

帮我在这个版本的基础上，保持前3段不变，挑战和做法重新修改下，挑战上，高效协作没有问题，造成这个挑战的3个方面呢，第一个，我想分别写成高帧率的流视频怎么处理；第二个，我想写成任务如何分配给这些智能体（因为大任务的拆分细粒度问题，但是你需要跟高效协作联系起来讲挑战）；第三个，写智能体高效的交互规则，最小化推理，最大化准确率。做法部分呢，刚好和这三个挑战对应起来；第一个我想写成每个智能体配备触发机制，关键片段触发；第二个呢设计了一个自适应的任务分配机制，根据任务复杂度建立智能体，并分配任务；第三个呢，设计了一个高效交互机制。

在安防与交通管控等场景中，流视频监控的核心目标是实时解析摄像头画面、判定场景变化并触发响应，要求在秒级时延内输出可用于决策的稳定语义结果。传统方法依赖于目标检测、语义分割与跟踪等单任务模型，这些方法能够识别对象及其轨迹，但在长时序数据中却难以捕捉事件间的因果与叙事关系。随着视觉语言模型（VLM）的引入，事件级语义理解得到了显著提升。VLM可以有效地识别诸如“打斗”或“拥堵”等宏观事件，为流视频理解提供了新的方向。

然而，尽管VLM在事件级理解上表现出色，它仍在对象级的精细认知上存在不足。事件由多个对象构成，其成因与后果往往取决于对象的数量、位置以及它们之间的交互关系。例如，在治安场景中，我们需要明确参与者的身份、人数和行为；在交通事故处理中，需要识别责任车辆、受损情况和关联方规模。尽管某些研究通过微调等方式，增强了VLM对对象的识别能力，但这通常以牺牲事件级语义建模为代价。这样一来，VLM在流视频监控中无法同时保持高质量的事件级与对象级表征，限制了其在复杂监控场景下的应用。

为了克服这一局限性，我们提出了一个多智能体协同的流视频理解框架。该框架通过将事件级和对象级的理解任务分配给不同的智能体，从而实现高效的协同工作。一部分智能体专注于事件级语义建模，抽取场景的叙事骨架；另一部分智能体则聚焦于对象级的识别与关系解析。通过跨层融合，这些智能体协同工作，输出既能提供全局语义，又能捕捉细节的统一结果。通过语义分层与功能解耦，框架避免了单一模型在不同语义层次上的性能拉扯，赋予系统在复杂流视频下既能“看得全”又能“看得细”的能力。

然而，实现多智能体的高效协同以应对流视频分析的实时性需求，仍面临三方面挑战。首先是高帧率流视频的处理效率。流视频往往以数十帧每秒的速度持续输入，各智能体需要在毫秒级响应中完成场景分析与语义判断。如何在不牺牲时效的情况下过滤冗余信息、聚焦关键片段，是高效协作的首要挑战。其次是任务的细粒度分配问题。事件级与对象级任务在复杂度、频率和计算负载上差异显著，如果缺乏自适应的任务拆解与分配机制，就可能造成部分智能体过载、部分闲置，从而削弱整体协作效率。如何在动态场景中合理地将分析任务拆分并分配给合适的智能体，是系统性能优化的核心。第三是智能体间的高效交互机制。不同层次的智能体需要频繁交换语义信息以保持一致性，但过多的通信与冗余推理会导致延迟增加。如何建立高效的语义交互规则，在最小化推理开销的同时最大化全局理解的准确性，是系统可扩展性的关键。

针对上述挑战，我们提出了三个层次的解决方案。首先，为应对高帧率流视频的处理需求，我们为每类智能体设计了关键片段触发机制。智能体可根据场景变化和历史状态动态判断是否执行分析任务，仅在潜在语义突变或关键事件发生时触发推理，从源头上抑制冗余计算、降低时延。其次，为解决任务分配的动态适配问题，我们提出了自适应任务分配机制。系统根据当前任务复杂度与各智能体特性自动生成任务图谱，并在执行过程中动态调整任务边界，使每个智能体始终工作在最合适的粒度和负载区间，确保算力与信息流的最优分配。最后，为提升智能体间的语义协作效率，我们设计了高效语义交互机制。该机制允许事件智能体向对象智能体提供语境提示（如在“抢劫”场景中优先关注人数、武器与受害者），对象智能体则反向反馈数量、状态与交互证据，强化事件理解。通过轻量级语义压缩与一致性对齐，系统在保持低通信成本的同时实现了跨层面的语义统一。



(a) **TAO** first generate bounding boxes to identify objects in each frame. (b) Next, **TAO** score these boxes using object-centric video anomaly detection algorithms to extract potential anomalous boxes. To ensure robustness, **TAO** apply filtering to eliminate redundant boxes. (c) Finally, the filtered boxes and original frames are input into a prompt-based segmentation model to produce pixel-level anomaly segmentation masks.

### 三、左侧区域

#### ① Sec.3.1 – Bounding Boxes Extraction (目标框提取)

从视频帧序列 ( $f_1, f_2, \dots, f_N$ ) 中检测出所有物体，例如人、车等。

→ 输出：每帧的一组矩形框 (Bounding Boxes)。

#### ② Sec.3.1 – Anomalous Boxes Extraction (异常框提取)

对每个检测到的框计算异常分数（根据速度、姿态、运动轨迹等特征）。

如果分数高于阈值 ( $\tau$ )，就认为该目标可能异常。

#### ③ Sec.3.2 – Robust Filtering of Anomalous Boxes (鲁棒过滤)

过滤掉短暂出现、重叠或错误的异常框，只保留时间上持续出现的“真实异常对象”。

#### ④ Sec.3.3 – Segmentation Model Inference (分割推理)

将过滤后的异常框输入 SAM2 模型 (Segment Anything Model 2)，生成像素级分割结果。

最终输出：

→ Anomaly Segmentation (异常区域分割视频)，即每个异常物体的红色掩码区域。

小结：左栏是整张图的“目录 + 时间轴”，展示 TAO 的四阶段主流程。

### 三、中间区域

异常框提取：从检测结果中筛选潜在异常对象。

- 顶部四张图显示普通目标检测的输出：绿色框表示检测到的所有对象；
- 底部两张图显示筛选出的潜在异常目标：黄色框表示异常分数超过阈值 ( $\tau$ ) 的对象；
- 中间黄色“ $\Delta$ ”标志提示：这些是候选异常框，还需后续验证；

- 时间顺序箭头 (从  $f_{\{i-1\}}$  到  $f_{\{i+1\}}$ ) 表示帧的连续关系。

输入：每帧检测到的所有目标框

输出：一批“潜在异常框 (Potential Boxes) ”

鲁棒过滤：TAO 的关键创新，利用时间一致性筛选真正的异常目标。

图中展示内容：

- 左至右三帧：( $f_{\{i-1\}}$ )、( $f_{\{i\}}$ )、( $f_{\{i+1\}}$ )
- 绿色和黄色框已经被替换为红框：红框代表“通过鲁棒性验证的稳定异常对象”；
- 箭头表示这些红框在连续帧中保持一致；
- 该过程使用论文中的 Boxes Robustness Filtering Algorithm (红色标签) 完成。

输出：稳定、时序一致的异常框 (Anomalous Boxes) 。

分割推理：使用 SAM2 (Segment Anything Model 2) 来生成最终的像素级分割结果。

步骤说明：

- 将经过过滤的异常框 (红色 × 标记) 作为提示输入；
- 仅在每隔 1 帧选取关键帧，减少计算；
- SAM2 根据这些提示生成整段视频的分割掩码；
- 输出：红色掩码覆盖的异常区域 (Anomaly Segmentation) 。

输出：每帧对应的像素级异常掩码 (mask) 。

### 三、右侧区域 (Boxes Robustness Filtering Algorithm 详细逻辑)

Step 0: Initialization (初始化)

Step 1: Inherit Existing Anomalous Boxes (继承已有异常框)

Step 2: Assign New Anomalous Boxes (分配新异常框)

Step 3: Save Boxes Every 1 Frames (每隔 1 帧保存一次)