# TOWARDS FASTER ANOMALY DETECTION AND REASONING VIA KV CACHE COMPRESSION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Industrial Anomaly Detection (IAD) has become crucial for anomaly perception in complex industrial scenarios. Recently, Multimodal Large Language Models (MLLMs) have demonstrated strong global visual understanding and zero-shot cross-domain reasoning capabilities, leading to their growing application in IAD tasks. However, existing methods often overlook the high inference cost of MLLMs and the substantial key-value (KV) cache redundancy caused by high-resolution inputs. In this work, we investigate the attention distribution patterns of MLLMs in IAD and observe that models predominantly focus on normal regions, while anomalous regions receive limited effective attention, potentially resulting in incomplete anomaly perception. To address this issue, we propose AnomalyKV, a novel anomaly-aware KV cache compression framework for efficient inference. Our method performs fine-grained KV cache budget allocation, enabling instance-level dynamic compression across different samples. Extensive experiments demonstrate that AnomalyKV significantly improves inference efficiency while maintaining comparable performance across multiple mainstream MLLMs. Notably, AnomalyKV achieves 2.57× real-time acceleration in full-scenario tasks, outperforming existing mainstream KV cache compression methods with negligible performance degradation.

## 1 INTRODUCTION

Visual anomaly detection is an important research direction in computer vision, with applications spanning remote sensing anomaly detection (Du et al., 2021; Khatoon et al., 2025; Li et al., 2024c; Zhao et al., 2015; Račetin & Krtalić, 2021; Guo et al., 2016), industrial defect inspection (Bergmann et al., 2020; Cao et al., 2023; Chen et al., 2022; Huang et al., 2022; Mou et al., 2023; Reiss & Hoshen, 2023; Roth et al., 2022; Xie et al., 2023; You et al., 2022), and medical image diagnosis (Bao et al., 2024; Fernando et al., 2021; Han et al., 2021; Huang et al., 2024; Wolleb et al., 2022; Zhang et al., 2024; Zhao et al., 2021). As a key technology in automated manufacturing pipelines, Industrial Anomaly Detection (IAD) aims to identify defects in real time to mitigate production risks. Traditional IAD methods (Liu et al., 2024b; Xie et al., 2024) generally rely on threshold-based strategies and modeling the feature distributions of normal samples. However, complex real-world conditions often make high-quality normal data difficult to obtain, while conventional methods struggle to localize unseen defects and provide interpretable reasoning regarding defect causes and consequences.

Recent Multimodal Large Language Models (MLLMs) (Chen et al., 2024; Dai et al., 2023; Li et al., 2024a; 2023; 2024b; 2022; Liu et al., 2023; 2024a; Zhu et al., 2024) have demonstrated strong visual understanding and reasoning capabilities across challenging multimodal tasks. Consequently, several studies (Li et al., 2025; Xu et al., 2025; Zhao et al., 2025; Li et al., 2026; Guan et al., 2025; Chao et al., 2025; Miao et al., 2025; Liao et al., 2026) have introduced MLLMs into anomaly understanding tasks by integrating domain knowledge and specialized modules. Despite their promising performance, existing methods primarily emphasize anomaly reasoning accuracy while overlooking inference efficiency. In practice, however, inference latency and memory overhead are equally critical, especially when processing high-resolution images and large-scale datasets. We further observe that mainstream MLLMs, such as Gemma4-E-4B-IT, incur substantial time costs under large-sample inference settings.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

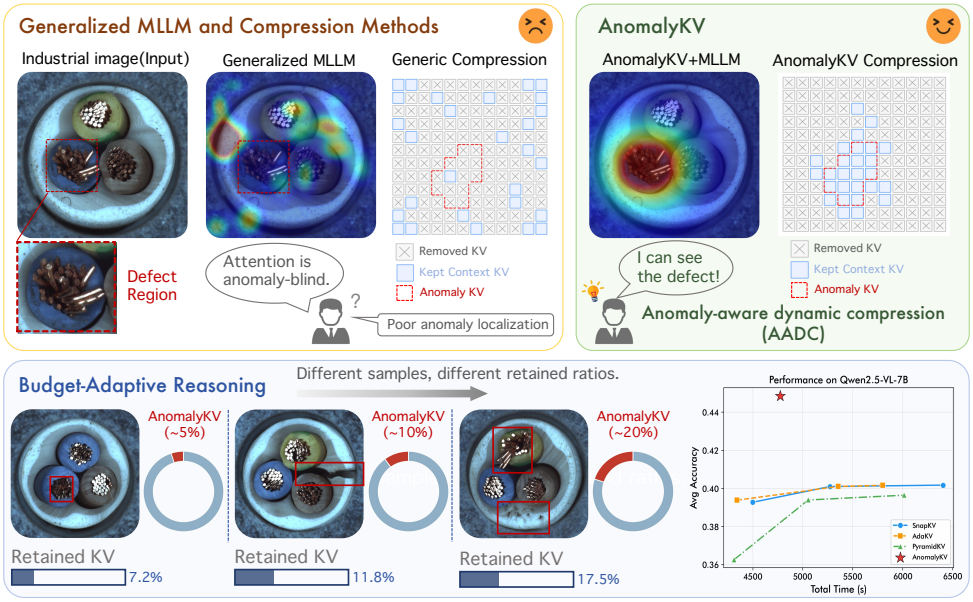



Figure 1: **Anomaly-Aware KV Cache Compression for Anomaly Reasoning.** Generic MLLMs and existing KV cache compression methods often fail to focus on defect-relevant KV entries. To address this, AnomalyKV combines coarse anomaly localization with anomaly-aware dynamic compression (AADC), retaining important KV pairs while pruning redundant context. By dynamically allocating KV budgets according to anomaly severity, AnomalyKV improves anomaly reasoning accuracy and inference efficiency under limited KV budgets.

To address these limitations, we first investigate the attention distribution patterns of mainstream MLLMs on benchmarks containing precise defect annotations. Our observations reveal that models predominantly focus on normal regions, while only limited attention is allocated to anomalous areas, potentially leading to incomplete anomaly perception. Based on this insight, we propose AnomalyKV, the first anomaly-aware KV cache compression framework for efficient anomaly reasoning. Specifically, we employ a zero-shot, training-free anomaly detection method for coarse-grained defect localization, followed by an importance-aware KV cache compression strategy to prune redundant information while preserving critical anomaly-related representations. This design significantly reduces inference latency and mitigates the high computational overhead of MLLMs.

However, KV cache compression methods designed for text and general vision tasks are difficult to directly extend to anomaly understanding, mainly due to the lack of reliable importance assessment for key-value pairs within anomalous regions. To overcome this issue, we introduce Anomaly-Aware Dynamic Compression (AADC), which enables fine-grained KV cache budget allocation for individual samples across large-scale multi-scenario IAD datasets. By dynamically solving instance-level compression ratios, AnomalyKV achieves stable inference acceleration with nearly lossless performance while maintaining explicit anomaly awareness.

Extensive experiments demonstrate the effectiveness of AnomalyKV in both anomaly perception and inference efficiency. Evaluations on the latest multimodal IAD benchmark, AnomalyCoT (Cheng et al., 2025), using mainstream MLLMs including Qwen2.5-VL-7B-Instruct (Bai et al., 2025) and Gemma4-E-4B-IT show that AnomalyKV preserves performance using only 8.1% KV cache on Qwen2.5-VL-7B-Instruct while significantly improving inference efficiency. Moreover, AnomalyKV consistently outperforms existing KV cache compression methods in anomaly reasoning accuracy and runtime efficiency under constrained cache budgets, highlighting the potential of anomaly-aware KV compression for fine-grained reasoning tasks.

Our main contributions are summarized as follows:

- We propose AnomalyKV, a novel adaptive anomaly-focused framework that integrates coarse-grained anomaly detection with fine-grained dynamic compression to enable efficient

inference with progressively enhanced anomaly perception. To the best of our knowledge, this is the first KV cache compression method tailored for the IAD domain.
- We introduce AADC, a flexible compression strategy that significantly improves the inference efficiency of state-of-the-art MLLMs while overcoming the inherent limitations of conventional KV cache compression in IAD tasks, minimizing performance degradation caused by such compression.
- Extensive experiments on the latest and largest-scale multimodal IAD benchmark demonstrate that our method effectively reduces inference time cost while maintaining stable performance, thereby substantially enhancing practical inference efficiency.

## 2 Related Work

### 2.1 Industrial Anomaly Detection

Traditional industrial anomaly detection methods typically employ conventional image processing techniques to identify and locate defects using only normal samples for training. Unsupervised IAD approaches can be specifically categorized into three types: synthesis-based methods (Zavrtanik et al., 2021), reconstruction-based methods (Deng & Li, 2022; Jiang et al., 2024), and feature-embedding-based methods (Yi & Yoon, 2020; Gudovskiy et al., 2022). These methods often rely on high-quality normal samples and require pre-trained models for specific object categories, limiting their practical effectiveness in real-world scenarios. Recent research has primarily leveraged few-shot and zero-shot anomaly detection approaches (Zhou et al., 2024) to address IAD tasks, which generally aim to guide vision-language models to generate results through text prompts. However, these methods still mainly operate in a pre-trained paradigm, exhibiting significant limitations in deep, continuous reasoning capabilities.

### 2.2 Multimodal Large Language Model for IAD

In recent years, MLLMs have demonstrated significant potential in a range of vision-language tasks, particularly in those requiring strong reasoning capabilities. Similarly, research integrating MLLMs with IAD has attracted growing attention. Among these, Myriad (Li et al., 2024e) stands out as a pioneering approach that employs a pre-trained IAD model to locate defective regions and then guides the MLLM through textual instructions for reasoning, establishing a foundational framework combining visual expert models with large models. Nevertheless, constructing visually expert models tailored to specific needs is often challenging and time-consuming, primarily due to the pre-training phase. This poses substantial challenges for practical deployment in real-world scenarios.

Meanwhile, MMAD (Jiang et al., 2025) became the first comprehensive benchmark for anomaly understanding by designing multiple key IAD subtasks. Building upon this foundation, AnomalyCoT (Cheng et al., 2025) constructs challenging composite IAD tasks with high-quality long reasoning chains and significantly expands the sample scenarios to form the largest-scale anomaly understanding benchmark. AnomalyGPT (Gu et al., 2024) synthesizes anomalous samples along with their corresponding textual descriptions, and employs a prompt learner to guide the VLM, further enabling the image decoder to predict anomaly maps. Advanced approaches like OmniAD (Zhao et al., 2025) and AnomalyR1 (Chao et al., 2025) introduce reinforcement learning to enhance model reasoning, achieving a highly effective unified paradigm for detection and understanding.

### 2.3 KV Cache Compression

With the continuous evolution of large language models and their expanding capabilities, multimodal long-context inputs have become a common requirement, making memory limitations an urgent issue to address. Existing compression methods can be broadly categorized into token compression and KV cache compression approaches. Early KV cache compression techniques were primarily designed for LLMs; for instance, SnapKV (Li et al., 2024f) selects important KV positions based on attention scores to achieve automatic KV cache compression. More recent methods focus on advanced VLMs, such as SparseMM (Wang et al., 2025), which allocates asymmetric computational budgets according to visual attention scores from attention heads, thereby preserving KV pairs with higher attention within windows. Although these studies have achieved significant progress on general tasks, there remains a gap in strategies for accelerating reasoning tailored to specific domain thinking patterns.

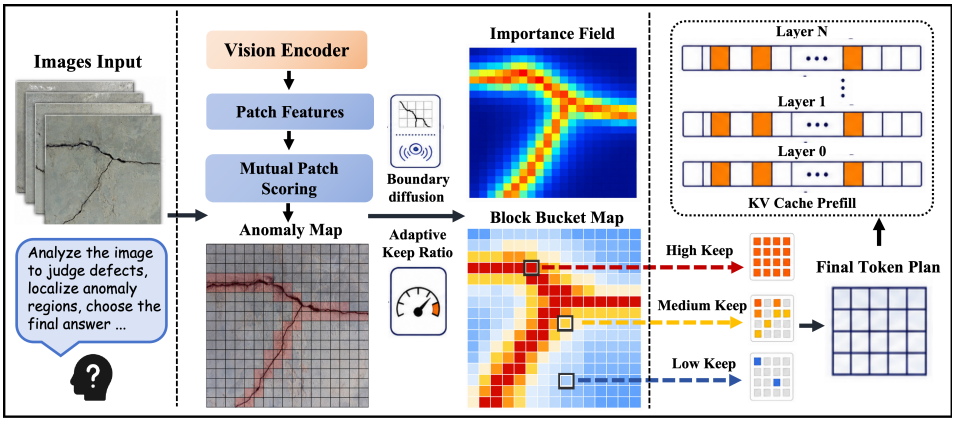



Figure 2: The pipeline of our anomaly-guided KV cache compressor.

## 3 Locate, Budget, Compress: Anomaly-Guided KV Cache Compression

Fig. 2 shows the full pipeline. Given a category of industrial images $\{I_i\}$ and a shared VQA prompt, our compressor runs three stages before the MLLM enters its prefill. The *locate* stage reuses the MLLM's vision encoder, scores every patch against its peers in the same category, and emits an anomaly map per image (Sec. 3.1). The *budget* stage diffuses this map across the patch grid into a boundary-aware importance field, then converts the field into block-level keep tiers and a per-image keep ratio (Sec. 3.2). The *compress* stage finalizes a single token-selection plan from these tiers and applies it once to every prefill attention layer, so generation runs on a fixed, pre-compressed KV cache (Sec. 3.3).

### 3.1 Locating Anomalies by Mutual Scoring (LAMS)

Within one product category most images are normal, and anomalies are rare, small, and visually subtle (Liu et al., 2024b; Xie et al., 2024). This sparsity is the handle the upstream compressor needs. A patch is anomalous when it sits far from the same-location patches of the other images in its category, so we can localize anomalies without any training simply by scoring every patch against its peers (Li et al., 2024d). We trace this path in the left panel of Fig. 2, from raw images through the vision encoder and mutual patch scoring to a per-image anomaly map; the locations that stand out form the anomaly bounding-box set $\mathcal{B}_i$ that Sec. 3.2 consumes.

For image $I_i$ we reuse the MLLM's own vision encoder (Bai et al., 2025; Zhu et al., 2025). We take the patch features at a fixed intermediate layer $\ell^*$, giving $Z_i \in \mathbb{R}^{H\times W\times C}$, and $\ell_2$-normalize along the channel dimension to get $\tilde{Z}_i$. The MLLM consumes the same $\tilde{Z}_i$ at generation time, so we cache it on CPU and reuse it as-is. For a category of $N$ images with the same $(H, W)$, we flatten each $\tilde{Z}_i$ to $\mathbb{R}^{L\times C}$ with $L = HW$. For every ordered pair $(i, j)$ with $j \neq i$, we form the patch-to-image nearest-neighbour distance

$$m_{i\to j}(\ell) = \min_{\ell'} \left\| \tilde{Z}_i[\ell] - \tilde{Z}_j[\ell'] \right\|_2, \tag{1}$$

for $\ell = 1, \ldots, L$. Stacking these into $M_i \in \mathbb{R}^{L\times(N-1)}$, a patch is anomalous only if it is far from *every* other image's most similar patch. We aggregate with an interval mean over the low-quantile band $[\tau_0, \tau_1]$ of the sorted row,

$$A_i(\ell) = \frac{1}{k_1 - k_0 + 1} \sum_{k=k_0}^{k_1} M_i[\ell, (k)], \tag{2}$$

with $k_0 = \lfloor \tau_0(N-1) \rfloor$, $k_1 = \lfloor \tau_1(N-1) \rfloor$, and default $(\tau_0, \tau_1) = (0, 0.3)$. The low band keeps the score sensitive to the few peers that truly resemble patch $\ell$; an outlier image that happens to be far from

everything inflates the upper tail but leaves the band intact, and averaging inside the band is more stable than a single minimum, which one near-duplicate neighbour can dominate. We reshape $A_i$ back to $(H, W)$, bilinearly up-sample it to image resolution, min–max normalize it to $[0, 1]$, threshold at $\theta$, and fit an axis-aligned rectangle to every connected component of area at least $a_{\min}$. This gives the anomaly bounding-box set $\mathcal{B}_i = \{B_{i,1}, \ldots, B_{i,K_i}\}$ that Sec. 3.2 consumes. A token-count guard $L \le L_{\max}$ keeps the $\mathcal{O}(N^2L^2)$ cost of Eq. (1) tractable on high-resolution inputs. We enforce it by bilinearly down-sampling $\tilde{Z}_i$ before scoring and up-sampling $A_i$ afterwards. Because the encoder forward is shared with the MLLM, the only extra cost LAMS adds over uncompressed inference is the scoring step and the bounding-box pass.

### 3.2 Budgeting Tokens by Diffusion Fields

The bounding-box set $\mathcal{B}_i$ from Sec. 3.1 is binary and hard-edged. Two properties of industrial images make this inadequate for setting a KV-cache budget. First, anomalies are spatially sparse but diagnostically multi-scale. A defect often covers well under 5% of the image, yet the texture around it still carries strong diagnostic cues. Second, importance near a defect must stop at semantic boundaries. Budget that spreads across product or part seams is wasted on tokens the MLLM could reconstruct without ever looking. We therefore turn $\mathcal{B}_i$ into a continuous, boundary-aware importance field $\phi_i^* \in [0, 1]^{H\times W}$ by treating it as the source term of an anisotropic reaction–diffusion process; the middle panel of Fig. 2 visualizes this field and the block bucket map it induces. The field is the single quantitative object on which Sec. 3.3 builds its token-selection plan.

We keep the notation of Sec. 3.1. The patch features $\tilde{Z}_i \in \mathbb{R}^{H\times W\times C}$ and the bounding-box set $\mathcal{B}_i$ both live on the patch grid $\Omega = [H] \times [W]$. We first build a scalar structure map

$$G_i(x, y) = \left\|\nabla \tilde{Z}_i(x, y)\right\|_2, \tag{3}$$

taken as the norm of a centered spatial finite difference summed over channels. $G_i$ is large at product and part boundaries and small in flat regions. It will act as a diffusion barrier that keeps importance from leaking across semantic seams. We then define two spatial fields driven by $\mathcal{B}_i$. For each patch $(x, y)$, let $M_i(x, y)$ be the fraction of the patch cell covered by the boxes of $\mathcal{B}_i$, and let $C_i(x, y) = \max_{B\in\mathcal{B}_i} \exp(-d_B(x, y)^2/(2\sigma^2))$ be a Gaussian decay of the distance $d_B$ from $(x, y)$ to the nearest box. The source term combines the three,

$$s_i = \alpha\,(w_M M_i + w_C C_i)\,(\beta + (1 - \beta)G_i), \tag{4}$$

with $w_M + w_C = 1$ and $\beta \in (0, 1)$ a floor that stops the structure factor from killing the anomaly signal where $G_i$ is near zero. Box coverage and spatial proximity jointly decide how much importance is injected at a patch, and structured patches get a multiplicative bonus.

The importance field is the steady state of the anisotropic reaction–diffusion equation

$$\partial_t \phi_i = \nabla\cdot(D_i \nabla\phi_i) - \kappa_i \phi_i + s_i, \tag{5}$$

with spatially varying coefficients

$$D_i = \exp(-G_i^2/k^2), \quad \kappa_i = \kappa_0 + \kappa_1(1 - C_i). \tag{6}$$

$D_i$ is small at semantic edges, so diffusion is slowed there. $\kappa_i$ is larger far from any box, so importance decays away from defects. The steady state is a convolution of $s_i$ with an exponentially decaying, direction-aware kernel whose decay scale is $\sqrt{1/\kappa_0}$ and whose cross-boundary spread is suppressed by $D_i$.

We discretize Eq. (5) on the $H \times W$ grid with a damped Jacobi iteration:

$$\phi_i^{(t+1)} = (1-\omega)\phi_i^{(t)} + \omega\,\frac{s_i + L_d\phi_i^{(t)}}{\kappa_i + S_d}, \tag{7}$$

where $\mathcal{N}$ is the 4-neighbourhood, $d_i$ is the edge-harmonic mean of $D_i$, and $S_d = \sum_{\mathcal{N}} d_i$,

**Algorithm 1** Importance Field via Reaction–Diffusion

**Input:** features $\tilde{Z}_i$, boxes $\mathcal{B}_i$.
**Output:** field $\phi_i^*$, structure map $G_i$.

$G_i \leftarrow \|\nabla \tilde{Z}_i\|_2$
$M_i, C_i \leftarrow$ box coverage and Gaussian decay
$s_i \leftarrow$ source by Eq. (4)
$D_i \leftarrow e^{-G_i^2/k^2}$; $\kappa_i \leftarrow \kappa_0 + \kappa_1(1 - C_i)$
$\phi_i^{(0)} \leftarrow s_i/\kappa_i$
**for** $t = 0, \ldots, T - 1$ **do**
    update $\phi_i^{(t+1)}$ by Eq. (7)
**end for**
$\phi_i^* \leftarrow \text{normalize}(\phi_i^{(T)} + \eta G_i)$
**return** $(\phi_i^*, G_i)$

$L_d\phi = \sum_{\mathcal{N}} d_i \phi$. We initialize at $\phi_i^{(0)} = s_i/\kappa_i$ and run $T$ iterations with relaxation $\omega \in (0, 1]$. We add a residual structure term back at the end to keep the field well defined on images with no detected box,

$$\phi_i^* = \text{normalize}\big(\phi_i^{(T)} + \eta\, G_i\big) \in [0,1]^{H\times W}, \tag{8}$$

where the normalization is a min–max rescaling to $[0, 1]$. When $\mathcal{B}_i = \varnothing$, the source vanishes and $\phi_i^*$ falls back to a normalized structure map instead of collapsing to zero, so the downstream allocator still sees a well-defined signal. Algorithm 1 summarizes the full procedure.

**Complexity.** Building the structure map is $\mathcal{O}(HWC)$, the source and coefficients are $\mathcal{O}(HW)$, and each Jacobi sweep is $\mathcal{O}(HW)$. The total per image is $\mathcal{O}(THW + HWC)$ with $T$=8, linear in token count. The field is decoupled from attention, so the method is FlashAttention-compatible.

### 3.3 Compressing the Cache by Block Plans

The importance field $\phi_i^*$ is continuous, but a KV cache is addressed by discrete token indices. We therefore turn $\phi_i^*$ into a block-level token-selection plan that is built once per image and reused by every prefill attention layer. We tile the patch grid $\Omega = [H] \times [W]$ into disjoint $b \times b$ blocks $\{\mathcal{P}_r\}_{r=1}^R$ with default $b = 16$. Each block gets an importance score equal to the mean of $\phi_i^*$ on its patches, $\bar{\phi}_{i,r} = |\mathcal{P}_r|^{-1} \sum_{(x,y)\in\mathcal{P}_r} \phi_i^*(x, y)$. Working at the block level rather than the patch level absorbs local noise in $\phi_i^*$ and keeps the selected tokens spatially coherent.

The total visual budget comes from $\phi_i^*$ itself. Let $m_i = \overline{\phi_i^*} \in [0, 1]$ be the per-image importance mass. We set a content-adaptive keep ratio and budget

$$\rho_i = \rho_{\min} + (\rho_{\max} - \rho_{\min})\, m_i, \quad T_i = \lceil \rho_i L \rceil, \tag{9}$$

with $L = HW$ and $(\rho_{\min}, \rho_{\max})$ a global floor and cap, defaults $(0.05, 0.20)$. Images with broadly excited importance fields keep a larger share of tokens within the cap, while near-uniform normal images are compressed aggressively. We then spread $T_i$ across blocks in proportion to $\bar{\phi}_{i,r}$ with a minimum-keep floor of $T_{\min} = 1$ per block, so no region is ever fully silenced. Inside each block we take the top patches ranked by $\phi_i^*$, with $G_i$ of Eq. (3) breaking ties. The right panel of Fig. 2 illustrates the result: blocks landing on the diffused defect form the *high-keep* tier and retain almost all their patches, blocks on its periphery fall into the *medium-keep* tier and retain a mid-density subset, and blocks far from any defect collapse into the *low-keep* tier with only the floor $T_{\min}$ patches. Collecting the selected indices over all blocks gives the visual token plan $\mathcal{T}_i^{\text{vis}}$.

**Algorithm 2** Anomaly-Guided KV-Cache Compression

**Input:** images $\{I_i\}_{i=1}^N$, prompts $\{q_i\}$, MLLM with encoder $\mathcal{E}$ and $L_{\text{LLM}}$ layers.
**Output:** answers $\{y_i\}_{i=1}^N$.

**for** $i = 1, \ldots, N$ **do**
  $\tilde{Z}_i \leftarrow \ell_2\text{-norm}(\mathcal{E}^{(\ell^*)}(I_i))$
**end for**
**for** each subgroup $\mathcal{G}$ with equal $(H, W)$ **do**
  $\{\mathcal{B}_i\}_{\mathcal{G}} \leftarrow \text{LAMS}(\{\tilde{Z}_i\}_{\mathcal{G}})$
**end for**
**for** $i = 1, \ldots, N$ **do**
  $(\phi_i^*, G_i) \leftarrow \text{FIELD}(\tilde{Z}_i, \mathcal{B}_i)$
  $\bar{\phi}_{i,r} \leftarrow$ block means of $\phi_i^*$
  $T_i \leftarrow \lceil \rho_i L \rceil$ by Eq. (9)
  $T_{i,r} \leftarrow$ split $T_i \propto \bar{\phi}_{i,r}$, floor $T_{\min}$
  $\mathcal{T}_i^{\text{vis}} \leftarrow$ top-$T_{i,r}$ patches per block
  $\mathcal{T}_i \leftarrow \mathcal{A}_i \cup \mathcal{U}_i \cup \mathcal{S}_i \cup \mathcal{T}_i^{\text{vis}}$
**end for**
**for** $i = 1, \ldots, N$ **do**
  prefill with cache $\{(K^{(\ell)}, V^{(\ell)})[\mathcal{T}_i]\}_\ell$
  $y_i \leftarrow$ decode
**end for**
**return** $\{y_i\}_{i=1}^N$

Text tokens around the visual span cannot be scored by $\phi_i^*$, so we compress them with an anchor-preserving policy. Let $\mathcal{A}_i$ be the anchor index set that is kept unconditionally, covering the `<|im_start|>/system` header, the `<|im_end|>` boundary of the system turn, the user-turn header, and the image-tag tokens adjacent to the visual span. Let $\mathcal{U}_i$ be the last-$\tau$ tail indices with default $\tau = 40$, so the user's question and the assistant turn header reach the model intact. Let $\mathcal{S}_i$ be the remaining body of system-level boilerplate uniformly subsampled at ratio $\rho_{\text{sys}}$, default 0. The three sets form the text plan $\mathcal{T}_i^{\text{txt}} = \mathcal{A}_i \cup \mathcal{U}_i \cup \mathcal{S}_i$, which is concatenated with $\mathcal{T}_i^{\text{vis}}$ in original order into a single token-selection plan $\mathcal{T}_i$.

At inference, the same $\mathcal{T}_i$ is applied in every layer's attention module during prefill—one plan, all layers, as sketched at the far right of Fig. 2. After the projection $K, V = XW_K, XW_V$, we gather $\hat{K} = K[\mathcal{T}_i]$ and $\hat{V} = V[\mathcal{T}_i]$, and store only $(\hat{K}, \hat{V})$ in the KV cache. Decoding steps with $q = 1$

Table 1: Performance on AnomalyCoT with reduced KV cache budgets. Full KV refers to caching all KV pairs of the MLLM (upper bound).

| Methods | AnomalyCoT | | | | | | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | MVTecAD | MVTec-LOCO | VisA | GoodsAD | MPDD | MSD | MIAD | DS | NanoTwice | Self | |
| **Qwen2.5-VL-7B-Instruct** | | | | | | | | | | | |
| Full KV | 62.64 | 61.76 | 53.41 | 48.12 | 43.79 | 48.68 | 35.17 | 28.03 | 28.16 | 55.53 | 45.26 |
| *KV Cache Size = 64* | | | | | | | | | | | |
| SnapKV | 59.00 | 38.01 | **49.50** | **44.80** | 38.06 | 42.67 | 29.20 | 24.07 | 24.03 | **49.91** | 39.28 |
| AdaKV | 59.22 | **40.94** | 49.23 | 43.66 | **38.08** | **43.05** | **29.45** | 23.55 | 23.60 | 49.83 | **39.39** |
| PyramidKV | **59.89** | 36.23 | 47.03 | 44.01 | 37.36 | 37.62 | 27.93 | **24.72** | **27.02** | 40.49 | 36.28 |
| *KV Cache Size = 128* | | | | | | | | | | | |
| SnapKV | 57.86 | **59.54** | 48.32 | 43.06 | **38.70** | 43.26 | 29.73 | **23.10** | 23.07 | **50.26** | 40.11 |
| AdaKV | 57.73 | 58.77 | 48.19 | 43.07 | 38.09 | **43.50** | **29.88** | 23.02 | 23.15 | 50.19 | **40.11** |
| PyramidKV | **59.33** | 38.08 | **49.44** | **43.70** | 37.96 | 42.52 | 29.75 | 22.93 | **23.48** | 49.81 | 39.40 |
| *KV Cache Size $\leq$ 256* | | | | | | | | | | | |
| SnapKV | 57.56 | 56.68 | 48.33 | 43.04 | 38.71 | 43.60 | 30.09 | 22.95 | 23.08 | 50.45 | 40.18 |
| AdaKV | 57.50 | 57.53 | 48.43 | 43.02 | 37.81 | 43.45 | 30.00 | 22.96 | 23.09 | 50.51 | 40.17 |
| PyramidKV | 57.96 | 53.50 | 48.44 | 43.12 | 37.31 | 42.84 | 29.43 | 22.93 | 23.06 | 49.91 | 39.65 |
| **AnomalyKV** | **62.42** | **65.29** | **54.70** | **50.17** | **44.83** | **57.17** | **32.10** | **31.56** | **38.46** | **56.59** | **44.86** |
| **InternVL3-8B** | | | | | | | | | | | |
| Full KV | 44.50 | 69.81 | 52.16 | 43.59 | 39.11 | 66.88 | 29.27 | 33.47 | 9.72 | 61.68 | 43.14 |
| *KV Cache Size = 64* | | | | | | | | | | | |
| SnapKV | 36.03 | 47.14 | **41.36** | 31.94 | 27.32 | 52.57 | 21.22 | 22.87 | 11.98 | 50.62 | 33.07 |
| AdaKV | 28.91 | **50.74** | 28.27 | 28.60 | 20.28 | **52.62** | 22.93 | **24.56** | 18.37 | 50.00 | 32.17 |
| PyramidKV | **38.29** | 40.18 | 40.59 | **32.66** | **29.34** | 50.03 | **24.44** | 24.49 | **18.78** | **52.24** | **34.78** |
| *KV Cache Size $\leq$ 128* | | | | | | | | | | | |
| SnapKV | 39.99 | 63.09 | 46.94 | 39.00 | 32.83 | 59.57 | 22.22 | 27.55 | 13.57 | 53.47 | 36.41 |
| AdaKV | **40.78** | 68.14 | 45.92 | 33.57 | 34.66 | 60.80 | 23.27 | 30.70 | 23.40 | 51.31 | 36.33 |
| PyramidKV | 40.05 | 46.30 | 46.08 | 38.76 | 33.50 | 57.71 | 24.14 | 27.16 | 24.22 | 53.66 | 36.54 |
| **AnomalyKV** | 39.53 | **68.54** | **60.14** | **49.92** | **43.18** | **71.31** | **35.71** | **36.04** | **36.00** | **60.93** | **40.70** |
| *KV Cache Size = 256* | | | | | | | | | | | |
| SnapKV | 37.39 | 62.70 | 45.05 | 36.48 | 32.00 | 59.77 | 22.16 | 26.36 | 2.61 | **54.57** | 36.03 |
| AdaKV | **41.50** | **66.90** | 46.65 | 34.89 | **38.47** | 56.03 | **23.33** | **31.06** | **23.56** | 52.76 | **36.74** |
| PyramidKV | 40.25 | 62.59 | **46.97** | **39.93** | 32.12 | **60.86** | 21.20 | 26.95 | 13.56 | 53.19 | 35.98 |
| **Gemma4-E-4B-IT** | | | | | | | | | | | |
| Full KV | 52.34 | 55.01 | 57.62 | 45.51 | 42.86 | 61.39 | 32.20 | 25.81 | 28.57 | 53.38 | 42.81 |
| *KV Cache Size = 64* | | | | | | | | | | | |
| SnapKV | 56.40 | 35.41 | 46.90 | 42.20 | 35.45 | **40.07** | 26.60 | 21.47 | 21.43 | **47.31** | 36.68 |
| AdaKV | 55.91 | **37.63** | 45.93 | 40.36 | 34.78 | 39.75 | 26.15 | 20.25 | 20.30 | 46.52 | 36.09 |
| PyramidKV | **61.07** | 37.42 | **48.21** | **45.19** | **38.54** | 38.81 | **29.11** | **25.90** | **28.21** | 41.67 | **37.46** |
| *KV Cache Size $\leq$ 128* | | | | | | | | | | | |
| SnapKV | 56.72 | 58.40 | 47.18 | 41.92 | 37.56 | 42.12 | 28.59 | 21.96 | 21.93 | 49.12 | 38.97 |
| AdaKV | 56.31 | 57.35 | 46.77 | 41.65 | 36.67 | 42.08 | 28.46 | 21.59 | 21.73 | 48.76 | 38.69 |
| PyramidKV | 59.71 | 38.46 | 49.82 | **44.08** | **38.34** | 42.90 | **30.13** | 23.30 | 23.85 | 50.19 | 39.78 |
| **AnomalyKV** | **60.48** | **74.76** | **58.90** | 35.98 | 31.03 | **74.69** | 29.46 | **28.33** | **46.15** | **57.88** | **43.74** |
| *KV Cache Size = 256* | | | | | | | | | | | |
| SnapKV | 57.24 | 56.36 | 48.01 | 42.72 | **38.39** | 43.28 | 29.77 | 22.63 | 22.76 | 50.13 | 39.86 |
| AdaKV | 57.08 | **57.10** | 48.01 | 42.59 | 37.38 | 43.02 | 29.58 | 22.54 | 22.67 | 50.08 | 39.75 |
| PyramidKV | **58.89** | 54.44 | **49.38** | **44.06** | 38.25 | **43.78** | **30.37** | **23.87** | **24.00** | **50.84** | **40.59** |

append new tokens to the cache without further compression, so autoregressive generation sees a static, pre-compressed context. Algorithm 2 summarizes the full pipeline, from a category of raw images to per-image compressed caches and answers.

**End-to-end complexity.** On top of the encoder forward $\mathcal{O}(NF_{\mathcal{E}})$ that any MLLM pipeline pays, AnomalyKV adds (i) one mutual-scoring pass per category at $\mathcal{O}(N^2 L_{\max}^2)$ under the token-count guard, and (ii) per-image planning at $\mathcal{O}(THW + HWC)$, dominated by the importance field with $T$=8. Compressed prefill costs $\mathcal{O}(L_{\text{LLM}}|\mathcal{T}_i|^2 d)$ in attention, with $|\mathcal{T}_i| \approx \rho_i L \leq \rho_{\max} L$ — strictly below the uncompressed $\mathcal{O}(L_{\text{LLM}} L^2 d)$ by a factor of $\rho_i^2$. Because the plan is built once and reused by all $L_{\text{LLM}}$ layers, planning contributes well under one percent of end-to-end wall-clock.

## 4 EXPERIMENTS

We conducted comprehensive experiments to validate AnomalyKV's effectiveness in anomaly perception and cost reduction. First, we detail the experimental setup in Section 4.1. Next, in Section 4.2, we compare our method with mainstream KV cache compression techniques, demonstrating that our approach maintains strong performance on domain-specific tasks while significantly optimizing inference efficiency. Finally, in Section 4.3, we perform systematic ablation studies and analysis to justify the rationality of the key design components in our framework.

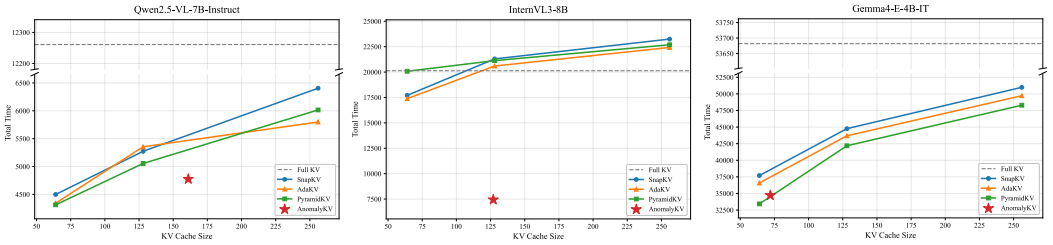



Figure 3: Total inference time comparison of different KV-cache compression methods across three vision-language models.

### 4.1 Experimental Settings

**Models.** In this experiment, we evaluate AnomalyKV using several mainstream multimodal large models to ensure its applicability: Qwen2.5-VL-7B-Instruct (Bai et al., 2025), InternVL3-8B (Zhu et al., 2025), and Gemma4-E-4B-IT. The visual encoder of Qwen2.5-VL-7B-Instruct employs a Vision Transformer (ViT) based on dynamic resolution, integrating window attention mechanisms and optimizing the ViT architecture with SwiGLU and RMSNorm to accelerate computation and significantly improve inference speed. InternVL3-8B adopts the specialized architecture of the InternVL series, utilizing the InternViT-300M configuration as its visual encoder. Similarly, the latest Gemma4-E-4B-IT also uses a ViT-based visual encoder, incorporating learned 2D positions and multidimensional RoPE. Additional implementation details are provided in Appendix A.

**Baselines.** We conduct a comprehensive comparison of AnomalyKV with three baselines—SnapKV (Li et al., 2024f), AdaKV (Feng et al., 2025), and PyramidKV (Cai et al., 2024)—which represent the most mainstream KV cache compression methods currently available. Full KV serves as the performance benchmark, directly caching the KV pairs for each token at every layer.

**Benchmark.** To comprehensively evaluate the effectiveness of AnomalyKV across different scenarios, we use AnomalyCoT (Cheng et al., 2025) to test the performance of KV cache compression methods on anomaly understanding tasks. The dataset provides precise annotated bounding boxes for all samples, facilitating verification of whether our optimization in detection methods is effective. For all category scenarios, we follow the evaluation metrics adopted by AnomalyCoT.

### 4.2 Main Results

**Setups.** We evaluate the baseline methods with a unified and fixed KV cache budget range: 64, 128, 256. Considering AnomalyKV's fine-grained adaptive allocation of KV cache budgets, to preserve the dynamic nature of compression methods without introducing additional constraints and to ensure fair comparisons with baselines, we compute the average KV cache budgets used by AnomalyKV for inference across different MLLMs: 72, 127, 161, and comprehensively compare their performance against higher corresponding baseline budgets: 128, 128, 256. In our experiments, we use consistent prompts across all scenarios, with the prompt template shown in Fig. **??**.

**Results.** In Table 1, we present the detailed evaluation results of three models across ten data scenarios in AnomalyCoT. Experimental results show that AnomalyKV not only significantly outperforms baseline methods under extreme KV cache sizes but also maintains performance close to that of uncompressed MLLMs, achieving nearly lossless KV cache compression. For instance, when performing anomaly understanding tasks using Qwen2.5-VL-7B-Instruct, there is virtually no performance degradation at a KV cache size of 161, surpassing the baseline method by approximately 5% accuracy compared to the maximum KV cache size setting of 256. These results demonstrate the superior performance of our method in anomaly understanding tasks. Furthermore, findings in Table 1 clearly indicate that visually unbiased KV cache compression methods often "lose focus" on specific domain reasoning tasks, whereas our approach achieves better results by progressively enhancing the model's ability to perceive key targets.

Figure 3 demonstrates that our method achieves compression corresponding to different KV cache sizes across three MLLMs, significantly reducing the overall time cost and substantially improving

inference efficiency. For instance, when the KV cache size is 127, AnomalyKV achieves a maximum speedup of 2.71× on InternVL3-8B. Experimental results show that our approach greatly optimizes model inference efficiency, with particularly notable performance in scenarios involving high-resolution, long-context, multimodal inputs.

### 4.3 Ablation Studies and Analysis

**Performance Influence of Visual Encoder Structure.** To investigate the impact of visual encoder architecture on coarse-grained anomaly perception, we conducted a series of combinatorial experiments. In these experiments, we first uniformly sampled eight different layers across the visual encoder's layer sequence for testing. Considering the potential performance gains from multi-layer fusion, after identifying the optimal layer, we combined it with its adjacent layers to form two dual-layer groups. Based on this, we selected four consecutive layers covering both groups, forming a single four-layer combination. By evaluating different grouping configurations, we were able to examine the actual effects of local and multi-layer fusion. We evaluated the visual encoders from Qwen2.5-VL-7B-Instruct and InternVL3-8B on the AnomalyCoT test set. The experimental results show that the peak localization accuracy achieved using a single layer of the visual encoder outperforms combinations of different layers, while dual-layer combinations consistently yield better results than four-layer ones. These findings suggest that multi-layer fusion may actually degrade detection performance, whereas leveraging a single effective layer provides greater overall benefit. Ablation studies on the visual encoder architecture highlight the critical role of intermediate layers in initial anomaly perception assistance. Experimental details are provided in Appendix C.1, with results shown in Fig. 4.

**Performance Influence of Different Metrics in AnomalyKV.** We comprehensively evaluated the key components of our method and conducted ablation experiments using different MLLMs—namely Qwen2.5-VL-7B-Instruct and InternVL3-8B—to ensure analytical completeness. First, we directly identified potentially anomalous regions by computing intra-sample entropy, replacing coarse-grained anomaly detection approaches. Second, after determining the importance of each token in a sample, we applied fixed-ratio compression consistent with the average compression ratio of existing methods based on token significance, thereby ablating the dynamic compression step. Additionally, after deriving fine-grained compression ratios for samples, we performed fully random compression according to these ratios. Ultimately, we found that under any ablation strategy across different models, AnomalyKV's performance decreased to varying degrees. These results clearly demonstrate the necessity and effectiveness of each major component in our method. Detailed experimental protocols and results are provided in Appendix C.2 and Table 5.

## 5 Conclusion

This paper systematically analyzes the inherent limitations of MLLMs in IAD reasoning. Our findings uncover a critical attention phenomenon, where only a negligible fraction of KV pairs corresponding to anomalous regions receive effective attention. Leveraging this insight, we propose AnomalyKV, a novel KV cache compression framework that enables fine-grained adaptive allocation of KV cache budgets. By prioritizing the retention of information with higher importance for anomaly perception in the current scene, AnomalyKV achieves efficient inference with stable performance.

## Limitations

While our study presents innovative insights, our evaluation is constrained to benchmarks within a single domain. In future research, we will assess our method across a broader spectrum of visual anomaly detection benchmarks to validate its generalizability and efficacy. Additionally, although we have achieved substantial improvements in inference efficiency while minimizing performance loss, our approach still relies on a coarse-grained anomaly detection step, which partially limits real-time inference speed. Future investigations will delve into the design of streamlined framework solutions for KV cache compression tailored to specific domains.

## REFERENCES


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. *arXiv preprint arXiv:2502.13923*, 2025.

Jinan Bao, Hanshi Sun, Hanqiu Deng, Yinsheng He, Zhaoxiang Zhang, and Xingyu Li. BMAD: Benchmarks for medical anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 4042–4053, 2024.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4183–4192, 2020.

Zefan Cai, Yichi Zhang, Bofei Gao, Yuliang Liu, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Junjie Hu, and Wen Xiao. PyramidKV: Dynamic KV cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

Tri Cao, Jiawen Zhu, and Guansong Pang. Anomaly detection under distribution shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6488–6500, 2023.

Yuhao Chao, Jie Liu, Jie Tang, and Gangshan Wu. AnomalyR1: A GRPO-based end-to-end MLLM for industrial anomaly detection. *arXiv preprint arXiv:2504.11914*, 2025.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14455–14465, 2024.

Yuanhong Chen, Yu Tian, Guansong Pang, and Gustavo Carneiro. Deep one-class classification via interpolated Gaussian descriptor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 383–392, 2022.

Jiaxi Cheng, Yuliang Xu, Shoupeng Wang, Tao Ma, Yuchen He, Jinghe Zhang, Sihang Cai, Jiawei Zhen, Jingyi Jia, Yao Wan, Yan Xia, and Zhou Zhao. Anomalycot: A multi-scenario chain-of-thought dataset for multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. Datasets and Benchmarks Track.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9737–9746, 2022.

Qian Du, Bo Tang, Weiying Xie, and Wei Li. Parallel and distributed computing for anomaly detection from hyperspectral remote sensing imagery. *Proceedings of the IEEE*, 109(8):1306–1319, 2021.

Yuan Feng, Junlin Lv, Yukun Cao, Xike Xie, and S Kevin Zhou. Ada-KV: Optimizing KV cache eviction by adaptive budget allocation for efficient LLM inference. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Deep learning for medical anomaly detection – a survey. *ACM Computing Surveys*, 54(7):1–37, 2021.

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Anomalygpt: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 1932–1940, 2024.

Wei Guan, Jun Lan, Jian Cao, Hao Tan, Huijia Zhu, and Weiqiang Wang. EMIT: Enhancing MLLMs for industrial anomaly detection via difficulty-aware GRPO. *arXiv preprint arXiv:2507.21619*, 2025.

Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 98–107, 2022.

Qiandong Guo, Ruiliang Pu, and Jun Cheng. Anomaly detection from hyperspectral remote sensing imagery. *Geosciences*, 6(4):56, 2016.

Changhee Han, Leonardo Rundo, Kohei Murao, Tomoyuki Noguchi, Yuki Shimahara, Zoltán Ádám Milacski, Saori Koshino, Evis Sala, Hideki Nakayama, and Shin'ichi Satoh. MADGAN: unsupervised medical anomaly detection GAN using multiple adjacent brain MRI slice reconstruction. *BMC Bioinformatics*, 22(Suppl 2):31, 2021.

Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In *European Conference on Computer Vision (ECCV)*, pp. 303–319, 2022.

Chaoqin Huang, Aofan Jiang, Jinghao Feng, Ya Zhang, Xinchao Wang, and Yanfeng Wang. Adapting visual-language models for generalizable anomaly detection in medical images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11375–11385, 2024.

Xi Jiang, Ying Chen, Qiang Nie, Jianlin Liu, Yong Liu, Chengjie Wang, and Feng Zheng. Toward multi-class anomaly detection: Exploring class-aware unified model against inter-class interference. *arXiv preprint arXiv:2403.14213*, 2024.

Xi Jiang, Jian Li, Hanqiu Deng, Yong Liu, Bin-Bin Gao, Yifeng Zhou, Jialin Li, Chengjie Wang, and Feng Zheng. Mmad: A comprehensive benchmark for multimodal large language models in industrial anomaly detection. In *International conference on learning representations*, volume 2025, pp. 87273–87295, 2025.

Amna Khatoon, Weixing Wang, Mengfei Wang, Limin Li, and Asad Ullah. Tinyml-enabled fuzzy logic for enhanced road anomaly detection in remote sensing. *Scientific Reports*, 15(1):20659, 2025.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. LLaVA-Med: Training a large language-and-vision assistant for biomedicine in one day. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. LLaVA-NeXT-Interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024b.

Jingtao Li, Xinyu Wang, Hengwei Zhao, and Yanfei Zhong. Learning a cross-modality anomaly detector for remote sensing imagery. *IEEE Transactions on Image Processing*, 33:6607–6621, 2024c.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning (ICML)*, 2022.

Xurui Li, Ziming Huang, Feng Xue, and Yu Zhou. MuSc: Zero-shot industrial anomaly classification and segmentation with mutual scoring of the unlabeled images. In *International Conference on Learning Representations (ICLR)*, 2024d.

Yanhui Li, Yunkang Cao, Chengliang Liu, Yuan Xiong, Xinghui Dong, and Chao Huang. Iad-r1: reinforcing consistent reasoning in industrial anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 6583–6591, 2026.

Yuanze Li, Haolin Wang, Shihao Yuan, Ming Liu, Debin Zhao, Yiwen Guo, Chen Xu, Guangming Shi, and Wangmeng Zuo. Myriad: A large multimodal model applying vision experts for industrial anomaly detection. *arXiv preprint arXiv:2310.19070*, 2024e.

Yuanze Li, Shihao Yuan, Haolin Wang, Qizhang Li, Ming Liu, Chen Xu, Guangming Shi, and Wangmeng Zuo. Triad: Empowering lmm-based anomaly detection with expert-guided region-of-interest tokenizer and manufacturing process. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21917–21926, 2025.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024f.

Jingyi Liao, Yongyi Su, Rong-Cheng Tu, Zhao Jin, Wenhao Sun, Yiting Li, Dacheng Tao, Xun Xu, and Xulei Yang. Ad-fm: multimodal llms for anomaly detection via multi-stage reasoning and fine-grained reward optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15234–15242, 2026.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, 2024a.

Jiaqi Liu, Guoyang Xie, Jinbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep industrial image anomaly detection: A survey. *Machine Intelligence Research*, 21(1): 104–135, 2024b.

Junwen Miao, Penghui Du, Yi Liu, Yu Wang, and Yan Wang. Agentiad: Tool-augmented single-agent for industrial anomaly detection. *arXiv preprint arXiv:2512.13671*, 2025.

Shancong Mou, Xiaoyi Gu, Meng Cao, Haoping Bai, Ping Huang, Jiulong Shan, and Jianjun Shi. Rgi: robust gan-inversion for mask-free image inpainting and unsupervised pixel-wise anomaly detection. In *The Eleventh International Conference on Learning Representations*, 2023.

Ivan Račetin and Andrija Krtalić. Systematic review of anomaly detection in hyperspectral remote sensing applications. *Applied Sciences*, 11(11):4878, 2021.

Tal Reiss and Yedid Hoshen. Mean-shifted contrastive loss for anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2155–2162, 2023.

Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14298–14308, 2022.

Jiahui Wang, Zuyan Liu, Yongming Rao, and Jiwen Lu. SparseMM: Head sparsity emerges from visual concept responses in MLLMs. *arXiv preprint arXiv:2506.05344*, 2025.

Julia Wolleb, Florentin Bieder, Robin Sandkühler, and Philippe C. Cattin. Diffusion models for medical anomaly detection. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 35–45, 2022.

Guoyang Xie, Jinbao Wang, Jiaqi Liu, Feng Zheng, and Yaochu Jin. Pushing the limits of few-shot anomaly detection in industry vision: Graphcore. In *International Conference on Learning Representations (ICLR)*, 2023.

Guoyang Xie, Jinbao Wang, Jiaqi Liu, Jiayi Lyu, Yong Liu, Chengjie Wang, Feng Zheng, and Yaochu Jin. IM-IAD: Industrial image anomaly detection benchmark in manufacturing. *IEEE Transactions on Cybernetics*, pp. 2720–2733, 2024.

Jiacong Xu, Shao-Yuan Lo, Bardia Safaei, Vishal M Patel, and Isht Dwivedi. Towards zero-shot anomaly detection and reasoning with multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20370–20382, 2025.

Jihun Yi and Sungroh Yoon. Patch SVDD: Patch-level SVDD for anomaly detection and segmentation. In *Asian Conference on Computer Vision (ACCV)*, 2020.

Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 4571–4584, 2022.

Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. DRAEM – a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8330–8339, 2021.

Ximiao Zhang, Min Xu, Dehui Qiu, Ruixin Yan, Ning Lang, and Xiuzhuang Zhou. MediCLIP: Adapting CLIP for few-shot medical image anomaly detection. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 458–468, 2024.

Chunhui Zhao, Yulei Wang, Bin Qi, and Jia Wang. Global and local real-time anomaly detectors for hyperspectral remote sensing imagery. *Remote sensing*, 7(4):3966–3985, 2015.

He Zhao, Yuexiang Li, Nanjun He, Kai Ma, Leyuan Fang, Huiqi Li, and Yefeng Zheng. Anomaly detection for medical images using self-supervised and translation-consistent features. *IEEE Transactions on Medical Imaging*, 40(12):3641–3651, 2021.

Shifang Zhao, Yiheng Lin, Lu Han, Yao Zhao, and Yunchao Wei. Omniad: Detect and understand industrial anomaly via multimodal reasoning. *arXiv preprint arXiv:2505.22039*, 2025.

Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection. In *International Conference on Learning Representations (ICLR)*, 2024.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

# A Implementation Details

**Models.** We evaluate AnomalyKV on three representative multimodal large language models: Qwen2.5-VL-7B-Instruct, InternVL3-8B, and Gemma4-E-4B-IT. For all models, we use the original HuggingFace checkpoints without task-specific fine-tuning or weight modification. For Gemma4-E-4B-IT, we use the HuggingFace implementation rather than the original JAX implementation. Each model is paired with its official processor for image preprocessing and prompt construction, so the input format, visual tokenization, and chat template follow the default setting of the corresponding model family.

**Hardware and inference protocol.** All experiments are conducted on a server with eight NVIDIA RTX 4090 GPUs. We run inference with batch size 1 and process each image independently. Unless otherwise stated, all models and methods use the same prompt template, greedy decoding with temperature 0, and a maximum generation length of 512 tokens. The reported latency is end-to-end wall-clock inference time, including image encoding, LAMS mutual scoring, compression planning, prefill, and autoregressive decoding. Each timing experiment is repeated three times, and we report the average runtime.

**Benchmark and evaluation.** We conduct experiments on the training split of AnomalyCoT. The evaluation follows the official protocol of AnomalyCoT, and the reported accuracy is computed according to its task definition. The bounding-box annotations are used only for analysis, localization evaluation, and hyperparameter sensitivity studies; they are not used as additional supervision during answer generation. When a model output cannot be automatically parsed, we manually inspect the response and map it to the corresponding evaluation label.

**Baselines.** We compare against SnapKV, AdaKV, and PyramidKV using their official or open-source implementations. We keep the original compression mechanisms and recommended implementation choices of these methods, and evaluate them under the cache-budget settings provided by the released codebase, namely 64, 128, and 256. Full KV denotes the uncompressed setting that preserves all key-value pairs for the input context.

**Budget alignment.** The baseline compressors are evaluated with fixed KV cache budgets of 64, 128, and 256, because they require a predefined budget before inference. In contrast, AnomalyKV performs dynamic, content-adaptive compression: its retained KV budget is determined per input from the anomaly-aware token plan and therefore does not need to match an exact preset value. We thus report the average retained KV budgets of AnomalyKV in the main experiments, namely 72, 127, and 161. For fair comparison, each AnomalyKV operating point is compared against the closest higher fixed-budget baseline setting, corresponding to 128, 128, and 256, respectively.

**AnomalyKV configuration.** For coarse anomaly localization, we extract patch features from the 7-th visual encoder layer. This layer is selected based on the visual-encoder ablation study reported in Appendix C.1. We set the LAMS threshold to 0.8, which provides the best trade-off in our sensitivity analysis in Appendix B. The low-quantile interval for mutual scoring is fixed to $(\tau_0, \tau_1) = (0, 0.3)$.

For anomaly-aware compression, we use a block size of $b = 16$. The adaptive visual keep ratio is bounded by $(\rho_{\min}, \rho_{\max}) = (0.05, 0.20)$, and the minimum block-level keep floor is $T_{\min} = 1$. The diffusion field is computed for $T = 8$ Jacobi iterations. Text tokens are compressed with the anchor-preserving policy described in Sec. 3.3: chat boundary tokens and image-adjacent special tokens are always retained, the last $\tau = 40$ text tokens are preserved, and system-level boilerplate is otherwise discarded by setting $\rho_{\text{sys}} = 0$.

# B Hyperparameter Sensitivity

We analyze the two hyperparameters that most directly affect the quality of the anomaly-aware token plan: the LAMS threshold $\theta$ used to convert anomaly maps into connected components, and the block size $b$ used to aggregate patch-level importance into cache-selection units. Unless otherwise specified, the sensitivity experiments are conducted with InternVL3-8B on the full AnomalyCoT training split. Bounding boxes are used only as diagnostic annotations to measure whether the retained regions cover

| Setting | Value |
|---|---|
| *Inference* | |
| Weights | HF original |
| Batch size | 1 |
| Decoding strategy | Greedy |
| Temperature | 0 |
| Maximum generation length | 512 tokens |
| Runtime measurement | End-to-end |
| Number of repeated runs | 3 |
| *AnomalyKV* | |
| Encoder layer | 7-th |
| LAMS $\theta$ | 0.8 |
| LAMS interval | $(0, 0.3)$ |
| Block size $b$ | 16 |
| Keep ratio | $(0.05, 0.20)$ |
| $T_{\min}$ | 1 |
| Diffusion iterations | 8 |
| Text tail length $\tau$ | 40 |
| $\rho_{\text{sys}}$ | 0 |

Table 2: Core implementation settings. Model names and processors are described in the text.

| $\theta$ | Coverage | Area | Cov./Area |
|---|---|---|---|
| 0.50 | 72.1 | 40.6 | 1.77 |
| 0.60 | 65.4 | 28.9 | 2.26 |
| 0.70 | 58.7 | 22.3 | 2.63 |
| 0.75 | 52.1 | 18.5 | 2.82 |
| **0.80** | **61.2** | **18.5** | **3.31** |
| 0.85 | 45.8 | 15.2 | 3.01 |
| 0.90 | 33.9 | 11.8 | 2.87 |

Table 3: Sensitivity to the LAMS threshold $\theta$ on InternVL3-8B. Coverage measures the percentage of ground-truth anomalous pixels covered by selected areas, Area measures the selected area percentage relative to the full image, and Cov./Area reports their ratio.

true anomalous areas. They are not used by AnomalyKV during inference. We report GT anomaly coverage, defined as the fraction of ground-truth bounding-box pixels covered by the selected regions, and the selected area ratio relative to the full image.

**LAMS threshold.** Table 3 compares different threshold values for coarse anomaly localization. Lower thresholds retain broader regions but include more normal background, while overly high thresholds suppress weak or small defects. We therefore compare coverage, selected area, and their ratio. We choose $\theta = 0.8$, which achieves the best coverage–area trade-off in our sweep, covering 61.2% of annotated anomalous pixels with an 18.5% selected area ratio.

**Block size.** Table 4 studies the block size used by the block-level token-selection plan. Smaller blocks provide finer spatial granularity but are more sensitive to noisy localization, whereas larger blocks stabilize the selected regions but can retain excessive normal context. We use $b = 16$ because it provides the best trade-off between anomaly coverage and selected area: it reaches 61.2% coverage while covering only 18.5% of the image area.

## C Ablation Experimental Details

This appendix provides the implementation details of the ablation experiments used to examine the design choices in our framework. The purpose of these experiments is to isolate the effect of each

| $b$ | Coverage | Area | Cov./Area |
|---|---|---|---|
| 4 | 42.3 | 13.4 | 3.15 |
| 8 | 52.9 | 16.7 | 3.17 |
| **16** | **61.2** | **18.5** | **3.31** |
| 32 | 65.7 | 22.8 | 2.88 |
| 64 | 69.1 | 30.3 | 2.28 |

Table 4: Sensitivity to the block size $b$. A moderate block size balances fine-grained anomaly coverage against redundant normal-region retention.

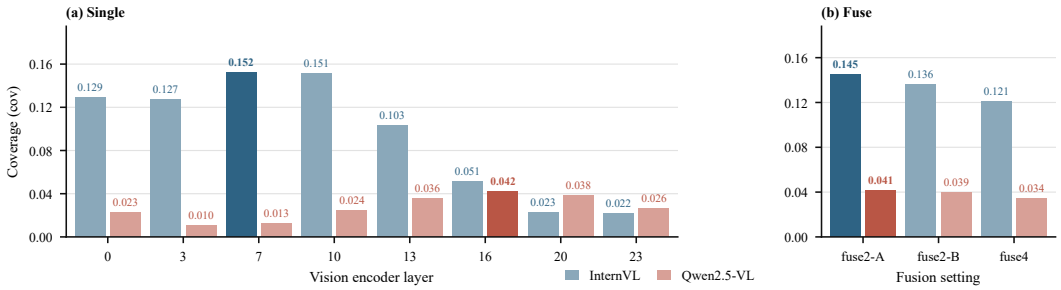



Figure 4: Vision encoder ablations. (a) Single-layer anchors. (b) Fusion brackets. Bars compare InternVL and Qwen2.5-VL under the same ablation setting; darker bars and bold values indicate the best result for each backbone in each panel. For InternVL, fuse2-A/B/fuse4 correspond to $L = 7$–$10$, $L = 10$–$13$, and $L = 7$–$16$. For Qwen2.5-VL, they correspond to $L = 13$–$16$, $L = 16$–$20$, and $L = 13$–$23$.

controlled variable while keeping the remaining inference pipeline fixed. Unless otherwise specified, all ablations use the same data split, image grouping protocol, prompt template, preprocessing pipeline, scoring function, and evaluation metric as the main experiments in Sec. 4. No model parameters are updated during these experiments; all results are obtained by running inference with different fixed configurations.

### C.1 Vision Encoder Layer Ablation

We evaluate how the choice of visual encoder layer affects localization performance. For each backbone, we extract patch-level visual features from a specified vision encoder layer and feed these features into the same downstream anomaly scoring pipeline. The downstream scoring and evaluation procedure are kept unchanged across all layer choices, so the only controlled variable is the selected visual feature layer.

**Backbones.** We conduct the layer ablation on two representative multimodal backbones: InternVL and Qwen2.5-VL. For each backbone, we access the hidden states of the vision encoder and use the patch tokens from the selected layer. The class token, if present, is removed before reshaping. The remaining patch tokens are reshaped into a regular spatial feature grid and normalized before being passed to the downstream scoring stage.

**Single-layer protocol.** For the single-layer ablation, we evaluate the same set of eight depth anchors for both backbones:

$$\ell \in \{0, 3, 7, 10, 13, 16, 20, 23\}.$$

Each run uses exactly one layer. For example, when $\ell = 7$, only the patch features from layer 7 are used, and no feature from any other layer is fused. This protocol allows us to compare shallow, middle, and deep visual representations under the same evaluation setting.

**Fusion protocol.** After running the single-layer sweep, we additionally evaluate a small set of layer-fusion configurations around the strongest single-layer region of each backbone. We use two types of fusion settings:

Table 5: Performance influence of different metrics in AnomalyKV. I and Q denote InternVL3-8B and Qwen2.5-VL-7B-Instruct, respectively.

| Variant | I-Avg. ↑ | I-Time ↓ | Q-Avg. ↑ | Q-Time ↓ |
|---|---|---|---|---|
| **AnomalyKV** | **0.4070** | **7441.23** | **0.4486** | **4774.06** |
| Entropy BBox | 0.3731 | 8742.73 | 0.4032 | 5398.44 |
| Random Pruning | 0.3881 | 7957.09 | 0.4163 | 5167.56 |
| Fixed Compression | 0.3874 | 8599.49 | 0.4303 | 5693.47 |

- `fuse2`: average the patch features from two adjacent layers.
- `fuse4`: average the patch features from four consecutive layers.

For InternVL, the tested fusion brackets are $L = 7$–$10$, $L = 10$–$13$, and $L = 7$–$16$. For Qwen2.5-VL, the tested fusion brackets are $L = 13$–$16$, $L = 16$–$20$, and $L = 13$–$23$. In each fusion run, features from the selected layers are first aligned at the patch-grid level and then averaged element-wise before normalization and scoring. No learned fusion weight is introduced.

**Feature extraction.** All images in a group are forwarded through the same visual encoder under the selected configuration. The extracted patch features are represented as a tensor of shape $[B, H, W, C]$, where $B$ is the number of images in the group, $H \times W$ is the patch grid size, and $C$ is the channel dimension. We apply $\ell_2$ normalization along the channel dimension before computing anomaly scores. This keeps the cosine-based scoring comparable across different layers and fusion settings.

**Evaluation protocol.** We follow the homogeneous-group evaluation protocol. The evaluation contains 7,916 images from 517 groups. Each group contains images from the same semantic category, and the scoring procedure is performed group by group. For each image, the final patch-level anomaly map is upsampled to the original image resolution. The predicted anomalous region is obtained by selecting the top-scoring area according to the same top-$k$ heatmap rule used in the main experiments. Coverage (*cov*) is then computed between the predicted anomalous region and the annotated bounding box. Reported values are group-size-weighted means over all evaluated groups.

**Controlled variables.** To make the layer ablation comparable, we fix all non-layer variables across runs. The image grouping order, maximum number of images per group, preprocessing function, scoring branch, score normalization setting, heatmap upsampling, top-$k$ thresholding, and coverage computation are kept identical. The only variables changed are the visual layer index or the fusion bracket.

### C.2 Component Ablation Protocol

We also conduct component-level ablations to verify the role of different scoring and compression choices. These experiments are designed as controlled replacements: one component is modified at a time while all other parts of the inference pipeline remain fixed. The same models, benchmark split, prompts, decoding setup, and evaluation metrics are used for all variants. Table 5 summarizes the corresponding accuracy and runtime changes.

**Entropy-only variant.** In the entropy-only variant, we replace the original coarse anomaly scoring signal with an intra-sample entropy score. For each image, token-level or region-level uncertainty is computed from the model response distribution or intermediate importance map, depending on the available output of the backbone. Regions with higher entropy are treated as more likely to be anomalous. The rest of the pipeline, including token selection and evaluation, follows the same procedure as the main method.

**Fixed-ratio compression variant.** In the fixed-ratio variant, we remove the dynamic allocation step and apply a constant compression ratio to all samples. The fixed ratio is chosen to match the average retained-token budget of the full method, so that the comparison focuses on whether per-sample allocation is useful rather than on total budget differences. Token importance scores are still computed, but the final retained set is selected using the same global ratio for every sample.



📖 **Question**

Which of the following options correctly identifies the defects in the image?
(A) bulge
(B) surface damage
(C) no defect
(D) dent
(E) crack

😐 **InternVL3**

The image shows a red vintage truck with no visible defects such as bulges, surface damage, dents, or cracks. The vehicle appears to be in good condition, with a clean and intact exterior. The answer is (C) no defect.

😄 **AnomalyKV**

The image provided shows a red pickup truck. Upon close inspection, there are no visible bulges, surface damages, dents, or cracks on the truck's body. The vehicle appears to be in good condition. Therefore, the correct option is: (C) no defect

Figure 5: Qualitative response of the original InternVL3 model. Without anomaly-aware cache selection, the model tends to rely on global visual context and may under-emphasize small defect regions.



📖 **Question**

Which of the following options correctly identifies the defects in the image?
(A) miss
(B) broken
(C) paint defect
(D) no defect
(E) deformation.

😐 **InternVL3**

The image shows a chain hanging loosely from a wire, with a looped section on another wire. This indicates a potential defect in the chain or its attachment.
Option (A) miss: This does not apply as there is no missing part.
Option (B) broken: The chain appears to be broken or not properly attached, as it hangs loosely.
Option (C) paint defect: There is no visible paint defect.
Option (D) no defect: There is a clear defect in the chain.
Option (E) deformation: The chain is deformed, hanging in an irregular manner.
The anomaly is in the chain, specifically the broken or loose section. The coordinates of the defect region are approximately [[(300, 100), (400, 400)]], where the chain is hanging loosely. The answer is (B).

😄 **AnomalyKV**

The image shows a chain hanging loosely from a structure, which suggests that the chain is broken or not properly attached. This indicates a defect in the chain's integrity. The correct option is: (B) broken

Figure 6: Qualitative response of InternVL3 with AnomalyKV. By preserving KV entries around anomaly-relevant regions, AnomalyKV encourages the model to ground its answer in defect-specific evidence.

**Random-with-same-budget variant.** In the random-with-same-budget variant, we keep the per-sample budget produced by the full method but replace the importance-based selection rule with random selection. For each sample, the number of retained tokens is unchanged, but the retained

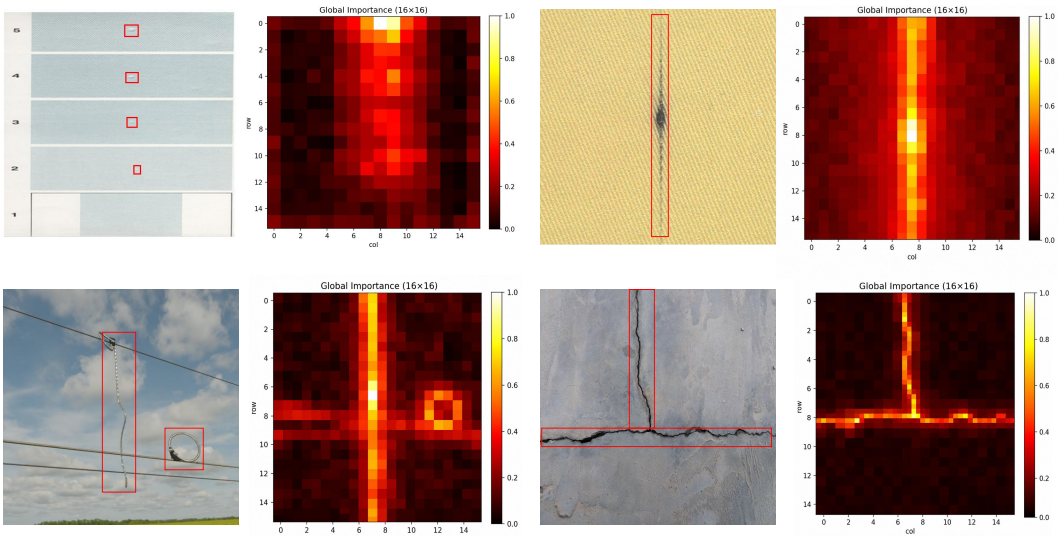



Figure 7: Visualization of anomaly-aware KV cache compression. The figure shows the original image together with the anomaly heatmap, illustrating how AnomalyKV identifies defect-relevant regions and preserves more KV entries around them during compression.

positions are sampled uniformly from the candidate token set. This variant controls for the effect of the budget itself and tests whether the selected token locations matter.

**Evaluation fairness.** All component ablations are evaluated with the same input images, prompts, random seed, and decoding settings. For stochastic variants, the seed is fixed before sampling retained tokens. The reported score is computed with the same metric as the main experiment. No additional tuning is performed separately for ablation variants.

### C.3 Reproducibility Notes

For all ablation experiments, we use deterministic data ordering and fixed random seeds whenever sampling is involved. Each configuration is executed independently from feature extraction to final metric computation. Intermediate feature tensors are not shared across configurations unless they correspond to the same backbone, same input image, and same selected layer. This avoids accidental leakage between single-layer and fusion settings.

All reported numbers are computed from the final saved evaluation records. For each record, we store the backbone name, selected layer or fusion bracket, group identifier, image identifier, predicted heatmap, bounding box annotation, and coverage score.

## D Visualization

Fig. 7 visualizes the original image and the anomaly heatmap used by AnomalyKV to guide anomaly-aware KV cache compression. Fig. 5 and Fig. 6 then provide two qualitative examples comparing the original InternVL3 response with the response produced after applying AnomalyKV. Each example includes the question prompt and the model answer, making it possible to inspect not only the final prediction but also the reasoning focus of the model.

The comparison illustrates the practical effect of anomaly-aware KV cache compression. The original model processes the full visual context but does not explicitly prioritize defect-related patches, so its answer can be dominated by normal object appearance or generic industrial descriptions. In contrast, AnomalyKV first identifies candidate anomalous regions, diffuses their importance to nearby diagnostic context, and allocates more cache budget to these regions during prefill. This changes the information preserved in the KV cache: normal background tokens are compressed more aggressively, while tokens near visual irregularities remain available throughout generation. As a

result, the model response becomes more localized and more consistent with the defect evidence shown in the image, supporting the quantitative gains observed in the main experiments in Table 1.