COPRA: Conditional Parameter Adaptation with Reinforcement Learning for Video Anomaly Detection

Darryl C. Jacob Auburn University
Xinyu Liu Auburn University
Kai Wang Tencent
Pan He Auburn University

Abstract

Vision-language models (VLMs) have demonstrated impressive performance in video anomaly detection (VAD), while providing interpretable predictions. However, we identify a key yet underexplored VAD issue: the mismatch between training and inference in both data distribution and model configuration. First, many VLM-based VAD approaches rely on a static parameter paradigm after training (or post-adaptation), causing them to overfit the training distribution and limiting generalization to unseen domains. Consequently, these methods lack effective test-time adaptation under substantial distribution shifts, such as novel environments or previously unseen anomaly types. Second, they treat VLMs as reasoning-based classifiers trained on extremely sparse frames from long videos, but perform inference on densely sampled short segments via sliding windows, leading to inherent inconsistencies between training and testing. To address these limitations, we propose COPRA, a conditional parameter adaptation framework for VLM-based VAD. Rather than relying on static post-training adaptation (e.g., fixed prompts or shared parameter updates), COPRA introduces instance-conditioned parameter adaptation, where a lightweight generator predicts input-specific parameter updates to dynamically modulate a frozen VLM on the fly, enabling per-segment adaptation during both training and inference. Empirically, our method achieves strong performance on standard VAD benchmarks. More broadly, it consistently outperforms static baselines in both in-domain and cross-domain settings, and further generalizes beyond VAD to unseen tasks such as multiple-choice Video Question Answering and Dense Captioning. These results suggest that COPRA provides an effective weight space generation mechanism for foundation models, enabling more scalable, adaptive, and context-aware video understanding.

1 Introduction

Video anomaly detection (VAD) seeks to identify and localize rare and unexpected events in long videos, often under significant variability in scene context, camera viewpoints, and anomaly types. Recent VAD approaches based on vision-language models (VLMs) Zanella et al. [2024], Zhang et al. [2024], Ye et al. [2025], Zhu et al. [2025b] are particularly promising, via leveraging rich semantic priors from large-scale pretraining and enabling interpretable predictions.

However, effectively adapting VLMs to VAD remains an actively evolving and largely underexplored challenge. As illustrated in Figure 1, existing VAD approaches Zanella et al. [2024], Zhang et al. [2024], Ye et al. [2025], Huang et al. [2025], Zhu et al. [2025b], Zou et al. [2026] typically learn a single, shared parameter space, despite the substantial heterogeneity across anomaly types, temporal dynamics, and scene contexts. During training, each heterogeneous sample induces a distinct gradient direction that may conflict with others, leading to parameters that converge to a compromise rather than being optimal for any single sample Tencent HY Team [2026]. Such static, shared adaptations often overfit to dataset-specific patterns encoded in prompts Ye et al. [2025] or model parameters Zhang et al. [2024], rather than capturing transferable notions of abnormality.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

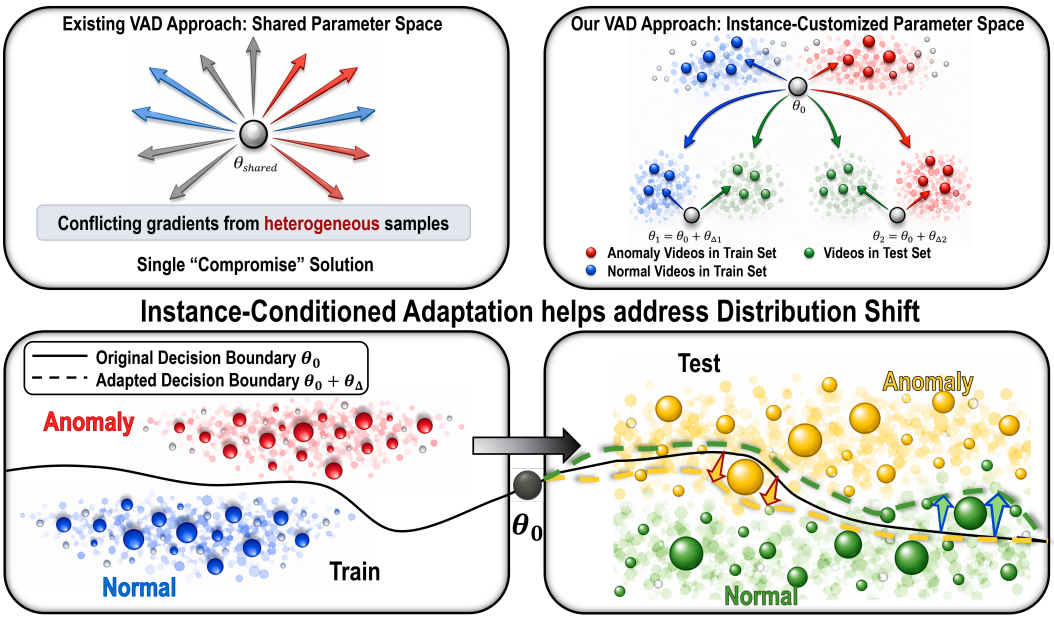

Figure 1: Illustration of the generalization gap in VAD caused by training–inference mismatch and train–test distribution shift. Existing methods rely on shared, static parameters, forcing diverse video patterns into a single compromise solution. As a result, *the decision boundary learned from the training dataset often fails to generalize to unseen test distributions, causing some samples to fall on the wrong side of the boundary*. We address this limitation through instance-conditioned parameter adaptation, where each video *dynamically pulls the decision landscape toward a more suitable boundary via input-specific parameter updates*, improving robustness and generalization.

This leads to a pronounced generalization gap, which we term the *shared parameter bottleneck*. For example, detecting a traffic accident Kim et al. [2025], physical assault Sultani et al. [2018], or suspicious retail behavior requires distinct visual cues, temporal dynamics, and semantic priors. A single shared adaptation must reconcile these heterogeneous demands, leading to degraded robustness and generalization. Moreover, prevailing VAD pipelines exhibit a fundamental training–inference mismatch: models are trained on sparsely sampled frames from long videos with weak video-level supervision, but deployed on densely sampled sliding windows for fine-grained localization. As a result, parameters learned from sparse inputs are applied to dense local decisions, degrading performance under variations in viewpoints, environments, motion patterns, and anomaly semantics.

In this paper, we advocate an instance-specific perspective on VAD Tencent HY Team [2026], Han et al. [2026]. Rather than learning a single shared parameter set, we infer model parameters on the fly for each video segment via instance-conditioned adaptation, where most parameters are shared and frozen, and only a small subset is dynamically customized to emphasize input-specific behaviors. This is motivated by the diversity of anomaly scenarios: some events (e.g., explosions, fires, collisions) can be identified from a few salient frames, while others (e.g., shoplifting, burglary, loitering) require modeling subtle motion patterns and temporal interactions. Consequently, reliable detection spans cues from coarse appearance to fine-grained temporal reasoning, making existing VAD approaches based on a shared-parameter paradigm inherently a compromise. A promising direction is to enable on-the-fly, input-conditioned adaptation that tailors model behavior to each video input.

Accordingly, we propose COPRA, a conditional parameter adaptation framework for VLM-based VAD that predicts input-specific parameter updates for each video, dynamically modulating a largely frozen VLM on the fly. In practice, COPRA produces instance-conditioned LoRA weights for customized per-input adaptation. As illustrated in Figure 1, COPRA addresses the limitation of existing static adaptation approaches, where *the decision boundary learned from the training dataset often fails to generalize to unseen test distributions*, by dynamically pulling the decision landscape toward a more suitable boundary for each input, thereby improving robustness and generalization under distribution shift. We further employ reinforcement learning fine-tuning to guide token-level parameter generation under video-level supervision via downstream, non-differentiable rewards,

without ground-truth parameter updates or reference reasoning trajectories, and explore flexible frame sampling strategies.

Our main contributions can be summarized as follows:

1. We identify a fundamental limitation of existing VAD approaches with the *shared parameter bottleneck*, where static, shared adaptations force heterogeneous requirements into a single compromised solution, and advocate a shift toward on-the-fly, input-conditioned adaptation.

2. We present COPRA, the first framework to formulate VAD as an instance-conditioned adaptation problem, where a VLM is dynamically specialized for each video segment.

3. We introduce a conditional parameter generation framework for VLMs that enables dynamic, per-input adaptation through generated LoRA weights. We further incorporate reinforcement learning fine-tuning under video-level supervision and investigate scenario-aware frame sampling strategies. Our framework mitigates training–inference mismatch and reduces cross-domain interference, improving robustness across diverse anomaly patterns.

4. We demonstrate strong performance and robust generalization across standard VAD benchmarks, consistently outperforming static adaptation baselines in both in-domain and cross-dataset settings. Moreover, the learned adaptation transfers effectively beyond VAD to downstream video understanding tasks.

2 Related Work

Traditional VAD. Fully supervised VAD methods rely on fine-grained temporal annotations for precise localization, but suffer from high labeling cost and limited scalability Acsintoae et al. [2022], Gao et al. [2025]. Weakly supervised approaches instead learn from video-level labels, typically via multiple instance learning (MIL) or contrastive objectives Sultani et al. [2018], Li et al. [2022b], Tian et al. [2021], Chen et al. [2023], Joo et al. [2023], Yang et al. [2024b], Wu et al. [2024b], Zhou et al. [2023], Wu et al. [2024a]. Unsupervised methods further relax supervision by modeling normality and detecting deviations using reconstruction or prediction-based objectives Gong et al. [2019], Park et al. [2020], Zaheer et al. [2022], Thakare et al. [2023], Geng et al. [2026].

VLMs for VAD. Recent works leverage VLMs for their strong cross-modal reasoning capabilities Pratt et al. [2023], Liu et al. [2023]. Existing approaches largely treat VLMs as fixed operators: some attach task-specific detection heads to frozen backbones Cai et al. [2025], others decouple captioning and reasoning via VLM–LLM pipelines Zanella et al. [2024], Yang et al. [2024a], while prompt-based methods (e.g., VERA Ye et al. [2025]) or post-hoc frame selection strategies Zhang et al. [2025, 2024] avoid updating model parameters.

Shared Parameter Bottleneck in VLM-based VAD. As previously observed, existing VLM-based VAD approaches adapt models through shared parameters, either by optimizing global prompts Ye et al. [2025] or fine-tuning model weights Zhang et al. [2024]. This introduces a fundamental limitation: model behavior is governed by training data statistics and a fixed operator, hindering generalization to novel videos. As a result, performance degrades on unseen data. Consistent with model editing studies Tencent HY Team [2026], Han et al. [2026], a single shared update is inherently driven toward a compromise, limiting its ability to capture diverse anomaly patterns.

Conditional Parameter Generation. Our approach closely relates to weight-space generation Han et al. [2026], where model parameters are synthesized via hypernetworks or generative models. Early hypernetworks Ha et al. [2017] generate weights from layer-conditioned embeddings and, in some settings, input-dependent signals. However, these methods were primarily developed for Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), where the generated weights are comparatively small; the adaptation mechanism is not designed for modern large vision-language models. Subsequent methods scale this idea via checkpoint reconstruction Peebles et al. [2022], Schürholt et al. [2021], Wang et al. [2024], but rely on pre-collected adapter banks. Recently, HY-WU Tencent HY Team [2026] generates conditional LoRA weights from downstream supervision, but is not designed for temporally grounded video reasoning.

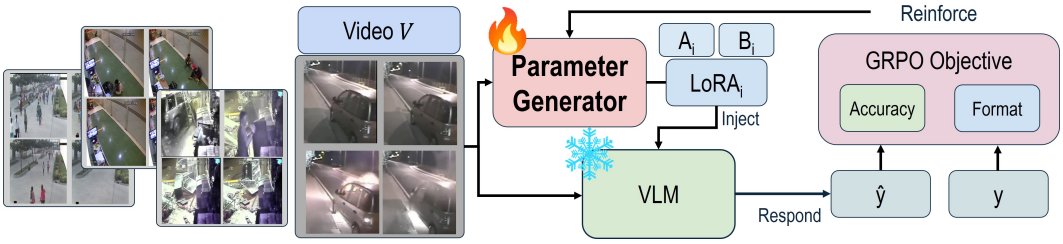

Figure 2: Overview of **COPRA**, an instance-conditioned parameter generation framework for VAD. The key advantage is a learned functional memory that maps each instance to tailored parameters, enabling improved robustness and cross-domain generalization compared to static adaptation methods.

3 The COPRA framework

In this section, we present **COPRA** (**Co**nditional **PaR**ameter **A**daptation), a framework that formulates VAD as an instance-conditioned adaptation problem. To our knowledge, this is the first approach to dynamically specialize VLM behavior for each input video in VAD tasks.

Problem Definition. Given a video $V = \{F_1, \ldots, F_n\}$, the goal is to predict frame-level anomaly labels $y_t \in \{0, 1\}$ for each frame F_t. In practice, only a video-level label $y \in \{0, 1\}$ is available, where $y = 1$ indicates the presence of at least one anomaly and $y = 0$ denotes a normal video. The objective is to infer fine-grained frame-level scores from this coarse video-level supervision.

Overview. Our COPRA framework adapts VLM parameters on a per-instance basis to overcome the limitations of static, shared adaptations in existing VAD methods, such as supervised fine-tuning (SFT), parameter-efficient tuning (e.g., adapters, LoRA), or prompt optimization. Formally, given a pretrained VLM $f(\cdot; \theta)$, static adaptation learns the global update $\Delta_{\theta_{\text{static}}}$ and produces predictions as

$$\hat{y} = f\big(V; \theta + \Delta_{\theta_{\text{static}}}\big). \tag{1}$$

As can be seen, $\Delta_{\theta_{\text{static}}}$ remains fixed across inputs after training—e.g., fixed guiding questions in VERA Ye et al. [2025] and fine-tuned LoRA parameters in Zhang et al. [2024]—creating a *shared parameter bottleneck* that leads to a pronounced generalization gap. Instead, the proposed framework generates instance-conditioned parameter updates:

$$\Delta_\theta(V) = g_{\varphi_2}\big(\upsilon_{\varphi_1}(V)\big), \quad \hat{y} = f\big(V; \theta + \Delta_\theta(V)\big). \tag{2}$$

Here, υ_{φ_1} and g_{φ_2} denote two sub-networks with parameters φ_1 and φ_2, respectively. The encoder υ_{φ_1} extracts a compact conditioning signal from the input video V, which is then fed into g_{φ_2} to produce an input-specific parameter update Δ. This formulation enables adaptive, instance-conditioned modulation of the baseline model and is closely connected to recent advances in weight-space generation Han et al. [2026], where model parameters are dynamically synthesized using hypernetworks or generative models. As illustrated in Figure 1, Δ_θ serves to adaptively shift the original decision boundary—initially suboptimal for a given instance—toward a more suitable, input-specific configuration, thereby achieving a better VAD performance.

After establishing the foundation, we present the COPRA framework, organized around three key questions: (1) how to design and implement the encoder and parameter generator, (2) how to train them effectively, and (3) how to enable flexible temporal adaptation via a sampling mechanism. The latter is designed to control the granularity at which parameter updates are generated and applied during inference, supporting video-level, segment-level, and hybrid adaptation.

Visual Input as Generation Condition. To obtain a compact yet representative video feature embedding vectors during training, we uniformly sample a fixed number of frames (e.g., $K = 8$). Starting from the first frame, given a video V with n frames, we compute a stride $S = \lfloor \frac{n}{K} \rfloor$ and construct the subset $\tilde{V} = \{F_1, F_{1+S}, F_{1+2S}, \ldots, F_{1+(K-1)S}\}$. The sampled frames $\tilde{V}$ are processed by υ_{φ_1}, instantiated as a frozen vision encoder (e.g., InternViT in InternVL2-8B Chen et al. [2025]), to produce tokenized visual representations. It allows us to reuse the VLM's feature extractor, as υ_{φ_1} is implemented as a submodule of f in our setting. These representations serve as the sole conditioning signal for g_{φ_2} in our framework, similar to the spirit of Zhang and Viteri [2025] and Tencent HY Team [2026] for guiding model behavior through targeted adaptation.

Conditional Parameter Generation. Our parameter generator g_{φ_2} is a Transformer that maps the concatenated sequence of per-frame visual tokens, i.e., $\upsilon_{\varphi_1}(\tilde{V})$, into tokenized, instance-conditioned LoRA parameters for the language backbone. We initialize the generated LoRA parameters with a learnable global latent prior p_{global}, which injects dataset-level adaptation patterns into the generated update: $\Delta_\theta \leftarrow \Delta_\theta + \mathrm{p}_{global}$. Importantly, we do not seek to eliminate shared structure; adding p_{global} is analogous to combining a shared latent component with an input-conditioned residual transformation. Specifically, the generator g_{φ_2} applies factorized self-attention Tencent HY Team [2026] over parameter tokens, together with cross-attention to visual features. Intra-layer attention captures dependencies within each backbone block, while inter-layer attention models structural correlations across depth. The resulting representations are then fed into lightweight MLP-based projection heads to generate the LoRA A and B matrices.

We inject these generated LoRA matrices as residual updates into the backbone, covering major projection components such as query–key–value projections, gating modules, and output transformations. We adopt the standard LoRA initialization scheme Hu et al. [2022], where B is initialized to zero and A is randomly initialized. This design preserves the pretrained model behavior at the start of training (i.e., a VAD-agnostic initialization) and enables a smooth transition toward instance-conditioned adaptation as learning progresses. The details of the architecture can be found in the appendix. To train the generator, we adopt Group Relative Policy Optimization (GRPO) Shao et al. [2024].

End-to-end Reinforcement Learning for COPRA. We train COPRA end-to-end by learning a "policy" g_{φ_2} that maps visual conditioning signals to instance-specific parameter updates. Specifically, let L denote the number of backbone layers, and let s_l denote the number of $r \times d$ parameter tokens required to reconstruct the low-rank adaptation matrices for layer l. The generator is defined as $g_\varphi : V \rightarrow \Delta_\theta \in \mathbb{R}^{L \times s_l \times r \times d}$, where Δ_θ denotes the generated LoRA parameters.

These parameters are injected into the frozen backbone language model to form $\pi_{\theta+\Delta_\theta}$ (e.g., InternLM2 or QwenLM), yielding an instance-adapted policy. Updating only these generated modules while keeping the remaining parameters fixed results in the adapted VLM $f(\cdot; \theta + \Delta_\theta)$. The gradients propagate through the generated LoRA weights back to g_{φ_2}. Details of the reward design, loss functions, and additional implementation choices are provided in the appendix; these components are intentionally kept simple and closely follow the standard GRPO formulation Shao et al. [2024].

Flexible Sampling for Temporal Adaptation. At inference time, we adapt the model to the VAD setting, where the goal is frame-level anomaly scoring. Instead of processing the entire video at once, we partition it into temporally overlapping 10-second segments, each centered at every 16th frame following prior practice Zanella et al. [2024], Ye et al. [2025]. For each segment, $K = 8$ frames are sampled and fed to the parameter generator to produce instance-conditioned LoRA updates, which adapt the VLM backbone for segment-level prediction. The resulting scores from overlapping segments are aggregated to the frame level and further refined using the VERA-style visual retrieval, temporal smoothing, and weighting pipeline Ye et al. [2025] to obtain continuous anomaly scores.

Since the sliding-window inference scheme may require frequent re-adaptation, we further investigate flexible temporal adaptation strategies. Specifically, we construct conditioning inputs from broader temporal contexts (e.g., full videos or multi-segment chunks of varying durations) and reuse the resulting parameters to score local segments, enabling flexible temporal adaptation through context-aware parameter sharing. Details are presented in the experimental section (Table 6).

4 Experiments

We evaluate COPRA in terms of performance, robustness, and generalization. Specifically, we address three questions: (i) whether instance-conditioned adaptation outperforms static adaptation approaches; (ii) whether this advantage persists under parameter-efficient settings compared to LoRA-based GRPO baselines; and (iii) whether the learned adaptation generalizes to unseen tasks and distribution shifts both within and beyond VAD.

4.1 Experimental Settings

In-Domain Datasets. We train our method on two standard VAD benchmarks: UCF-Crime Sultani et al. [2018] and XD-Violence Wu et al. [2020]. Following common weakly supervised VAD practice, we cast evaluation as frame-level binary anomaly classification (normal vs. abnormal).

Table 1: In-domain comparison with prior VAD methods on UCF-Crime and XD-Violence. Among explainable approaches, COPRA consistently improves over its frozen baseline and outperforms strong baselines such as VERA Ye et al. [2025]. On UCF-Crime, it raises InternVL2-8B from 84.31 to 87.14 AUC, and on XD-Violence from 74.32 to 76.52 AP.

(a) UCF-Crime		**(b) XD-Violence**	
Method	AUC (%)	Method	AP (%)
Prior Unexplainable VAD Methods		*Prior Unexplainable VAD Methods*	
RTFM Tian et al. [2021]	84.30	RTFM Tian et al. [2021]	77.81
SSRL Li et al. [2022a]	87.43	S3R Wu et al. [2022]	80.26
MSL Li et al. [2022b]	85.62	MSL Li et al. [2022b]	78.58
CLIP-TSA Joo et al. [2023]	87.58	CLIP-TSA Joo et al. [2023]	82.19
MGFN Chen et al. [2023]	86.98	MGFN Chen et al. [2023]	80.11
OVVAD Wu et al. [2024a]	86.40	OVVAD Wu et al. [2024a]	66.53
Prior Explainable VAD Methods		*Prior Explainable VAD Methods*	
LAVAD Zanella et al. [2024]	80.28	LAVAD Zanella et al. [2024]	62.01
Holmes-VAD Zhang et al. [2024]	84.61	Holmes-VAD Zhang et al. [2024]	84.96
VADor Lv and Sun [2024]	85.90	ZS ImageBind-V Zanella et al. [2024]	25.36
LLAVA-1.5 Liu et al. [2024]	72.84	LLAVA-1.5 Liu et al. [2024]	50.26
VERA (InternVL2-8B) Ye et al. [2025]	86.55	VERA (InternVL2-8B) Ye et al. [2025]	70.54
Frozen Baseline		*Frozen Baseline*	
Qwen2-VL-7B-Instruct (Frozen)	81.31	Qwen2-VL-7B-Instruct (Frozen)	64.36
InternVL2-8B (Frozen)	84.31	InternVL2-8B (Frozen)	74.32
COPRA		*COPRA*	
Qwen2-VL-7B-Instruct	**86.75**	Qwen2-VL-7B-Instruct	**73.96**
InternVL2-8B	**87.14**	InternVL2-8B	**76.52**

Table 2: COPRA consistently outperforms frozen VLM baselines across models and in-domain datasets. For fair comparison, all models use VERA's three-stage inference—(Step 1) initial scoring, (Step 2) retrieval refinement, and (Step 3) temporal smoothing/weighting—and we compare identical backbones with and without COPRA. COPRA improves AUC/AP at every stage for Qwen2-VL-7B-Instruct and InternVL2-8B, demonstrating gains beyond post-processing alone.

Model	Stage	**UCF-Crime**		**XD-Violence**	
		Frozen	**+ COPRA**	Frozen	**+ COPRA**
Qwen2-VL-7B-Instruct	Step 1	59.36	**69.73**	54.01	**63.09**
	Step 2	77.91	**84.69**	62.01	**72.60**
	Step 3	81.31	**86.75**	64.36	**73.96**
InternVL2-8B	Step 1	64.95	**71.49**	61.53	**67.24**
	Step 2	81.74	**85.21**	72.83	**75.80**
	Step 3	84.31	**87.14**	74.32	**76.52**

Cross-Domain Datasets. To evaluate whether COPRA generalizes beyond surveillance VAD, we conduct zero-shot evaluation, without fine-tuning, on two out-of-domain benchmarks with distinct objectives. VRU-Accident Kim et al. [2025], built from DADA Fang et al. [2019], DoTA Yao et al. [2023], and additional curated CAP and MANU videos, focuses on traffic safety scenarios involving vulnerable road users and supports both multiple-choice video question answering and dense captioning. We further evaluate on HIVAU-70K Zhang et al. [2025], an open-ended anomaly-centric video understanding benchmark. Notably, COPRA is trained only on UCF-Crime and directly evaluated on these datasets without additional fine-tuning, enabling a rigorous cross-domain generalization assessment.

Metrics. We follow standard weakly supervised VAD protocols, reporting frame-level AUC (Area Under the Curve, %) on UCF-Crime and Average Precision (AP, %) on XD-Violence, consistent with common practice in the VAD literature. For VRU-Accident multiple-choice video QA, we report classification accuracy (%). For dense captioning on VRU-Accident, we use semantic metrics

Table 3: Ablation on UCF-Crime comparing COPRA with static adaptation under GRPO training. The frozen baseline uses the standard VLM pipeline; static adaptation applies shared LoRA with GRPO; COPRA adds instance-conditioned LoRA to the frozen baseline under the same GRPO setting. COPRA consistently outperforms both baselines across all stages, indicating that the gains come from adaptive conditioning rather than GRPO alone.

Inference Stage	Frozen Baseline	GRPO Static Adaptation	**Frozen Baseline + COPRA Dynamic Adaptation**
Initial (Step 1)	59.36	61.34	**69.73**
Initial + Retrieval (Step 2)	77.91	79.51	**84.69**
Initial + Retrieval + Smoothing + Weighting (Step 3)	81.31	83.00	**86.75**

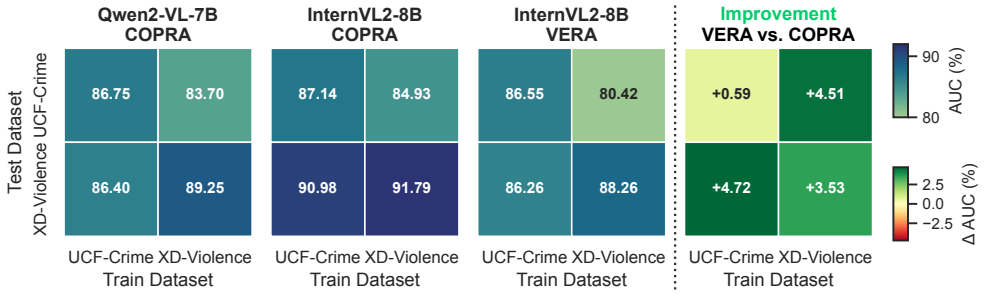

Figure 3: AUC (%) performance for cross-dataset generalization across train–test dataset pairs. From left to right: COPRA with Qwen2-VL-7B-Instruct, COPRA with InternVL2-8B, VERA with InternVL2-8B, and the AUC improvement of COPRA over VERA Ye et al. [2025].

Table 4: Cross-domain dense captioning results on VRU-Accident Kim et al. [2025]. Prec., Rec., and F1 denote ROUGE-1/2/L scores. COPRA improves semantic metrics (SPICE, METEOR, COMET) and boosts ROUGE recall and F1 for higher-order structures (bigrams and LCS), indicating better recovery of accident-relevant content and stronger alignment with ground-truth descriptions.

Model \ Metric	Semantic Metrics			ROUGE-1			ROUGE-2			ROUGE-L		
	SPICE	METEOR	COMET	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
InternVL2-8B	0.139	0.249	0.694	**0.398**	0.431	**0.410**	**0.091**	0.098	0.093	**0.201**	0.220	0.208
+ COPRA	**0.149**	**0.273**	**0.701**	0.359	**0.489**	**0.410**	0.090	**0.123**	**0.103**	0.186	**0.254**	**0.212**
Improvement (↑)	+0.010	+0.024	+0.007	-0.039	+0.058	+0.000	-0.001	+0.025	+0.010	-0.015	+0.034	+0.004

(SPICE Anderson et al. [2016], METEOR Lavie and Agarwal [2007], COMET Rei et al. [2020]) and ROUGE Lin [2004] for lexical overlap. For HIVAU-70K, we report BLEU Papineni et al. [2002], CIDEr Vedantam et al. [2014], METEOR, and ROUGE at the clip-, event-, and video-level.

4.2 Comparison to State-of-the-Art Methods

Table 1 shows that COPRA achieves improved performance over prior explainable methods, consistently across architectures. Notably, it approaches strong non-explainable baselines while employing a fundamentally different instance-conditioned adaptation mechanism and providing interpretable reasoning. Holmes-VAD is included among prior explainable VAD methods due to its strong performance. However, because it relies on fine-grained temporal annotations, it is not directly comparable under the weakly supervised VAD setting. Table 2 further shows that these gains generalize across backbones (Qwen2-VL-7B-Instruct vs InternVL2-8B), datasets (UCF-Crime and XD-Violence), with consistent improvements at each inference step when COPRA is leveraged.

4.3 Ablation Studies

Instance-Specific Parameters Outperform Static Adaptation. To further validate COPRA, we evaluate three configurations: (1) a *Frozen Baseline* with no adaptation; (2) *Static Adaptation with GRPO*, which applies shared, dataset-specific LoRA parameters (Eq. 1) trained with GRPO; and (3) *Frozen Baseline with COPRA*, which uses instance-conditioned LoRA parameters (Eq. 2) trained

Table 5: Cross-domain multiple-choice video question answering results on VRU-Accident Kim et al. [2025]. Compared with the frozen InternVL2-8B baseline, COPRA improves overall accuracy by 1.35 points, with the largest gain of 6.67 points on DoTA.

Model \ Dataset	DADA	CAP	DoTA	MANU	**Overall**
	1,338	1,722	600	2,340	6,000
InternVL2-8B	48.58	**52.50**	44.83	47.86	49.05
InternVL2-8B + COPRA	**49.70**	50.41	**51.50**	**50.51**	**50.40**
improvement (↑)	1.12	-2.09	6.67	2.65	1.35

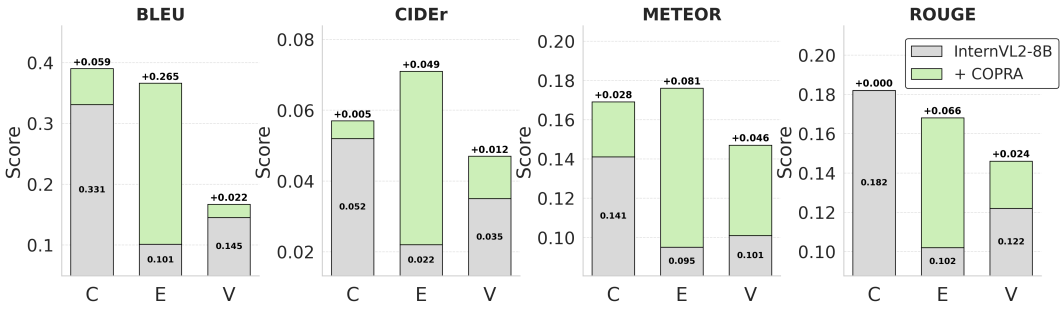

Figure 4: Cross-domain evaluation of natural language explanation quality on HIVAU-70K Zhang et al. [2025]. C, E, and V denote clip-, event-, and video-level evaluation, respectively. Compared with the InternVL2-8B baseline, COPRA improves most semantic and lexical overlap metrics across all temporal levels, including BLEU, CIDEr, METEOR, and ROUGE, demonstrating strong cross-domain generalization for natural language explanation generation.

Table 6: Effect of temporal granularity on UCF-Crime Test AUC for Qwen2-VL-7B. The 120s chunk setting achieves the best performance, improving over video-level generation by 0.65 AUC.

Metric \ Granularity	10s Segment	30s Chunk	60s Chunk	**120s Chunk**	240s Chunk	Video Full
AUC (%)	86.51	86.63	86.48	**87.40**	86.56	86.75
Improvement over video-level generation	-0.24	-0.12	-0.27	**+0.65**	-0.19	–

with GRPO. As shown in Table 3, *Frozen Baseline with COPRA* delivers substantial gains, improving AUC by 5.4 points over the frozen baseline and by 3.75 points over static adaptation with GRPO. While static adaptation applies a shared update across all inputs, leading to a compromised solution, COPRA performs on-the-fly, input-conditioned adaptation that better handles conflicting patterns such as anomaly types and scene contexts.

Instance-Specific Parameters Generalize In-Domain Datasets. A key question is whether improved in-domain performance reflects over-specialization. To test this, we train COPRA on UCF-Crime and evaluate on both UCF-Crime and XD-Violence, and conversely train on XD-Violence and evaluate on both datasets. As shown in Figure 3, COPRA consistently improves over baselines and outperforms VERA across all train–test combinations, indicating strong cross-dataset generalization.

Instance-Specific Parameters Generalize Cross-Domain Datasets. We further evaluate cross-domain generalization by directly applying the COPRA model trained on UCF-Crime without additional fine-tuning. On VRU-Accident, COPRA improves both dense captioning quality (Table 4) and zero-shot multiple-choice VideoQA accuracy (Table 5), suggesting strong transfer to safety-critical video understanding under substantial domain shift. Notably, the improvements indicate that instance-conditioned adaptation captures transferable abnormality cues beyond dataset-specific patterns learned during training. On HIVAU-70K, COPRA also improves natural-language explanations (Figure 4), yielding consistent gains across BLEU, CIDEr, METEOR, and ROUGE. These results further demonstrate that COPRA generalizes beyond anomaly scoring to broader cross-domain video understanding and reasoning tasks.

Flexible Sampling for Temporal Adaptation in COPRA. Existing VAD approaches typically use heavily overlapping sliding windows (e.g., 10 seconds) with the same static parameters applied to

Table 7: Effect of the learnable global latent in InternVL2-8B + COPRA on UCF-Crime after one epoch of training. Both COPRA variants improve over the frozen baseline across all inference stages, while including the learnable global latent during training achieves the strongest performance.

Inference Stage	Frozen Baseline	COPRA No Global Latent	**COPRA** Learnable Global Latent
Initial (Step 1)	64.95	**71.79**	71.49
Initial + Retrieval (Step 2)	81.74	84.43	**85.21**
Initial + Retrieval + Smoothing + Weighting (Step 3)	84.31	86.39	**87.14**

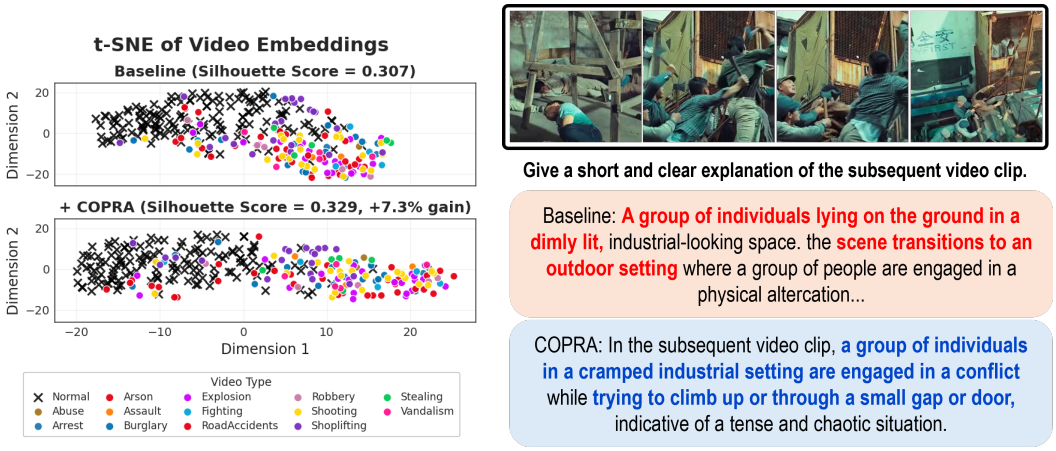

Figure 5: Qualitative analysis of InternVL2-8B + COPRA representations and responses. Left: Under shared t-SNE axes, COPRA produces more discriminative embeddings, with normal and anomalous videos more clearly separated; we quantify this separation using the silhouette score, where higher values indicate better clustering quality. Right: COPRA generates anomaly-relevant descriptions with behaviorally salient cues, whereas the baseline mainly produces generic scene descriptions.

all windows. In contrast, COPRA adapts to different temporal stages by generating chunk-specific parameters. Specifically, we divide videos into non-overlapping 30-, 60-, or 120-second chunks, sample $K = 8$ frames per chunk, and generate low-rank parameters once per chunk. Each 10-second segment is then scored using the parameters of its corresponding chunk, amortizing parameter generation while preserving broader temporal context. As shown in Table 6, COPRA remains robust across temporal granularities, with the best achieved from the 120-second chunks.

Role of the Learnable Global Latent. Table 7 shows that COPRA consistently outperforms the frozen baseline both with and without the learnable global latent p_{global} during training. Removing p_{global} eliminates the dataset-level prior, forcing the generator to rely solely on video-conditioned features. Including p_{global} improves post-processed results by approximately 0.75% AUC.

4.4 Qualitative Results and Case Studies

Figure 5 compares the representation space (left) and generated descriptions of the baseline model and COPRA (right). COPRA separates normal and anomalous videos more clearly, including categories such as shoplifting that remain close to normal videos in the baseline space. We quantify class separation using the silhouette score Rousseeuw [1987], computed on the two-dimensional t-SNE embeddings. We further analyze the generated parameters in Figures 9 and 10. Figure 8 provides a complementary case-study from UCF-Crime. This difference is reflected in the generated descriptions: the baseline produces a fragmented narrative that abruptly shifts between unrelated scenes, whereas COPRA generates a more coherent anomaly-focused description by consistently capturing the industrial setting, ongoing conflict, and tense attempts to escape through a narrow opening. We provide details in the appendix.

Table 8: Inference efficiency breakdown for InternVL2-8B with COPRA. We report wall-clock time and peak GPU memory for the main inference components: frozen baseline response generation, parameter generation, and adapted response generation. All measurements are averaged over 30 scoring windows with a maximum response length of 256 tokens.

Inference Component	Time (second)	VRAM (MB)
Baseline Response Generation	1.05892 ± 0.26664	2315.36048 ± 1.35751
Parameter Generation	0.11824 ± 0.08367	832.08968 ± 1.45848
COPRA Response Generation	1.17716 ± 0.33232	3147.45016 ± 1.99249

Table 9: End-to-end inference efficiency under different parameter-generation granularities. Coarser generation allows a single generated parameter update to be reused across multiple local scoring segments, substantially reducing per-segment latency and increasing throughput.

Generation Granularity	End-to-End Latency per Segment (ms)	Segment Throughput (segments/s)	Effective Source-Video FPS	Videos/hour
Segment	2372.90	0.421	6.73	6.32
60-second Chunk	456.79	2.189	34.95	32.82
Video	444.62	2.249	35.91	33.72

4.5 Inference Efficiency

COPRA generates approximately one million LoRA parameters for an 8B-parameter backbone, which introduces additional inference overhead. To make this cost transparent, we report three separate timing components: the response time of the frozen InternVL2-8B baseline, the time required by the parameter generator to produce instance-conditioned LoRA parameters, and the response time of the augmented backbone after applying the generated parameters. We separate these components because parameter generation does not necessarily occur for every model response. Depending on the inference granularity, the generated parameters may be reused across an entire video, a long temporal chunk, or multiple 10-second segments, allowing us to amortize the parameter generation cost over an entire video. We report memory usage in Table 8 using the same decomposition, including the baseline backbone, the parameter generator, and the augmented backbone, to provide a complete view of the computational cost introduced by instance-conditioned parameter generation. Comparing training costs under the same configuration, we find that static-LoRA requires approximately 6 hours on one A100 GPU for ≈4k videos, while training COPRA requires approximately 7 hours.

As shown in Table 9, generating parameters independently for every segment incurs the highest cost. Reusing generated parameters over longer temporal contexts substantially amortizes this cost. Accordingly, effective throughput increases to 34.95 FPS for 60-second chunk-level generation.

5 Conclusion

We identify several fundamental limitations of existing VAD approaches, including the *shared parameter bottleneck*, where static shared adaptations force heterogeneous requirements into a single compromised solution, as well as the mismatch between training and inference in both data distribution and model configuration. We introduced COPRA, an instance-conditioned parameter generation framework that adapted a frozen VLM to each video by generating low-rank weights trained with reinforcement learning under weak video-level supervision. COPRA improved robustness to diverse scenes, anomaly types, and temporal dynamics. Across both in-domain and cross-domain evaluations, COPRA consistently outperformed frozen and static adaptation baselines and could potentially serve as an effective weight space generation mechanism for foundation models, enabling scalable, adaptive, and context-aware video understanding. Limitations include limited interpretability of generated LoRA weights, lack of theoretical grounding, and additional computational overhead introduced by repeated response sampling during GRPO training, all of which warrant future investigation.

References

Andra Acsintoae, Andrei Florescu, Mariana-Iuliana Georgescu, Tudor Mare, Paul Sumedrea, Radu Tudor Ionescu, Fahad Shahbaz Khan, and Mubarak Shah. Ubnormal: New benchmark for supervised open-set video anomaly detection. In *CVPR*, pages 20143–20153, 2022.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic propositional image caption evaluation. In *ECCV*, pages 382–398, Cham, 2016. ISBN 978-3-319-46454-1.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Zhaolin Cai, Fan Li, Ziwei Zheng, and Yanjun Qin. HiProbe-VAD: Video anomaly detection via hidden states probing in tuning-free multimodal LLMs. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, page 592–601, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720352. doi: 10.1145/3746027.3755575. URL https://doi.org/10.1145/3746027.3755575.

Yingxian Chen, Zhengzhe Liu, Baoheng Zhang, Wilton Fok, Xiaojuan Qi, and Yik-Chung Wu. MGFN: magnitude-contrastive glance-and-focus network for weakly-supervised video anomaly detection. In *AAAI*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i1.25112. URL https://doi.org/10.1609/aaai.v37i1.25112.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025. URL https://arxiv.org/abs/2412.05271.

Jianwu Fang, Dingxin Yan, Jiahuan Qiao, Jianru Xue, He Wang, and Sen Li. Dada-2000: Can driving accident be predicted by driver attention? Analyzed by a benchmark. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 4303–4309. IEEE, 2019.

Shibo Gao, Peipei Yang, Haiyang Guo, Yangyang Liu, Yi Chen, Shuai Li, Han Zhu, Jian Xu, Xu-Yao Zhang, and Linlin Huang. The evolution of video anomaly detection: A unified framework from dnn to mllm, 2025. URL https://arxiv.org/abs/2507.21649.

Yuang Geng, Junkai Zhou, Kang Yang, Pan He, Zhuoyang Zhou, Jose C. Principe, Joel Harley, and Ivan Ruchkin. MLE-UVAD: Minimal latent entropy autoencoder for fully unsupervised video anomaly detection, 2026. URL https://arxiv.org/abs/2603.23868.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton Van Den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *ICCV*, pages 1705–1714, 2019. doi: 10.1109/ICCV.2019.00179.

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *ICLR*, 2017. URL https://openreview.net/forum?id=rkpACe1lx.

Xiaolong Han, Zehong Wang, Bo Zhao, Binchi Zhang, Jundong Li, Damian Borth, Rose Yu, Haggai Maron, Yanfang Ye, Lu Yin, and Ferrante Neri. A survey of weight space learning: Understanding, representation, and generation, 2026. URL https://arxiv.org/abs/2603.10090.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chao Huang, Yushu Shi, Jie Wen, Wei Wang, Yong Xu, and Xiaochun Cao. Ex-VAD: Explainable fine-grained video anomaly detection based on visual-language models. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pages 25750–25761. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/huang25ad.html.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.181. URL https://aclanthology.org/2024.findings-emnlp.181/.

Hyekang Kevin Joo, Khoa Vo, Kashu Yamazaki, and Ngan Le. CLIP-TSA: Clip-assisted temporal self-attention for weakly-supervised video anomaly detection. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 3230–3234, 2023. doi: 10.1109/ICIP49359.2023.10222289.

Younggun Kim, Ahmed S. Abdelrahman, and Mohamed Abdel-Aty. VRU-Accident: A vision-language benchmark for video question answering and dense captioning for accident scene understanding. In *ICCV*, pages 761–771, October 2025.

Alon Lavie and Abhaya Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/W07-0734/.

Guoqiu Li, Guanxiong Cai, Xingyu Zeng, and Rui Zhao. Scale-aware spatio-temporal relation learning for video anomaly detection. In *ECCV*, pages 333–350. Springer, 2022a.

S. Li, Fang Liu, and Licheng Jiao. Self-training multi-sequence learning with transformer for weakly supervised video anomaly detection. In *AAAI*, 2022b. URL https://api.semanticscholar.org/CorpusID:248982052.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013/.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pages 26296–26306, June 2024.

Hui Lv and Qianru Sun. Video anomaly detection and explanation via large language models, 2024. URL https://arxiv.org/abs/2401.05702.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *CVPR*, pages 14372–14381, 2020.

William Peebles, Ilija Radosavovic, Tim Brooks, Alexei A. Efros, and Jitendra Malik. Learning to learn with generative models of neural network checkpoints, 2022. URL https://arxiv.org/abs/2209.12892.

Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *ICCV*, pages 15691–15701, 2023.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.213. URL https://aclanthology.org/2020.emnlp-main.213/.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

Konstantin Schürholt, Dimche Kostadinov, and Damian Borth. Self-supervised representation learning on neural network weights for model characteristic prediction. In *NeurIPS*, 2021. URL https://openreview.net/forum?id=F1D8buayXQT.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. VLM-R1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *CVPR*, pages 6479–6488, 2018.

Tencent HY Team. HY-WU (part i): An extensible functional neural memory framework and an instantiation in text-guided image editing, 2026. URL https://arxiv.org/abs/2603.07236.

Kamalakar Vijay Thakare, Yash Raghuwanshi, Debi Prosad Dogra, Heeseung Choi, and Ig-Jae Kim. DyAnNet: A Scene Dynamicity Guided Self-Trained Video Anomaly Detection Network . In *WACV*, pages 5530–5539, Los Alamitos, CA, USA, January 2023. IEEE Computer Society. doi: 10.1109/WACV56688.2023.00550. URL https://doi.ieeecomputersociety.org/10.1109/WACV56688.2023.00550.

Yu Tian, Guansong Pang, Yuanhong Chen, Rajvinder Singh, Johan W. Verjans, and Gustavo Carneiro. Weakly-supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning . In *ICCV*, pages 4955–4966, Los Alamitos, CA, USA, October 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00493. URL https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00493.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2014. URL https://api.semanticscholar.org/CorpusID:9026666.

Kai Wang, Dongwen Tang, Boya Zeng, Yida Yin, Zhaopan Xu, Yukun Zhou, Zelin Zang, Trevor Darrell, Zhuang Liu, and Yang You. Neural network diffusion, 2024. URL https://arxiv.org/abs/2402.13144.

Jhih-Ciang Wu, He-Yen Hsieh, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Self-supervised sparse representation for video anomaly detection. In *ECCV*, page 729–745, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-19777-2. doi: 10.1007/978-3-031-19778-9_42. URL https://doi.org/10.1007/978-3-031-19778-9_42.

Peng Wu, Jing Liu, Yujia Shi, Yujia Sun, Fangtao Shao, Zhaoyang Wu, and Zhiwei Yang. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *ECCV*, 2020.

Peng Wu, Xuerong Zhou, Guansong Pang, Yujia Sun, Jing Liu, Peng Wang, and Yanning Zhang. Open-Vocabulary Video Anomaly Detection . In *CVPR*, pages 18297–18307, Los Alamitos, CA, USA, June 2024a. IEEE Computer Society. doi: 10.1109/CVPR52733.2024.01732. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01732.

Peng Wu, Xuerong Zhou, Guansong Pang, Lingru Zhou, Qingsen Yan, Peng Wang, and Yanning Zhang. VadCLIP: adapting vision-language models for weakly supervised video anomaly detection. In *AAAI*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024b. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i6.28423. URL https://doi.org/10.1609/aaai.v38i6.28423.

Yuchen Yang, Kwonjoon Lee, Behzad Dariush, Yinzhi Cao, and Shao-Yuan Lo. Follow the rules: Reasoning for video anomaly detection with large language models. In *ECCV*, page 304–322, Berlin, Heidelberg, 2024a. Springer-Verlag. ISBN 978-3-031-73003-0. doi: 10.1007/978-3-031-73004-7_18. URL https://doi.org/10.1007/978-3-031-73004-7_18.

Zhiwei Yang, Jing Liu, and Peng Wu. Text prompt with normality guidance for weakly supervised video anomaly detection. In *CVPR*, pages 18899–18908, 2024b. doi: 10.1109/CVPR52733.2024.01788.

Yu Yao, Xizi Wang, Mingze Xu, Zelin Pu, Yuchen Wang, Ella Atkins, and David J. Crandall. DoTA: Unsupervised Detection of Traffic Anomaly in Driving Videos . *IEEE T-PAMI*, 45(01): 444–459, January 2023. ISSN 1939-3539. doi: 10.1109/TPAMI.2022.3150763. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3150763.

Muchao Ye, Weiyang Liu, and Pan He. VERA: Explainable video anomaly detection via verbalized learning of vision-language models. In *CVPR*, pages 8679–8688, June 2025.

M. Zaigham Zaheer, Arif Mahmood, M. Haris Khan, Mattia Segu, Fisher Yu, and Seung-Ik Lee. Generative cooperative learning for unsupervised video anomaly detection. In *CVPR*, pages 14724–14734, 2022. doi: 10.1109/CVPR52688.2022.01433.

Luca Zanella, Willi Menapace, Massimiliano Mancini, Yiming Wang, and Elisa Ricci. Harnessing large language models for training-free video anomaly detection. In *CVPR*, pages 18527–18536, 2024.

Huaxin Zhang, Xiaohao Xu, Xiang Wang, Jialong Zuo, Chuchu Han, Xiaonan Huang, Changxin Gao, Yuehuan Wang, and Nong Sang. Holmes-VAD: Towards unbiased and explainable video anomaly detection via multi-modal LLM. *arXiv preprint arXiv:2406.12235*, 2024.

Huaxin Zhang, Xiaohao Xu, Xiang Wang, Jialong Zuo, Xiaonan Huang, Changxin Gao, Shanjun Zhang, Li Yu, and Nong Sang. Holmes-VAU: Towards long-term video anomaly understanding at any granularity. In *CVPR*, pages 13843–13853, 2025. doi: 10.1109/CVPR52734.2025.01292.

Jason Zhang and Scott W Viteri. Uncovering latent chain of thought vectors in large language models. In *Workshop on Neural Network Weights as a New Data Modality*, 2025. URL https://openreview.net/forum?id=ICuIdJzBPm.

Hang Zhou, Junqing Yu, and Wei Yang. Dual memory units with uncertainty regulation for weakly supervised video anomaly detection. In *AAAI*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i3.25489. URL https://doi.org/10.1609/aaai.v37i3.25489.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025a. URL https://arxiv.org/abs/2504.10479.

Liyun Zhu, Qixiang Chen, Xi Shen, and Xiaodong Cun. VAU-R1: Advancing video anomaly understanding via reinforcement fine-tuning. *arXiv preprint arXiv:2505.23504*, 2025b.

Shu Zou, Xinyu Tian, Lukas Wesemann, Fabian Waschkowski, Zhaoyuan Yang, and Jing Zhang. Unlocking vision-language models for video anomaly detection via fine-grained prompting. In *WACV*, pages 4223–4233, 2026.

A Technical appendices and supplementary material

A.1 Temporal Localization under Weak Video-Level Supervision

A central challenge in weakly supervised VAD is that video-level labels provide no direct information about the temporal location of anomalous events; anomalous videos may contain substantial normal content. We therefore analyze COPRA's frame-level predictions within anomalous videos in UCF-Crime to assess whether it can recover meaningful temporal distinctions despite this coarse supervision. Among ground-truth normal frames in anomalous videos, only 46.6% are classified as normal, while 53.4% are assigned an anomalous prediction. Despite this ambiguity, COPRA identifies 72.8% of the anomalous frames in anomalous videos. Figure 6 illustrates the trade-off between anomalous frame recall and false alarms within normal videos. Figure 7 complements these aggregate results with an example from XD-Violence, where COPRA's anomaly scoring aligns better with ground truth annotations than the pre-trained VLM.

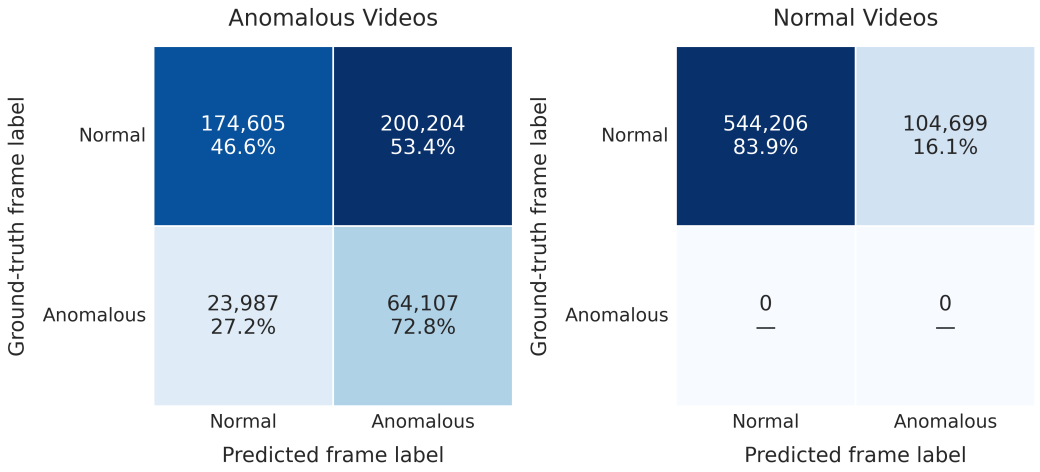

Figure 6: Frame-level confusion matrices separated by video-level label. Percentages are normalized within each ground-truth frame class. In anomalous videos, COPRA detects 72.8% of anomalous frames, but misclassifies 53.4% of the normal frames as anomalous, illustrating the localization ambiguity introduced by coarse video-level supervision.

A.2 Three-Stage Post-Processing for Inference

For frame-level VAD evaluation, we follow the three-stage inference pipeline of VERA Ye et al. [2025]. We retain all associated post-processing hyperparameters used by VERA. Thus, Step 1 isolates the raw segment-level predictions produced by the VLM, while Steps 2 and 3 show the effect of applying the same established post-processing pipeline to those predictions. Unless otherwise specified, COPRA uses video-level parameter generation with segment-level anomaly scoring.

Step 1: Raw segment-level anomaly scoring. Following VERA Ye et al. [2025], we define overlapping 10-second segments centered at every $d = 16$ frames and uniformly sample eight frames from each segment. Let V_u denote the sampled visual input for the u-th segment. Using the video-level parameters $\Delta_\theta(V)$, the raw segment anomaly score $\tilde{y}_u$ is given by Equation 2. Repeating this process for all H segments produces $\tilde{\mathbf{Y}} = [\tilde{y}_1, \ldots, \tilde{y}_H]$. Step 1 provides the clearest measurement of the effect of frozen, static, or instance-conditioned parameter adaptation.

Step 2: Retrieval-based refinement. A prediction based on a short segment may omit relevant scene context. Following VERA, we use a fixed visual encoder (e.g., ImageBind) to retrieve the K_{ret} most visually similar segments from the same video. Their Step 1 anomaly scores are linearly combined using softmax-normalized cosine- similarity weights.

Step 3: Temporal smoothing and frame-level weighting. We convolve the retrieval-refined segment scores with the same Gaussian kernel used by VERA to suppress high-frequency fluctuations. The smoothed score associated with each segment is then assigned to its corresponding frames to obtain a frame-level anomaly-score sequence. Finally, we apply VERA's Gaussian position-weighting function to model the video-level progression of an anomalous event.

A.3 Implementation

We instantiate a lightweight parameter generator that produces LoRA updates conditioned on visual inputs. While we share the overall design across models, we adapt dimensionality and layer mappings to match each backbone. For InternVL2-8B Chen et al. [2025], we generate parameters of rank $r = 8$, operating over all transformer layers and targeting both attention and feed-forward modules. We set the generator depth to half that of the backbone (16 layers for a 32-layer language model). COPRA uses an 87.37M-parameter generator while keeping the underlying vision-language backbone frozen. For Qwen2-VL-7B-Instruct Bai et al. [2025], we use an analogous configuration. Across different backbones, we standardize the internal generator dimension (512).

We adopt the default VLM-R1 Shen et al. [2025] training configuration, retaining all hyperparameters unless adjustments are required to meet memory constraints. Models are trained for one epoch over the full training dataset. We use a learning rate of 1×10^{-6} with linear decay, gradient clipping with a maximum norm of 1, no gradient accumulation, a KL regularization weight of $\beta = 0.1$, and an importance sampling ratio clipping coefficient of $\epsilon = 0.2$. Optimization is performed using AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 1 \times 10^{-8}$. All experiments are conducted on a single NVIDIA A100-SXM4 GPU (80GB). Training and evaluation take approximately three days on UCF-Crime and XD-Violence.

A.4 Prompt

```
You are the model.
** Model Description: **
You are designed to do binary classification. The input is a sequence of video
     frames for identifying whether there is an anomaly in the video; you
    need to output the class label, i.e., an integer in the set {0, 1}. 0
    represents normal video, and 1 represents abnormal video. Please answer
    the prompt questions.

** Prompt Questions: **
Answer the following questions based on what you see from the video frames and
     provide an explanation in one sentence.
Based on the analysis above, please conclude your answer to 'Is there any
    anomaly in the video?' in 'Yes, there is an anomaly' or 'No, there is no
    anomaly'.

Please give your output strictly in the following format:
<think>[Add your reasoning here]</think>
<answer>[ONLY the integer class label; make necessary assumptions if needed]</
    answer>

Please ONLY reply according to this format, don't give me any other words.
```

Listing 1: System Prompt for VAD Classification

Table 10: Cross-dataset generalization from UCF-Crime to XD-Violence using InternVL2-8B. The model is trained on UCF-Crime and evaluated on the unseen XD-Violence dataset. Compared with the frozen baseline, COPRA improves AUC at every inference stage.

Inference Stage	Frozen Baseline	**COPRA**	Improvement (↑)
Initial (Step 1)	71.78	**75.90**	+4.12
Initial + Retrieval (Step 2)	89.53	**90.20**	+0.67
Initial + Retrieval + Smoothing + Weighting (Step 3)	90.63	**90.98**	+0.35

A.5 Reward Design and GRPO Objective

Our video-level rewards are:

$$\text{Accuracy}(s_0, a_0, \ldots, a_O) = \begin{cases} 1 & \text{output == ground truth} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$\text{Format}(s_0, a_0, \ldots, a_O) = \begin{cases} 1 & \text{output matches <think>...</think><answer>...</answer>} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

We deliberately use binary rewards for their simplicity and verifiability — a simple choice in RL settings to isolate the effectiveness of our approach. Our loss is given by: Shao et al. [2024]

$$L_{\text{GRPO}} = -\mathbb{E}_{\{(s_0, a_0^g, \ldots, a_O^g)\}_{g=1}^{G} \sim \pi_{\theta+\Delta_\theta}(\cdot|s_0)} \left[\frac{1}{O} \sum_{i=1}^{O} \min \left(\frac{\pi_{\theta+\Delta_\theta}(a_i^g \mid s_0, a_0^g, \ldots, a_{i-1}^g)}{\pi_{\theta_{\text{old}}}(a_i^g \mid s_0, a_0^g, \ldots, a_{i-1}^g)} \hat{A}^g, \text{clip} \left(\frac{\pi_{\theta+\Delta_\theta}(a_i^g \mid s_0, a_0^g, \ldots, a_{i-1}^g)}{\pi_{\theta_{\text{old}}}(a_i^g \mid s_0, a_0^g, \ldots, a_{i-1}^g)}, 1-\epsilon, 1+\epsilon \right) \hat{A}^g \right) \right]$$

$$L_{\text{total}} = L_{GRPO} + \beta \mathrm{D}_{\text{KL}}(\pi_{\theta+\Delta_\theta} \,||\, \pi_{\text{REF}})$$

Here, s_0 denotes the initial state consisting of the system prompt and sampled video frames, and O is the number of output tokens per generation $g \in [1, G]$. The system prompt is provided in Listing 1. π_{REF} is the frozen language model, before adaptation.

The advantage $\hat{A}^g$ is computed via group-wise normalization:

$$R(s_0, a_0, \ldots, a_O) = \text{Accuracy}(s_0, a_0, \ldots, a_O) + \text{Format}(s_0, a_0, \ldots, a_O)$$

$$\hat{A}^g = \frac{R(s_0, a_0^g, \ldots, a_O^g) - \text{mean}(\{R(s_0, a_0^{g'}, \ldots, a_O^{g'})\}_{g'=1}^{G})}{\text{std}(\{R(s_0, a_0^{g'}, \ldots, a_O^{g'})\}_{g'=1}^{G}) + 10^{-4}}$$

Importantly, the G sampled generations are all produced from a single adapted policy π_θ using temperature-based decoding, identical to standard token sampling in large language models. That is, diversity arises from stochastic decoding under fixed adapted parameters for a given video instance, rather than from generating each response with a different parameter adaptation for that same video. This is sufficient for meaningful advantage estimation because stochastic output sequences $\{a_0^g, \ldots, a_O^g\}_{g=1}^{G} \sim \pi_\theta(\cdot|s_0)$ under the same adapted parameters produce differing predictions and rewards. The resulting reinforcement signal has a natural structure: when a majority of output sequences are correct, the generator receives net positive reinforcement for the parameters it produced for that instance; when all G output sequences are uniformly correct, the normalized advantages collapse to zero and the generator receives no update; and when a majority are incorrect, the generator receives net negative reinforcement, discouraging the generator from producing similar parameters when conditioned on visually similar instances. This asymmetry ensures that the parameter generator is consistently pushed toward producing adapters that yield reliable correct predictions, without requiring explicit supervision over what the parameters themselves should look like.

A.6 Cross-Dataset Generalization Analysis for InternVL2-8B

Although we report Average Precision (AP) as the primary metric in the main paper, following the standard evaluation protocol for video anomaly detection, we analyze AUC to assess whether COPRA improves threshold-independent ranking quality across datasets. Table 10 evaluates InternVL2-8B trained on UCF-Crime and tested directly on the unseen XD-Violence benchmark, showing that

Table 11: Ablation on XD-Violence comparing COPRA with static adaptation under GRPO training for InternVL2-8B. COPRA outperforms across all three post-processing stages, demonstrating that the improvement is attributable to instance-conditioned adaptation rather than GRPO training.

Inference Stage	Frozen Baseline	GRPO Static Adaptation	**Frozen Baseline + COPRA Dynamic Adaptation**
Initial (Step 1)	61.53	65.97	**67.24**
Initial + Retrieval (Step 2)	72.83	74.90	**75.80**
Initial + Retrieval + Smoothing + Weighting (Step 3)	74.32	75.94	**76.52**

Table 12: UCF-Crime test AUC using InternVL3-8B Zhu et al. [2025a] after one epoch of training. Compared with the frozen baseline, COPRA improves AUC at every inference stage and also exceeds the InternVL2-8B COPRA result on UCF-Crime, which achieves 87.14% AUC.

Inference Stage	Frozen Baseline	**COPRA**	Improvement (↑)
Initial (Step 1)	74.77	**76.24**	+1.47
Initial + Retrieval (Step 2)	84.90	**85.84**	+0.94
Initial + Retrieval + Smoothing + Weighting (Step 3)	86.62	**87.33**	+0.71

COPRA improves over the frozen baseline at every inference stage. These gains indicate that instance-conditioned parameter generation improves cross-domain transfer before retrieval-based refinement, and that COPRA preserves the advantage under the full inference pipeline.

A.7 Additional Static-Dynamic Adaptation Ablation

To test whether the advantage of instance-conditioned adaptation over shared parameters extends beyond Qwen2-VL-7B-Instruct on UCF-Crime (Table 3), we conduct a matched ablation using InternVL2-8B on XD-Violence. As shown in Table 11, static LoRA improves over the frozen baseline at every inference stage. COPRA further improves over static LoRA at all three stages, with gains of 1.27, 0.90, and 0.58 AP at Steps 1 to 3, respectively. These results indicate that COPRA's advantage over shared adaptation extends across model backbones, datasets, and evaluation metrics.

When does Instance-Conditioned Adaptation Help? Instance-conditioned adaptation is most useful when anomaly cues vary substantially across categories, viewpoints, or temporal contexts, whereas a static adapter must compromise across heterogeneous inputs. Consistent with this hypothesis, COPRA improves raw Step 1 performance over static LoRA by 8.39 AUC points on UCF-Crime but by 1.27 AP points on XD-Violence. Although these gains are not directly comparable across metrics, the smaller margin on XD-Violence, which contains many visually salient online and movie-based anomalies, is consistent with reduced benefit from input-specific adaptation in more visually prototypical settings. We present this as an empirical interpretation rather than a theoretical result.

Table 13: XD-Violence test AP using InternVL3-8B after one epoch of training. Compared with the frozen baseline, COPRA improves AP at every inference stage and also exceeds the InternVL2-8B COPRA result on XD-Violence at Step 1, which achieves 67.24% AP.

Inference Stage	Frozen Baseline	**COPRA**	Improvement (↑)
Initial (Step 1)	67.08	**69.82**	+2.74
Initial + Retrieval (Step 2)	72.86	**74.81**	+1.95
Initial + Retrieval + Smoothing + Weighting (Step 3)	73.96	**75.70**	+1.73

A.8 Scaling to InternVL3-8B

We further evaluate COPRA with InternVL3-8B Zhu et al. [2025a] after one epoch of training to test whether instance-conditioned adaptation remains effective for a stronger backbone. The InternVL3-8B parameter generator follows the same COPRA design as InternVL2-8B, including rank-$r = 8$ LoRA updates, $\alpha = 16$, an internal generator dimension of 512, and bfloat16 inference, but adapts its dimensionality and target-module mapping to the evolved InternVL3 architecture.

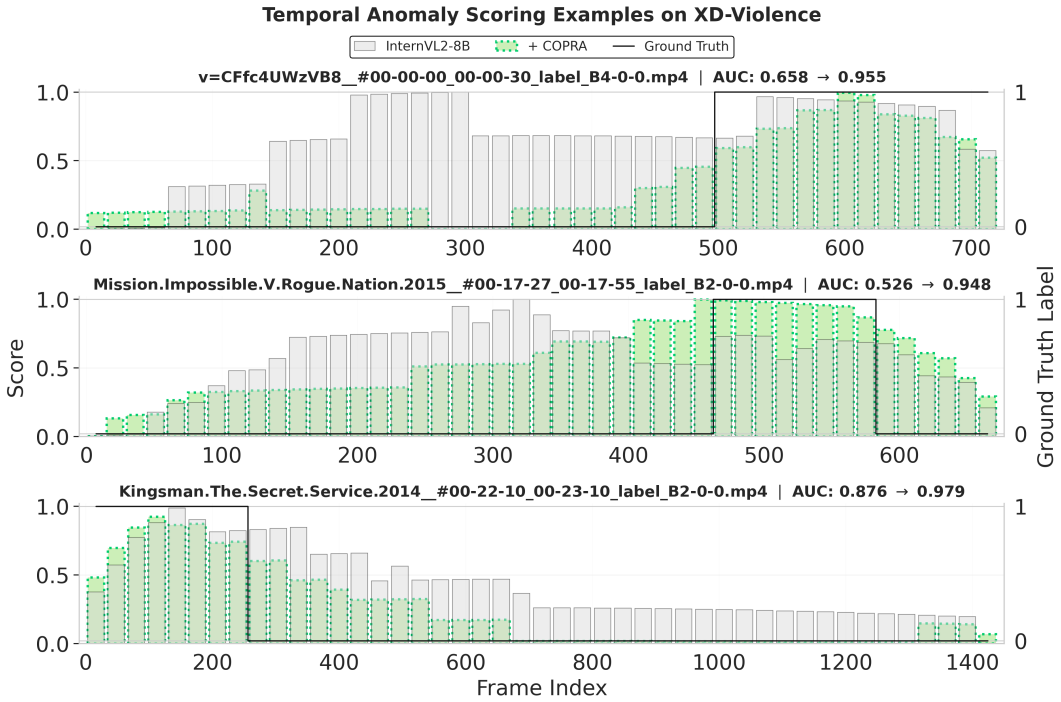

Figure 7: Qualitative temporal anomaly scoring example on XD-Violence. The grey bars show the InternVL2-8B baseline, the green bars show COPRA, and the black dashed signal indicates the ground-truth anomaly interval. Although the baseline assigns high anomaly scores to normal frames before the annotated event and drops during much of the abnormal interval, COPRA better aligns its peak response with the ground-truth abnormal segment. These examples illustrate how instance-conditioned parameter generation improves temporal anomaly localization.

Specifically, InternVL3-8B uses a 28-layer language backbone with hidden size 3584 and unfused attention projections, so COPRA targets the query, key, value, output, gate, up, and down projections separately. In contrast, InternVL2-8B uses a 32-layer backbone with hidden size 4096 and fused query-key-value attention, requiring different LoRA output shapes. Thus, the differences in generator configuration reflect architectural changes between InternVL2 and InternVL3 rather than changes to the COPRA method itself.

On UCF-Crime, COPRA improves over the frozen baseline at every inference stage, increasing AUC from 74.77 to 76.24 at Step 1 and from 86.62 to 87.33 after retrieval, smoothing, and weighting (Table 12). On XD-Violence, COPRA similarly improves AP across all stages, with the final post-processed score increasing from 73.96 to 75.70 (Table 13). These results suggest that COPRA's gains are not specific to InternVL2-8B, but persist when scaling to InternVL3-8B.

A.9 Details of t-SNE Embeddings and Silhouette Scores

For the embedding visualization, we extract last-layer hidden states after the model processes the task prompt and visual input. As a lightweight diagnostic, we use the final input-token representation as a sequence-level embedding, yielding one embedding per input video–prompt pair. This choice is motivated by prior work that uses final-token hidden states as text representations Jiang et al. [2024], although more specialized embedding methods may produce stronger representations. Under COPRA, the same input is processed with instance-conditioned parameters, which can shift the resulting summary embedding relative to the frozen baseline. We then project these high-dimensional embeddings into two dimensions using t-SNE van der Maaten and Hinton [2008], compute silhouette scores in the projected t-SNE space, and visualize the resulting representation space in Figure 5 (left).

For each video V_i, we compute the silhouette coefficient Rousseeuw [1987] on the two-dimensional t-SNE embedding: $s(V_i) = \frac{b(V_i)-a(V_i)}{\max\{a(V_i),b(V_i)\}}$, where $a(V_i)$ is the mean distance to videos from the

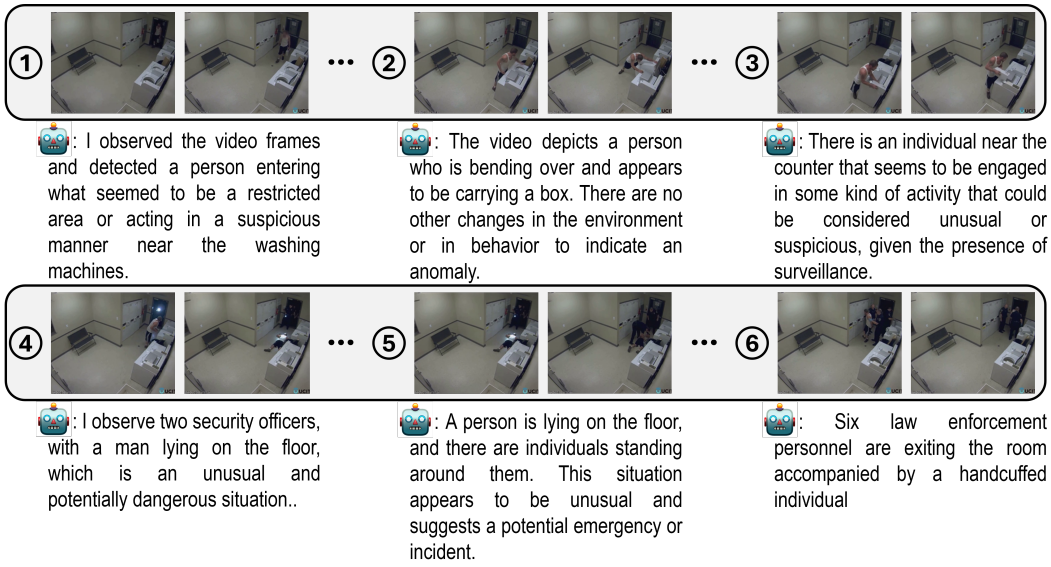

Figure 8: Illustration of COPRA's segment-level reasoning process on an arrest video from UCF-Crime. Each numbered group shows representative frames sampled from a temporal segment, where the visual context evolves from an initially empty room to suspicious human activity, physical confrontation, and the arrival of additional people. For each segment, the VLM produces a corresponding reasoning response, allowing the anomaly decision to be grounded in localized visual evidence and yielding a more reliable final anomaly score.

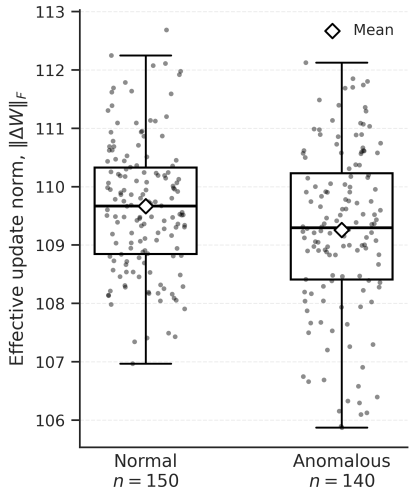

Figure 9: Distribution of the Frobenius norms of the generated effective LoRA updates, $\|\Delta W\|_F$, for normal and anomalous UCF-Crime videos. The mean update norm is approximately 0.4% higher for normal videos than for anomalous videos.

same class and $b(V_i)$ is the mean distance to the nearest competing class. We report the average silhouette coefficient over all videos.

A.10 Analyzing Generated Parameters

The generated LoRA parameters are highly stable, but not identical, across neighboring temporal windows. For each UCF-Crime test video, we select two consecutive central frames from the evaluation sampling sequence, generate the LoRA parameters from eight uniformly sampled frames in each window, and compute the cosine similarity between the resulting effective updates. Across

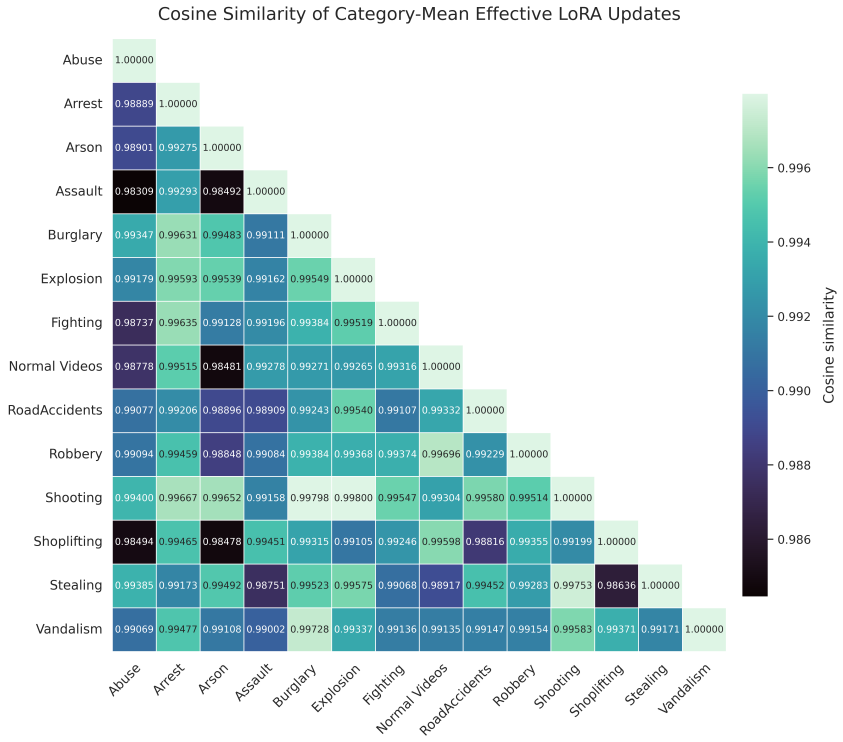

Figure 10: Pairwise cosine similarity between category-mean effective LoRA updates on UCF-Crime. Normal-video updates are similar to robbery and shoplifting, but less aligned with arson and abuse.

all videos, the mean cosine similarity between neighboring-window updates was 0.9998 ± 0.00017. This indicates that the generated parameters change smoothly as the temporal window advances.

We visualized the LoRA weights in Figure 9 by tracking the Frobenius Norm of the generated weights (which is proportional to the RMS of the matrix weights parameterizing the generated LoRA) for anomalous and normal videos. We find that the update norm for anomalous videos is roughly 0.5% smaller than the norm for normal videos. Furthermore, when tracking the cosine similarity between updates produced for videos belonging to 14 different categories in UCF-Crime in Figure 10, we observe some patterns in similar parameter updates for anomaly types that may be considered difficult to distinguish. For instance, the most similar LoRA parameters for Normal Videos are those generated for Robbery, Arrest and Shoplifting.

A.11 Role of Shared Parameters: Category-Level Effect of $\mathrm{p}_{\mathrm{global}}$

COPRA combines shared and instance-specific adaptation. The learnable global latent acts as a shared prior that captures transferable, dataset-level knowledge, while the conditional parameter generator uses each video's visual features to produce instance-conditioned parameter updates. The ablation results in Table 7 show that combining the global latent and conditional parameter generation yields the strongest results. These findings support our central argument: shared knowledge remains valuable, but it should be complemented by input-conditioned adaptation.

We compare per-category frame-classification accuracy with and without $\mathrm{p}_{\mathrm{global}}$ in Figure 11. It improves Abuse, Arson, Assault, Burglary, and Shoplifting, but reduces Explosion, RoadAccidents, and Vandalism. Notably, Shoplifting improves from 48.2% to 51.5% (+3.3 points), suggesting that the global prior can help with subtle anomalies. Overall, the global latent acts as a complementary dataset-level prior rather than a universally beneficial component, while instance-conditioned visual features remain essential for category-specific adaptation.

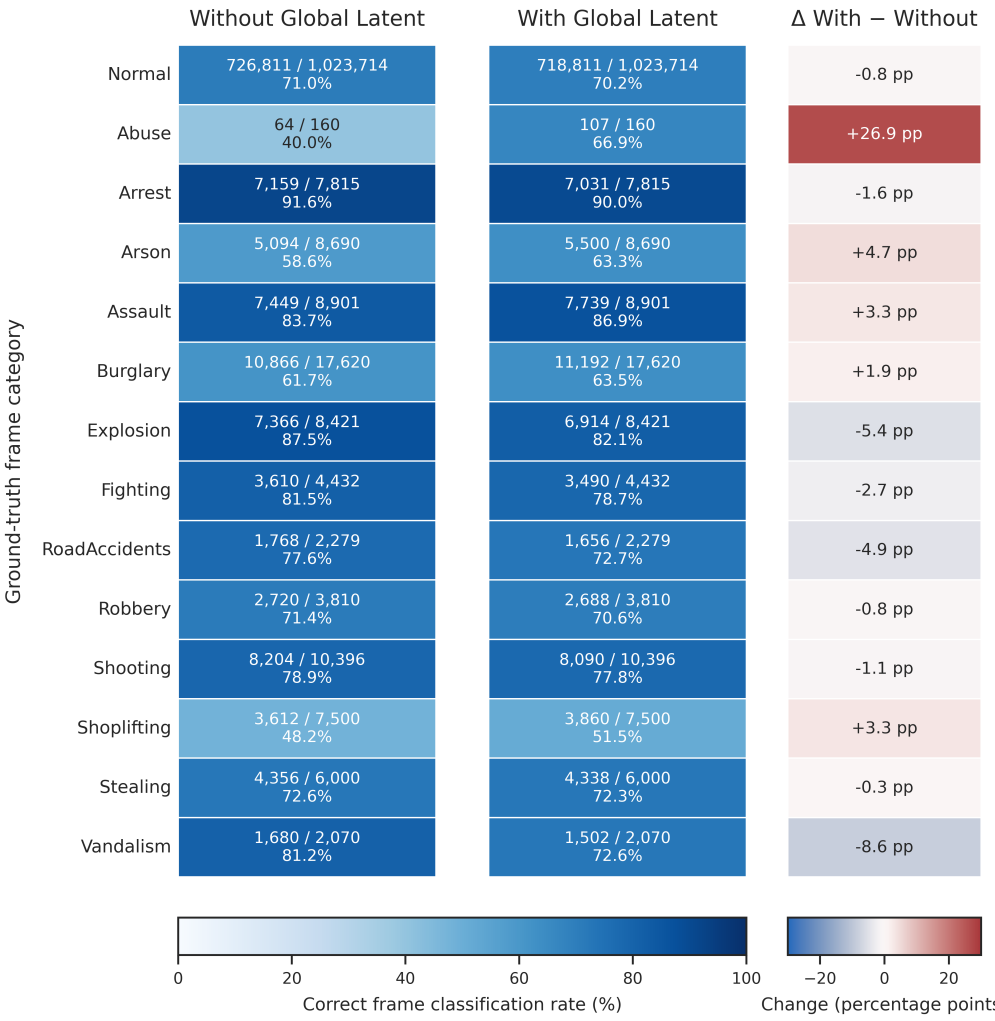

Figure 11: Category-level frame-classification accuracy on UCF-Crime with and without the learnable global latent. Each cell reports the number of correctly classified frames, the total number of frames in the category, and the resulting accuracy. The largest decrease occurs for Vandalism (-8.6 points).

A.12 Parameter Generator Architecture

We provide the parameter-generator architecture in detail for reproducibility. For InternVL2-8B, the conditioning features from the frozen vision encoder have dimension 4096 and are projected through a linear layer to the generator hidden dimension $d_{\text{model}} = 512$. The parameter generator contains 16 Transformer blocks, i.e., half the depth of the 32-layer InternVL2 language backbone, and contains 87.37M trainable parameters, while the underlying VLM remains frozen.

Rather than directly regressing flattened LoRA matrices, the generator operates on structured parameter tokens indexed by backbone layer, LoRA matrix type, and low-rank dimension. Each Transformer block applies factorized self-attention over this structure: one attention operation captures dependencies across backbone layers, while a second captures dependencies among parameter tokens within each layer. The resulting parameter tokens then cross-attend to the visual conditioning features, making the generated update input dependent. Each block uses RMSNorm, residual connections, and a SiLU feed-forward network, with positional structure encoded through learned layer/parameter embeddings and rotary positional embeddings.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's central claims: COPRA addresses the shared-parameter bottleneck and training–inference mismatch through instance-conditioned parameter generation for VLM-based VAD. These claims are reflected in the method in Section 3 and supported by the in-domain, cross-dataset, transfer, and ablation results reported in Sections 4.2 and 4.3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in Section 5 (Conclusion), including the restricted evaluation scope, the need for deeper interpretability analysis of generated parameters, and the additional cost of RL training.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present theoretical results or formal proofs. Instead, it introduces a methodological framework in Section 3 and evaluates it empirically in Section 4, so this checklist item is not applicable.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the main reproducibility details in Sections 3 (The COPRA Framework) and 4.1 (Experimental Settings), including the parameter-generation formulation, sampling strategy, datasets, metrics, backbone models, LoRA rank, generator configuration, training schedule, optimization settings, and inference pipeline. Additional prompt, reward, GRPO objective, and implementation details are provided in Appendix A.2, Appendix A.3, Appendix A.4, and Appendix A.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide open access to the implementation and reproduction instructions in an anonymous GitHub repository under the COPRA project (https://anonymous.4open.science/r/COPRA-C00F). The paper uses publicly available benchmark datasets, and the key experimental details needed to reproduce the main results are described in Sections 3 and 4.1 and Appendix A.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the experimental setting in Section 4.1, including the datasets and evaluation metrics. We specify backbone models, inference stages, LoRA rank, generator configuration, optimizer, learning rate, KL coefficient, clipping coefficient, training duration, and hardware. Additional implementation details, including the prompt, reward design, GRPO objective, and temporal adaptation settings, are provided in Section 4.3, Appendix A.3, Appendix A.4, and Appendix A.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars because repeated full training and dense video-level evaluation runs are computationally expensive for large VLM-based VAD, as discussed in Appendix A.3 (Implementation). Instead, we assess robustness through consistent improvements across datasets, model backbones, inference stages, cross-dataset transfer, and ablation settings in Sections 4.2 (Comparison to State-of-the-Art Methods) and 4.3 (Ablation Studies).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the main compute resources in Appendix A.3 (Implementation), including that experiments were conducted on a single NVIDIA A100-SXM4 GPU with 80GB memory, with training and evaluation taking approximately three days on UCF-Crime and XD-Violence.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The research conforms to the NeurIPS Code of Ethics and uses established public benchmark datasets for VAD and video understanding.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: Since we evaluate COPRA as a foundational research framework for public VAD benchmarks rather than as a deployed surveillance application, the paper does not provide a full broader-impact analysis of deployment risks, guardrails, or misuse scenarios.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This work does not release a new high-risk pretrained foundation model, image generator, or scraped dataset. The planned release is an adapter-based research implementation and reproduction code for experiments on established public benchmarks, so this item is not applicable.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper credits the original creators of the external datasets, models, and codebases used in the experiments, including UCF-Crime, XD-Violence, VRU-Accident, HIVAU-70K, InternVL2-8B, Qwen2-VL-7B-Instruct, VLM-R1, HY-WU, and VERA. The accompanying code release and supplemental material will include the relevant license and terms-of-use information for all reused assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets introduced by the paper are the COPRA implementation and reproduction scripts, which are documented and released through the anonymous GitHub repository under the COPRA project (`https://anonymous.4open.science/r/COPRA-C00F`). The paper provides the core methodological, training, evaluation, and limitation details in Sections 3–5 and Appendix A, while the release includes usage and reproduction instructions alongside the code.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or new research with human subjects. We conduct all experiments using existing benchmark datasets and pretrained vision-language models, so this item is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or new research with human subjects. All experiments use existing public benchmark datasets and pretrained vision-language models, so IRB approval or equivalent human-subjects review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper explicitly studies VLM-based video anomaly detection, where the language-model backbone is a core component of the method. Sections 3 and 4 describe how COPRA generates instance-conditioned LoRA parameters for the VLM language backbone, and we provide the prompt used for anomaly classification in Appendix A.4. We describe the GRPO-based training objective in Appendix A.5.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.