frames from the edge is critical for performance improvement.

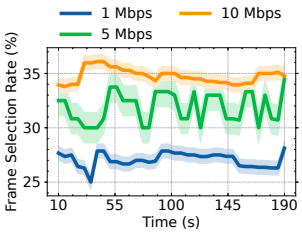


Figure 7: Percentage of frame selection rate.

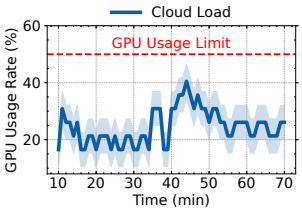


Figure 8: Percentage of GPU usage on cloud device.

5.3.4 Effectiveness of Frame Selector. We first demonstrate the rationale behind frame selection based on event coverage. Fig. 6 shows the cumulative distribution function (**CDF**) of event coverage across all videos from four selected anomaly types in the SVAD dataset. As observed, the majority of videos (> 85%) exhibit an event coverage greater than 0.5. Since event coverage reflects the variety of events within a video, a higher coverage suggests a greater need for selecting critical frames to ensure effective and comprehensive analysis.

Fig. 7 further illustrates the trace of frame selection rate under varying bandwidth constraints. As shown, the frame selection rate dynamically adjusts over time in response to bandwidth fluctuations. Under higher bandwidth conditions, HoloTrace adopts a higher frame selection rate to make full use of the available bandwidth for enhanced anomaly detection and Bi-CKG updating on the cloud. This adaptive behavior is attributed to the integration of bandwidth modeling into the frame selection strategy (Eq. (8)).

Finally, Fig. 8 shows the variation of GPU usage on the cloud over time, where the maximum available GPU resource is capped at 50% during the frame selection decision process (Eq. (8)). As observed, the GPU usage fluctuates dynamically but consistently stays below the 50% threshold, as intended.

6 Conclusion

In this work, we propose HoloTrace, a novel system that leverages a bidirectional knowledge graph generated by the LLM for video anomaly detection. HoloTrace employs an edge-cloud collaborative architecture: the edge identifies anomaly events using IER and HBR, then transmits critical frames to the cloud for auxiliary detection; the cloud is responsible for constructing and updating the Bi-CKG graph with real-world events. To address the limitations of existing VAD datasets, we also develop a large-scale VAD dataset, SVAD, with frame-level annotations. Extensive experiments on both SVAD and existing datasets demonstrate that HoloTrace outperforms state-of-the-art approaches.

Acknowledgments

This work is supported by the Project of the Department of Strategic and Advanced Interdisciplinary Research of Pengcheng Laboratory under grand No 2025QYA001, the National Key Research and Development Program of China under grant No. 2022YFB3105000, and the Shenzhen Key Lab of Software Defined Networking under grant No. ZDSYS20140509172959989.

References

[1] Andra Acsintoae, Andrei Florescu, Mariana-Iuliana Georgescu, Tudor Mare, Paul Sumedrea, Radu Tudor Ionescu, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Ubnormal: New benchmark for supervised open-set video anomaly detection. In *CVPR 2022*. IEEE, Louisiana, United States, 20111–20121.

[2] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and Daviv Reinitz. 2008. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE transactions on pattern analysis and machine intelligence* 30, 3 (2008), 555–560.

[3] Congqi Cao, Xin Zhang, Shizhou Zhang, Peng Wang, and Yanning Zhang. 2022. Adaptive graph convolutional networks for weakly supervised anomaly detection in videos. *IEEE Signal Processing Letters* 29 (2022), 2497–2501.

[4] Chenglizhao Chen, Xinyu Liu, Mengke Song, Luming Li, Xu Yu, and Shanchen Pang. 2024. Unveiling Context-Related Anomalies: Knowledge Graph Empowered Decoupling of Scene and Action for Human-Related Video Anomaly Detection. *ArXiv* abs/2409.03236 (2024).

[5] Dongyue Chen, Pengtao Wang, Lingyi Yue, Yuxin Zhang, and Tong Jia. 2020. Anomaly detection in surveillance video based on bidirectional prediction. *Image and Vision Computing* 98 (2020), 103915.

[6] Yang Cong, Junsong Yuan, and Ji Liu. 2011. Sparse reconstruction cost for abnormal event detection. In *CVPR 2011*. IEEE, Colorado Springs, United States, 3449–3456.

[7] Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *ArXiv* abs/2205.09712 (2022).

[8] Douyin. 2025. *Douyin*. Retrieved Apr. 9, 2025 from https://www.douyin.com/

[9] Fujitsu. 2024. *Fujitsu AI White Paper - Causal Knowledge Graph*. Retrieved Apr. 9, 2025 from https://www.fujitsu.com/global/documents/about/research/article/202410-causal-knowledge-graph/202410_White-Paper-Casual-Knowledge-Graph_EN.pdf

[10] Mariana-Iuliana Georgescu, Antonio Barbalau, Radu Tudor Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. 2021. Anomaly detection in video via self-supervised and multi-task learning. In *CVPR 2021*. IEEE, Virtual Event, 12737–12747.

[11] Mariana Iuliana Georgescu, Radu Tudor Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. 2021. A background-agnostic framework with adversarial training for abnormal event detection in video. *IEEE transactions on pattern analysis and machine intelligence* 44, 9 (2021), 4505–4523.

[12] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *CVPR 2023*. IEEE, Vancouver, Canada, 15180–15190.

[13] Google. 2025. *Google*. Retrieved Apr. 9, 2025 from https://www.google.com/videohp

[14] Wenyi Hong, Weihan Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. 2024. Cogvlm2: Visual language models for image and video understanding. *ArXiv* abs/2408.16500 (2024).

[15] Xiao Huang, Jingyuan Zhang, Dingcheng Li, and Ping Li. 2019. Knowledge graph embedding based question answering. In *AAAI 2019*. Association for Computing Machinery, Honolulu, United States, 105–113.

[16] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *ArXiv* abs/2410.21276 (2024).

[17] Ammar Mansoor Kamoona, Amirali Khodadadian Gostar, Alireza Bab-Hadiashar, and Reza Hoseinnezhad. 2023. Multiple instance-based video anomaly detection using deep temporal encoding–decoding. *Expert Systems with Applications* 214 (2023), 119079.

[18] Federico Landi, Cees GM Snoek, and Rita Cucchiara. 2019. Anomaly locality in video surveillance. *ArXiv* abs/1901.10364 (2019).

[19] Shuo Li, Fang Liu, and Licheng Jiao. 2022. Self-training multi-sequence learning with transformer for weakly supervised video anomaly detection. In *AAAI 2022*. Association for the Advancement of Artificial Intelligence, Virtual Event, 1395–1403.

[20] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. 2013. Anomaly detection and localization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence* 36, 1 (2013), 18–32.

[21] Long Lian, Baifeng Shi, Adam Yala, Trevor Darrell, and Boyi Li. 2023. Llm-grounded video diffusion models. *ArXiv* abs/2309.17444 (2023).

[22] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *CVPR 2024*. IEEE, Seattle, United States, 26286–26296.

[23] Jiaqi Liu, Qin Zhang, Luoyi Fu, Xinbing Wang, and Songwu Lu. 2019. Evolving knowledge graphs. In *INFOCOM 2019*. IEEE, Paris, France, 2260–2268.

[24] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. 2018. Future frame prediction for anomaly detection–a new baseline. In *CVPR 2018*. IEEE, Salt Lake City, United States, 6536–6545.

[25] Yi Liu, Haowen Hou, Fei Ma, Shiguang Ni, and Fei Richard Yu. 2025. MLLM-TA: Leveraging Multimodal Large Language Models for Precise Temporal Video Grounding. *IEEE Signal Processing Letters* 32 (2025), 281–285.

[26] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. 2021. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*. IEEE, Virtual Event, 13568–13577.

[27] Cewu Lu, Jianping Shi, and Jiaya Jia. 2013. Abnormal event detection at 150 fps in matlab. In *ICCV 2013*. IEEE, Sydney, Australia, 2720–2727.

[28] Weixin Luo, Wen Liu, and Shenghua Gao. 2017. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *ICCV 2017*. IEEE, Venice, Italy, 341–349.

[29] Hualin Ma and Liyan Zhang. 2022. Attention-based framework for weakly supervised video anomaly detection. *The Journal of Supercomputing* 78, 6 (2022), 8409–8429.

[30] Rashmika Nawaratne, Damminda Alahakoon, Daswin De Silva, and Xinghuo Yu. 2019. Spatiotemporal anomaly detection using deep learning for real-time video surveillance. *IEEE Transactions on Industrial Informatics* 16, 1 (2019), 393–402.

[31] Alina Nesen and Bharat Bhargava. 2020. Knowledge graphs for semantic-aware anomaly detection in video. In *AIKE 2020*. IEEE, California, United States, 65–70.

[32] Trong-Nguyen Nguyen and Jean Meunier. 2019. Anomaly detection in video sequence with appearance-motion correspondence. In *ICCV 2019*. IEEE, Seoul, Korea, 1273–1283.

[33] NVIDIA. 2025. *Jetson Xavier NX*. Retrieved Apr. 9, 2025 from https://www.nvidia.com/en-sg/autonomous-machines/embedded-systems/jetson-xavier-nx/

[34] Chaewon Park, MyeongAh Cho, Minhyeok Lee, and Sangyoun Lee. 2022. FastAno: Fast anomaly detection via spatio-temporal patch transformation. In *WACV 2022*. IEEE, Waikoloa, United States, 1908–1918.

[35] Alejandro Peña, Humberto Sossa, and Agustín Gutiérrez. 2008. Causal knowledge and reasoning by cognitive maps: Pursuing a holistic approach. *Expert Systems with Applications* 35, 1-2 (2008), 2–18.

[36] Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. 2013. Knowledge graph identification. In *ISWC 2013*. Springer Berlin Heidelberg, Sydney, Australia, 542–557.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, Virtual Event, 8748–8763.

[38] Nicolae-C Ristea, Florinel-Alin Croitoru, Radu Tudor Ionescu, Marius Popescu, Fahad Shahbaz Khan, Mubarak Shah, et al. 2024. Self-distilled masked auto-encoders are efficient video anomaly detectors. In *CVPR 2024*. IEEE, Seattle, United States, 15984–15995.

[39] Kelathodi Kumaran Santhosh, Debi Prosad Dogra, and Partha Pratim Roy. 2020. Anomaly detection in road traffic using visual surveillance: A survey. *ACM Computing Surveys (CSUR)* 53, 6 (2020), 1–26.

[40] Waqas Sultani, Chen Chen, and Mubarak Shah. 2018. Real-world anomaly detection in surveillance videos. In *CVPR 2018*. IEEE, Salt Lake City, United States, 6479–6488.

[41] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, et al. 2023. Video understanding with large language models: A survey. *ArXiv* abs/2312.17432 (2023).

[42] Yao Tang, Lin Zhao, Shanshan Zhang, Chen Gong, Guangyu Li, and Jian Yang. 2020. Integrating prediction and reconstruction for anomaly detection. *Pattern Recognition Letters* 129 (2020), 123–130.

[43] Chenchen Tao, Chong Wang, Sunqi Lin, Suhang Cai, Di Li, and Jiangbo Qian. 2024. Feature Reconstruction with Disruption for Unsupervised Video Anomaly Detection. *IEEE Transactions on Multimedia* 26 (2024), 10160–10173.

[44] Yu Tian, Guansong Pang, Yuanhong Chen, Rajvinder Singh, Johan W Verjans, and Gustavo Carneiro. 2021. Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In *Proceedings of the IEEE/CVF international conference on computer vision*. IEEE, Virtual Event, 4955–4966.

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *ArXiv* abs/2302.13971 (2023).

[46] Hongyong Wang, Xinjian Zhang, Su Yang, and Weishan Zhang. 2021. Video anomaly detection by the duality of normality-granted optical flow. *ArXiv* abs/2105.04302 (2021).

[47] Shu Wang and Zhenjiang Miao. 2010. Anomaly detection in crowd scene. In *IEEE 10th International Conference on Signal Processing Proceedings*. IEEE, Beijing, China, 1975–1981.

[48] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. 2023. Cogvlm: Visual expert for pretrained language models. *ArXiv* abs/2311.03079 (2023).

[49] Xuanzhao Wang, Zhengping Che, Bo Jiang, Ning Xiao, Ke Yang, Jian Tang, Jieping Ye, Jingyu Wang, and Qi Qi. 2021. Robust unsupervised video anomaly detection by multipath frame prediction. *IEEE transactions on neural networks and learning systems* 33, 6 (2021), 2301–2312.

[50] Peng Wu, Xuerong Zhou, Guansong Pang, Zhiwei Yang, Qingsen Yan, Peng Wang, and Yanning Zhang. 2024. Weakly supervised video anomaly detection and localization with spatio-temporal prompts. In *Proceedings of the 32nd ACM*

International Conference on Multimedia. Association for Computing Machinery, New York, United States, 9301–9310.

[51] Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science* 18, 6 (2024), 186357.

[52] Yuchen Yang, Kwonjoon Lee, Behzad Dariush, Yinzhi Cao, and Shao-Yuan Lo. 2025. Follow the rules: reasoning for video anomaly detection with large language models. In *ECCV 2024*. Springer Nature Switzerland, Milan, Italy, 304–322.

[53] Yuan Yuan, Dong Wang, and Qi Wang. 2016. Anomaly detection in traffic scenes via spatial-aware motion reconstruction. *IEEE Transactions on Intelligent Transportation Systems* 18, 5 (2016), 1198–1209.

[54] Weichao Yue, Jianing Chai, Xiaoxue Wan, Yongfang Xie, Xiaofang Chen, and Weihua Gui. 2023. Root cause analysis for process industry using causal knowledge map under large group environment. *Advanced Engineering Informatics* 57 (2023), 102057.

[55] Sanggeon Yun, Ryozo Masukawa, Minhyoung Na, and Mohsen Imani. 2024. Missiongnn: Hierarchical multimodal gnn-based weakly supervised video anomaly recognition with mission-specific knowledge graph generation. *ArXiv* abs/2406.18815 (2024).

[56] Luca Zanella, Willi Menapace, Massimiliano Mancini, Yiming Wang, and Elisa Ricci. 2024. Harnessing Large Language Models for Training-free Video Anomaly Detection. In *CVPR 2024*. IEEE, Seattle, United States, 18527–18536.

[57] Huaxin Zhang, Xiaohao Xu, Xiang Wang, Jialong Zuo, Chuchu Han, Xiaonan Huang, Changxin Gao, Yuehuan Wang, and Nong Sang. 2024. Holmes-VAD: Towards Unbiased and Explainable Video Anomaly Detection via Multi-modal LLM. *ArXiv* abs/2406.12235 (2024).

[58] Joey Tianyi Zhou, Jiawei Du, Hongyuan Zhu, Xi Peng, Yong Liu, and Rick Siow Mong Goh. 2019. Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security* 14, 10 (2019), 2537–2550.