

486

487 **VADTree: Explainable Training-Free Video Anomaly Detection**

488

**via Hierarchical Granularity-Aware Tree****Technical Appendices**

489

490 **Table of Contents**

491

**A Hierarchical Granularity-aware Tree 15**

492

A.1 TreeInit: Granularity-Aware Binary Tree Construction . . . . . 15

493

A.2 Proof of Coverage Completeness in Hierarchical Coarse-Fine Clustering . . . . . 15

494

**B Generic Event-centric Anomaly Scoring and Refining 16**

495

B.1 Prior-infused Node Scoring . . . . . 16

496

**C More Results 19**

497

C.1 More Experimental Details . . . . . 19

498

C.2 Effect of Intra-cluster Node Refinement Hyperparameters. . . . . 20

499

C.3 Effect of Inter-cluster Node Correlation Hyperparameters. . . . . 20

500

C.4 More Ablation Experiment . . . . . 21

501

C.5 Comparison of VADTree and Different Video Sampling Methods . . . . . 21

502

C.6 Stability Analysis of Main Results . . . . . 22

503

C.7 Additional Qualitative Results &amp; Case Studies . . . . . 22

504

**D Limitations 22**

505

**E Broader Societal Impacts 22**

506

507

508

509 The appendix begins by detailing the algorithmic process underlying the construction of the HGTree  
 510 and establishing the proof of its representational completeness. Furthermore, it presents additional  
 511 experimental details, including extensive ablation studies and comparative analyses. Finally, the  
 512 appendix examines the limitations of this work and its potential societal impact.

## 513 A Hierarchical Granularity-aware Tree

### 514 A.1 TreeInit: Granularity-Aware Binary Tree Construction

---

**Algorithm 1** TreeInit: Granularity-Aware Binary Tree Construction Algorithm (Section 3.1)

---

**Require:**

Video  $V_{1:T}$  with  $T$  frames,  
 Confidence scores  $\hat{C} = \{(\tau_i, \hat{c}_i)\}_{i=1}^N$ ,  
 Confidence threshold  $\gamma_{\min}$

**Ensure:** Binary tree  $\mathcal{T} = \{([s_j, e_j], [\hat{c}_s^j, \hat{c}_e^j])\}_{j=1}^M$

```

1:  $\mathcal{T} \leftarrow \emptyset, \mathcal{U} \leftarrow \emptyset$  ▷ Result set & consumed split points
2: Push root node  $\mathcal{D} \leftarrow [[1, T]]$  ▷ DFS stack initialization
3: while  $\mathcal{D} \neq \emptyset$  do
4:    $[l, r] \leftarrow \mathcal{D}.\text{pop}()$ 
5:    $\hat{c}_l \leftarrow \mathbb{I}(l = 1) \cdot 1 + \mathbb{I}(l > 1) \cdot \hat{c}_l$  ▷ Left boundary confidence
6:    $\hat{c}_r \leftarrow \mathbb{I}(r = T) \cdot 1 + \mathbb{I}(r < T) \cdot \hat{c}_r$  ▷ Right boundary confidence
7:    $\mathcal{T}.\text{add}([l, r], [\hat{c}_l, \hat{c}_r])$ 
8:   Find split  $\tau^* \leftarrow \arg \max_{\tau \in (\Psi \setminus \mathcal{U}) \cap (l, r)} \hat{c}_\tau$  ▷ Select the highest remaining confidence point
9:   if  $\hat{c}_{\tau^*} \geq \gamma_{\min}$  then
10:     $\mathcal{U}.\text{add}(\tau^*)$ 
11:     $\mathcal{D}.\text{push}([\tau^*, r])$  ▷ Right child
12:     $\mathcal{D}.\text{push}([l, \tau^*])$  ▷ Left child
13:   end if
14: end while
15: return Sort( $\mathcal{T}, l_j \uparrow$ ) ▷ Sort by start time

```

---

### 515 A.2 Proof of Coverage Completeness in Hierarchical Coarse-Fine Clustering

516 **Theorem 1** (Coverage Completeness). *Based on the method described in Section 3.1, we get*  
 517  $\mathcal{T}' = (S'_{\text{coarse}}, S'_{\text{fine}})$ , *where  $|S'_{\text{coarse}}| = M'_c$  and  $|S'_{\text{fine}}| = M'_f$ . Then:*

518 *The original video sequence  $V_{1:T}$  can be exactly reconstructed through temporal concatenation of*  
 519 *segments from either the coarse cluster  $S'_{\text{coarse}}$  or the fine cluster  $S'_{\text{fine}}$ .*

$$\bigcup_{\mathcal{N}_i \in S'_{\text{coarse}}} [l_i, r_i] = [1, T], \quad \bigcup_{\mathcal{N}_i \in S'_{\text{fine}}} [l_i, r_i] = [1, T]. \quad (11)$$

520 *Notations:*

- 521 •  $\mathcal{N}_i = ([l_i, r_i], [\hat{c}_l^{(i)}, \hat{c}_r^{(i)}])$ : A tree node represents a generalized event video segment, with  
 522 boundary frames and their confidences as  $[l_i, r_i]$  and  $[\hat{c}_l^{(i)}, \hat{c}_r^{(i)}]$  respectively.
- 523 •  $\prec$ : Parent-child relation in  $\mathcal{T}$  ( $\mathcal{N}_j \prec \mathcal{N}_i \iff \mathcal{N}_i$  is a child of  $\mathcal{N}_j$ )
- 524 •  $\mathcal{T}_{\text{leaf}} \triangleq \{\mathcal{N}_i \in \mathcal{T} \mid \nexists \mathcal{N}_j \prec \mathcal{N}_i\}$ : Leaf node set of  $\mathcal{T}$

525 **Proof. Part 1: Initial Coverage Guarantee** The root node  $\mathcal{N}_0 = ([1, T], [1, 1]) \in \mathcal{T}$  spans  
 526 the full video by definition. Through iterative splitting in Algorithm 1, each parent node  $\mathcal{N}_p =$   
 527  $([l_p, r_p], [\hat{c}_l^{(p)}, \hat{c}_r^{(p)}])$  is partitioned into non-overlapping child nodes:

$$\mathcal{N}_c^L = ([l_p, \tau^*], [\hat{c}_l^{(c)}, \hat{c}_{\tau^*}^{(c)}]), \quad \mathcal{N}_c^R = ([\tau^*, r_p], [\hat{c}_{\tau^*}^{(c)}, \hat{c}_r^{(c)}]) \quad (12)$$

528 where  $\tau^* \in (l_p, r_p)$ . This implies:

$$\bigcup_{\mathcal{N}_i \in \mathcal{T}_{leaf}} [l_i, r_i] = [1, T] \quad (13)$$

529 **Part 2: Coarse Cluster Guarantee** The RemoveDup operator filters nodes through:

$$\mathcal{S}'_{coarse} = \{\mathcal{N}_i \in \mathcal{S}_{coarse} \mid \nexists \mathcal{N}_j \prec \mathcal{N}_i\} \quad (14)$$

530 As the above operation exclusively targets non-leaf nodes in  $\mathcal{S}_{coarse}$  and leaves leaf nodes unchanged.  
 531 Therefore, the current leaf nodes satisfies the expression completeness of the original video shown in  
 532 Eq. 13, and then satisfies the first item of Eq. 11:  $\bigcup_{\mathcal{N}_i \in \mathcal{S}'_{coarse}} [l_i, r_i] = [1, T]$ .

533 **Part 3: Fine Cluster Guarantee**

534 The Complete operator ensures coverage via two mechanisms:

535 1. Boundary Alignment: For edge cases:

$$\begin{aligned} &\text{if } \min_{\mathcal{N}_i \in \mathcal{S}'_{fine}} l_i > 1 : \text{insert } \mathcal{N}_1 \text{ from } \mathcal{S}'_{coarse} \\ &\text{if } \max_{\mathcal{N}_i \in \mathcal{S}'_{fine}} r_i < T : \text{append } \mathcal{N}_{M'_c} \text{ from } \mathcal{S}'_{coarse} \end{aligned} \quad (15)$$

536 2. Gap Bridging: For any adjacent nodes  $\mathcal{N}_i = ([l_i, r_i], [\hat{c}_l^{(i)}, \hat{c}_r^{(i)}])$  and  $\mathcal{N}_{i+1} =$   
 537  $([l_{i+1}, r_{i+1}], [\hat{c}_l^{(i+1)}, \hat{c}_r^{(i+1)}])$  in  $\mathcal{S}'_{fine}$  with  $r_i < l_{i+1}$ :

$$\exists \{\mathcal{N}_c\} \subset \mathcal{S}'_{coarse} \text{ s.t. } \bigcup_c [l_c, r_c] = [r_i, l_{i+1}] \quad (16)$$

538 Through Eq. 15 and Eq. 16, the second term of Eq. 11 is satisfied:  $\bigcup_{\mathcal{N}_i \in \mathcal{S}'_{fine}} [l_i, r_i] = [1, T]$ .

539 **Conclusion:** Both coarse and fine cluster maintain complete temporal coverage through the Sec-  
 540 tion 3.1 process.  $\square$

## 541 B Generic Event-centric Anomaly Scoring and Refining

### 542 B.1 Prior-infused Node Scoring

543 This section mainly supplements the prompt details used by VLM and LLM. First, by employing  
 544  $P_b \circ P_c$  (as demonstrated in Section B.1.1), we input the prompt into the LLM [5]<sup>1</sup> to derive prior  
 545 knowledge that excludes ill-posed semantic cues. The prior knowledge  $B$  are shown in Table 5 and  
 546 Table 6 respectively.

547 The model configuration details of the VLM for describing video content and the LLM for scoring  
 548 anomalies are consistent with their open source repositories<sup>2 3</sup>.

#### 549 B.1.1 Multidimensional Prior Knowledge Generation Prompt

550 **UCF-Crime** "To help video anomaly detection agent review the occurrence of abnormal events, it  
 551 is now necessary to pre-analyze possible anomalies to establish a prior knowledge base that matches  
 552 abnormal events. The video taken has no sound, and may have a long distance or a blurry picture.  
 553 There may be Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fighting, RoadAccidents, Robbery,  
 554 Shooting, Shoplifting, Stealing and Vandalism 13 types of events. Please carefully analyze these scenes.  
 555 Then point out the characteristics of each abnormal event from the following three perspectives: the  
 556 scene environment, characters or specific objects, actions or behaviors that occurred."

<sup>1</sup><https://chat.deepseek.com/>

<sup>2</sup><https://huggingface.co/lmms-lab/LLaVA-NeXT-Video-7B>

<sup>3</sup><https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-14B>

**XD-Violence** “To help video anomaly detection agent review the occurrence of abnormal events, it is now necessary to pre-analyze possible anomalies to establish a prior knowledge base that matches abnormal events. The video taken has no sound, and may have a long distance or a blurry picture. There may be Abuse, Explosion, Fighting, Car Accident, Shooting and Riot 6 types of events. Please carefully analyze these scenes. Then point out the characteristics of each abnormal event from the following three perspectives: the scene environment, characters or specific objects, actions or behaviors that occurred.”

### B.1.2 Multidimensional Prior Knowledge

The prior knowledge  $B$  generated for the UCF-Crime and XD-Violence datasets are shown in Table 5 and Table 6 respectively.

#### UCF-Crime

Table 5: Multidimensional Prior Knowledge of UCF-Crime Dataset.

Abnormal Event Type	Scene Environment Features	Character/Object Features	Action/Behavior Features
Abuse	Secluded spaces (indoors/corners), non-public areas (private locations)	Two parties in physical conflict (perpetrator/victim), dragging tools (ropes/clubs)	Shoving/dragging, repeated hitting, restraining movement (pinning down)
Arrest	Public areas (streets/squares), zones with police vehicles or officers	Uniformed police, handcuffs, batons or firearms	Forced restraint, frisking, escorting to vehicles, lying on the ground
Arson	Areas with flammable materials (warehouses/vehicles), abnormal smoke/flames	Individuals holding flammable containers (gasoline bottles), ignition tools (lighters)	Throwing incendiary objects, fleeing quickly, repeatedly checking the fire
Assault	Narrow passages, crowded areas with sudden dispersion (subway stations/bar entrances)	Armed individuals (knives/blunt weapons), victims struggling on the ground	Sudden lunging, weapon swinging, victims adopting defensive postures
Burglary	Damaged doors/windows, unlit buildings at night, surveillance blind spots (back alleys)	Masked/dark-clothed individuals, lock-picking tools (pliers), backpacks (for loot)	Peering through windows, picking locks, rummaging through items
Explosion	Smoke spreading, flying debris, crowds fleeing outward from a central point	Suspicious packages/vehicles, post-explosion wreckage (metal fragments)	Throwing motions, sudden flash of flames, crowds crouching/running
Fighting	Public spaces (restaurants/stadiums) with concentrated physical conflicts, overturned furniture	Multiple people entangled, bleeding faces, torn clothing	Punching/kicking, hair-pulling, siege

Road Accidents	Collision points (intersections/curves), skid marks, scattered debris, traffic congestion	Deformed vehicles, deployed airbags, paramedics (uniforms/stretchers)	Sudden braking, vehicle rollovers, pedestrians being hit
Robbery	Streets/ATM areas, fast-moving vehicles (motorcycles/cars)	Threats with guns/knives, motorcycle helmets (face concealment), stolen items (bags)	Snatching and fleeing, threatening gestures, vehicles abruptly stopping/accelerating
Shooting	Crowds suddenly ducking/fleeing, vehicles braking abruptly, bullet holes in windows	Gun-wielding individuals, gunshot victims falling, spent shell casings	Aiming firearms, continuous firing, seeking cover
Shoplifting	Loitering near shelves, surveillance blind spots (corners), suspicious concealment (coats)	Frequently observing staff, hiding items (in bags/under clothing)	Concealing items in clothing, glancing around nervously, quickly leaving shelves
Stealing	Crowded areas (subways/markets), sudden disappearance of target items (wallets/phones)	Close proximity to victims, distractions (e.g., bumping), rapid transfer of stolen goods	Pickpocketing (hands reaching into pockets), passing loot to accomplices
Vandalism	Graffiti-covered walls, shattered glass, toppled public facilities (trash cans/fences)	Spray paint cans, hammers/stones, targets (cameras/glass)	Smashing motions, spraying walls, kicking facilities

**Table Notes:**

1. Scene environment features capture spatial anomalies (e.g., secluded corners) and physical damage patterns
2. Character/object features focus on suspicious entities and high-risk items
3. Action/behavior features characterize motion dynamics critical for low-quality video analysis

**Recognition Tips:**

1. **Blurry footage:** Track group behavior changes (crowd fleeing patterns)
2. **Long-distance:** Monitor environmental dynamics (smoke/glass shattering)
3. **Silent videos:** Analyze action intensity (repeated hitting motions)

568 **XD-Violence**

Table 6: Multidimensional Prior Knowledge of XD-Violence Dataset.

Abnormal Event Type	Scene Environment Features	Character/Object Features	Action/Behavior Features
Abuse	Secluded or private settings (alleyways, dimly lit rooms), lack of bystanders	Dominant/submissive individuals with indistinct blunt objects (belts, sticks)	Sudden aggressive movements (hitting/grabbing), victim recoiling/fleeing, prolonged physical contact

Explosion	Sudden bright flash with smoke/fire, structural damage (collapsed walls)	Chaotically moving people, objects near blast source (vehicles, trash bins)	Rapid light/smoke expansion, crowd scattering, lingering smoke/flames
Fighting	Public areas with bystanders (streets, bars), overturned furniture	Aggressive postures, broken objects nearby (bottles, chairs)	Repetitive punching/kicking, high-intensity movements, bystander reactions
Car Accident	Roads with sudden traffic stops, skid marks/debris (broken glass)	Collided vehicles (distorted shapes), injured individuals	Rapid deceleration/impact, post-crash behaviors (crowd gathering, emergency services)
Shooting	Public spaces with visible escape routes (malls, streets)	Shooter’s stance (arm extended), collapsing victims	Crowd panic (ducking/sprinting), post-shooting patterns (bodies, law enforcement)
Riot	Large crowds with fires/smoke, broken infrastructure (smashed windows)	Protesters with makeshift weapons, riot police/armored vehicles	Group violence (throwing objects), chaotic dispersal patterns (wave-like movements)

**Low-quality Analysis Guidelines:**

- Focus on movement patterns (sudden scattering/aggressive postures)
- Detect environmental changes (smoke/debris) over fine details
- Analyze group dynamics (crowd panic/clustering)
- Prioritize spatiotemporal anomalies (light flashes/vehicle stops)

**B.1.3 Video Content Description Prompt**

If directly driving the VLM to describe the video, its prompt  $P_d$  is as follows:

*"Please describe the content in the video in detail."*

If prior knowledge  $B$  is incorporated, the form of  $B \circ P_d$  is adopted.

**B.1.4 Abnormal Scoring Prompt**

The prompt  $P_s$  used in this section is divided into two parts: Scoring Criteria and Output Format Control.

**Scoring Criteria** *"How would you rate the scene described on a scale from 0 to 1, with 0 representing a standard scene and 1 denoting a scene with suspicious or potentially criminal activities?"*

**Output Format Control** *"Please provide the response in the form of a Python list and respond with only one number in the provided list below [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0] without any textual explanation. It should begin with '[' and end with ']'."*

**C More Results**

**C.1 More Experimental Details**

Based on the experimental details described in the Section 4, the  $\gamma_{min} = 0.4$  and K-Means clustering algorithm are used to generate the inference of HGTree. In the inter-cluster node refinement process,

we implemented a top-K control for the final weighted neighborhood node numbers. Additionally, This process also includes the temperature parameter  $\tau$  of softmax. In the Inter-cluster Node Correlation, the hyperparameter beta affects the weight of coarse and fine Clusters in the final anomaly score.

Our experiment is conducted based on  $2 \times 32\text{GB}$  GPUs. Performing a complete component experiment on dataset UCF-Crime takes approximately 32 hours. The vast majority of computational resources are consumed in the inference process of VLM and LLM.

## C.2 Effect of Intra-cluster Node Refinement Hyperparameters.

The neighborhood size parameter K in Eq. 8 governs the trade-off between localized feature precision and noise suppression. Table 8 demonstrates substantial AUC gains from neighborhood node refinement: 9.88% for fine cluster and 7.14% for coarse cluster when expanding K from 0 to 10. Both clusters exhibit maximal improvements within this critical initialization range. Performance stabilization occurs between K=10 and K=15, with fine cluster maintaining 83.03-83.05% AUC and coarse cluster 82.55-82.81%. Beyond K=15, gradual performance degradation confirms the optimal balance of contextual integration and noise suppression within this parameter range. The temperature

Table 7: Influence of top-K weighted neighborhood nodes on AUC.

K	0	5	10	15	20	25
Coarse Cluster AUC (%)	75.67	81.84	82.81	82.55	81.96	81.73
Fine Cluster AUC (%)	73.17	79.77	83.05	83.03	82.65	82.43

coefficient  $\tau$  in Eq. 8 regulates the entropy characteristics of Softmax-derived distributions while maintaining ordinal relationships between elements. Our empirical analysis (Table 8) reveals that as  $\tau$  approaches zero ( $\tau = 0.001$ ), the distribution collapses into a degenerate form concentrated solely on the maximal element equivalent to non-weighted selection. Progressive increases to moderate values result in stable AUC plateaus at 83.05% for fine-grained clusters with minimal variance. Notably, excessive temperature values ( $\tau = 100$ ) induce uniform distributions, degrading performance to 82.43% AUC for fine clusters. This analysis suggests optimal implementation parameters reside within  $\tau \in [0.01, 1]$ , balancing distribution sharpness with model stability. For the experiment of coarse cluster, we can get similar conclusions.

Table 8: Influence of softmax temperature  $\tau$  on AUC.

$\tau$	0.001	0.01	0.1	1	10	100
Coarse Cluster AUC (%)	78.72	80.68	82.81	82.42	82.21	82.20
Fine Cluster AUC (%)	77.83	80.72	83.05	83.05	83.02	83.02

## C.3 Effect of Inter-cluster Node Correlation Hyperparameters.

The  $\beta$  coefficient regulates parent-child node interplay in our cohesion-driven correlation (Eq. 10). As quantified in Table 9, the optimal control coefficient  $\beta = 0.3$  delivers peak AUC performance at 84.71%, indicating effective equilibrium between parent node contextual integration and child nodes semantic specificity. Additionally, limited AUC fluctuation demonstrates the hierarchy’s inherent noise suppression capability. This validates our variance weighted design as an effective strategy for multi-granularity fusion.

Table 9: Influence of inter-cluster-correlation control coefficient  $\beta$  on AUC

$\beta$	-0.3	-0.1	0	0.1	0.2	0.3	0.4	0.5	0.6
AUC (%)	84.48	84.51	84.55	84.58	84.66	84.71	84.70	84.64	84.57

#### C.4 More Ablation Experiment

We ablate different modules of our proposed method VADTree to prove the effectiveness of the proposed components, including fine cluster HGTree, prior-infused node scoring, intra-cluster nodes refinement and inter-cluster node correlation. We also tested the effectiveness of our components on the 10s fixed-length sliding temporal window (TW) sampling method. Table 10 shows the results of all ablated variants of VADTree. The experiment shows that each component has a significant impact on our final results. At the same time, these components are still effective for methods using fixed-length sliding temporal window sampling.

Table 10: Ablation study of VADTree components on the UCF-Crime dataset. The upper and lower panels present experiments using HGTree and 10 seconds fixed-length sliding temporal window (TW) sampling respectively.

HGTree	Fine Cluster	Prior-infused $f_{VLM}$	Refinement	Correlation	AUC (%)
✓	✓	✗	✓	✓	83.08
✓	✓	✓	✗	✓	77.97
✓	✓	✓	✓	✗	83.05
✓	✓	✓	✓	✓	<b>84.71</b>
✗	✗	✗	✗	✗	72.93
✗	✗	✓	✗	✗	75.21
✗	✗	✗	✓	✗	80.62
✗	✗	✓	✓	✗	82.81

#### C.5 Comparison of VADTree and Different Video Sampling Methods

In this experiment, we conducted a comparative analysis of VADTree against mainstream video sampling approaches, focusing on final anomaly detection performance and computational efficiency. The fixed-length sliding temporal window (TW) method, employed by LAVAD and VERA [51, 47], serves as our primary comparison method. Additionally, we propose three metrics to evaluate video sampling efficiency: (1) Number of Segments (NoS), defined as the total number of video segments sampled from the test dataset; (2) Mean Intersection over Union (mIoU), computed by first identifying the maximum temporal IoU between each anomalous event and all sampled segments within a video, then averaging these maximum values across all events; (3) Mean Intersection Frames (mIF), which quantifies the average number of abnormal event frames contained in the segment exhibiting the maximum temporal overlap with the event.

Table 11: Results of VADTree variants with different video sampling methods on the UCF-Crime Dataset. 16f represents a stride of 16 frames. **NoS** indicates the number of generated video segments. **mIoU** and **mIF** are used to measure the quality of video sampling.

Method	TW Length	Stride	NoS↓	mIoU↑	mIF↑	AUC (%)
sliding TW	5s	5s	7558	0.41	122	82.06
sliding TW	10s	10s	3852	0.40	191	82.81
sliding TW	20s	20s	1994	0.33	265	81.33
sliding TW	10s	16f [51, 47]	69634	0.51	210	82.87
VADTree-Coarse	-	-	2248	0.52	369	82.81
VADTree-Fine	-	-	6365	0.40	233	83.05
VADTree	-	-	8613	0.47	343	<b>84.71</b>

As demonstrated in Table 11, non-overlapping implementations of the TW strategy exhibit poor alignment with anomalous events. While dense overlapping sampling with short strides (16 frames) [51, 47] marginally improves AUC ROC it produces 8× more segments than VADTree, incurring significant computational costs without commensurate performance benefits. Our proposed VADTree achieves superior anomaly detection performance while maintaining comparable computational efficiency to non-overlapping TW baselines, demonstrating effective balance between precision and resource utilization.



## C.6 Stability Analysis of Main Results

Considering the high computational cost, we conducted error analysis on some of the main experiments and reported their mean and variance. The randomness of the experimental results mainly comes from the randomness of the generated content during VLM and LM inference. As shown in Table 12, the  $\delta$  across all configurations is statistically insignificant compared to the performance gaps between different methods (Table 1 and Table 2). This confirms that the observed performance is robust against experimental randomness.

Table 12: Stability analysis of VADTree with different HGTree configurations.

Method	Exp-1	Exp-2	Exp-3	Mean Results	$\delta$
VADTree-Coarse	82.81	82.75	82.92	82.83	0.17
VADTree-Fine	83.05	82.86	83.05	82.99	0.19
VADTree	84.71	84.49	84.73	84.64	0.24

## C.7 Additional Qualitative Results & Case Studies

In the example shown in the Fig. 4, the overall score volatility in fine cluster is high ([0.7, 0.9, 0.9, 0.9, 0.6] in ② and [0.9, 0.6, 0.9, 1.0] in ③), while the anomaly score of coarse cluster ② node is low. After refinement, the above situation has been improved, but the correct anomaly score may be suppressed (③: 0.8→0.64). In addition, since the anomaly reasoning of each node is independent and lacks mutual reference and dimension, this may lead to unstable anomaly scores in the same video segment. The final inter-cluster correlation resolves these shortcomings well. This example indicates that VADTree can significantly enhance the stability of training-free VAD inference.

## D Limitations

Like existing training-free VAD methods, VADTree’s performance relies heavily on the visual perception capabilities of VLMs. Most VLMs mainly focus on more complex semantic understanding and reasoning, and there are still significant limitations in accurately obtaining various small shallow abnormal semantics (such as the flame of a lighter after an explosion). This constraint may hinder accurate anomaly detection. If essential visual characteristics are not captured during the encoding stage, it becomes unlikely for VADTree to effectively carry out abnormal reasoning or perform temporal inter-cluster corrections. Therefore, a primary challenge for VLM based VAD is to guarantee that both visual and temporal features are adequately captured.

## E Broader Societal Impacts

Our training-free paradigm enables efficient video anomaly detection with minimal computational costs, yet its deployment in safety-critical scenarios (e.g., public surveillance) requires careful consideration of privacy-preserving mechanisms. While avoiding explicit biometric data processing, the memory bank design storing scene prototypes could theoretically retain sensitive environmental patterns. We advocate transparency audits to mitigate potential privacy risks in real-world implementations.

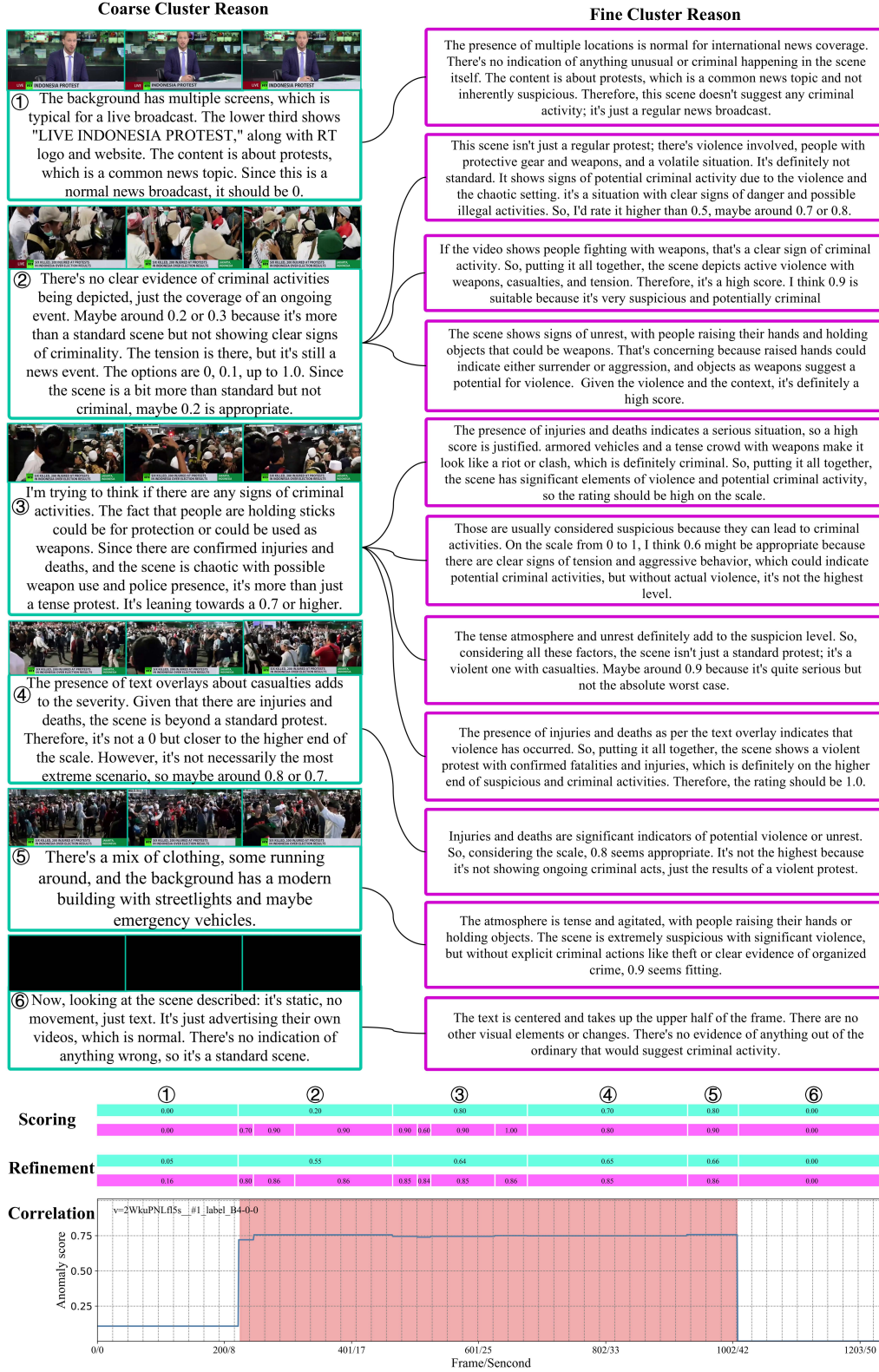


Figure 4: We showcase qualitative results obtained by VADTree on test video. We presented a video anomaly explanation (scoring based on HGTree (text description in the rectangular box), including initial anomaly score(Scoring), refined score(Refinement), and final anomaly score(Correlation)). Based on HGTree for video representation, the different granularity reasoning results of Coarse and fine clusters on anomalies can complement each other.