ADoRe: Normality-Guided Token Pruning for Efficient Anomaly Detection

Anonymous authors
Paper under double-blind review

Abstract

Recent anomaly detectors reach strong accuracy, but their deployment is bound by throughput. A faster decision shortens inspection cycle time and raises the number of objects a surveillance system handles in real time. Defects are small, however, so inputs must stay at high resolution, and the Vision Transformer (ViT) encoder accounts for most of the inference time. Token pruning is a natural remedy, yet existing methods inherit from image classification the intuition that a token which cannot change a single global prediction is safe to discard. This intuition fails in anomaly detection, which labels every location rather than the whole image: faint defects are discarded while salient normal regions are kept. What can be released is a token whose normality is settled. Under ground-truth labels, releasing such tokens carries none of the accuracy cost that pruning usually incurs and can reverse it, for two reasons: anomalous patches would keep their scores while released ones can only lose theirs, and the survivors, encoded with fewer tokens, become easier to separate. Based on this, we propose **ADoRe**, a training-free module that scores each token against a normality reference trimmed to resist the anomalies it must separate, and accumulates its rank over a band of intermediate layers. The most normal patches are released and enter the anomaly map as zero without interpolation. Across CLIP, DINOv2, and video detectors, ADoRe accelerates the encoder by up to 2.47$\times$ while matching or exceeding the unpruned baseline in most settings, and at a fixed time budget, spending the savings on a larger encoder or more crops surpasses the unpruned pipeline.

1 Introduction

Anomaly detection (AD) models the distribution of normal data alone and flags regions that deviate from it. Its main applications, visual inspection in manufacturing and video surveillance, both demand high throughput. In industrial inspection, a faster verdict shortens the cycle time per part (Batzner et al., 2024), and in video surveillance, where each detected object is encoded separately (Reiss & Hoshen, 2025), it raises the number of objects handled in real time. Defects, however, are small and often span only a few pixels, so the input must be kept at high resolution, and passing it through a Vision Transformer (ViT) becomes the latency bottleneck. The encoder accounts for most of the runtime (Tables S6 and S7), making it the primary target for acceleration.

Token pruning is the natural way to accelerate a ViT. It has been developed mainly for image classification (Liang et al., 2022; Fayyaz et al., 2022; Rao et al., 2021; Yin et al., 2022; Bolya et al., 2023) and extended to dense prediction (Tang et al., 2023; Liu et al., 2024), but rarely to AD, whose leading detectors still run on a frozen ViT in full (Section 2.2). Classification removes tokens that cannot change its single global prediction, but AD needs a decision at every location, and influence on a global prediction no longer tells which tokens are removable. A faint defect contributes little to that prediction and is discarded early, while salient normal regions survive to the last layer and leak false positives (Fig. 1, left). The dense-prediction variants do decide per location, but either need category-labeled heads or must still supply a feature at each removed location. In AD, the token that can leave is instead one whose normality is already settled.

Correcting the criterion removes the usual accuracy cost of pruning, and can reverse it. Pruning ground-truth normal tokens, we split the gain into a scoring term, the effect of removing false-positive candidates from the score pool, and an encoding term, the effect of encoding the survivors

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.

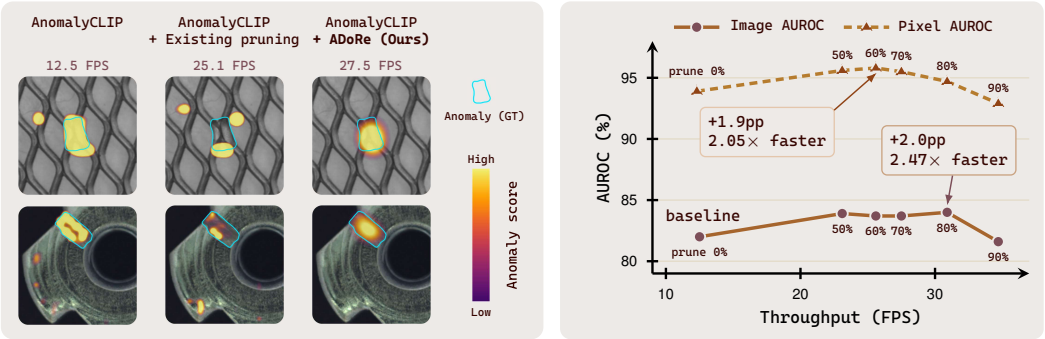


Figure 1: **What a pruning criterion should remove. (Left)** AnomalyCLIP (Zhou et al., 2024) anomaly maps on MVTec-AD (Bergmann et al., 2019) at 70% pruning under two criteria, beside the unpruned model. A classification-style criterion (Liang et al., 2022) keeps salient normal regions to the final layer, leaking false positives, and barely marks the low-saliency defect (ground truth in cyan); ADoRe clears the surroundings and recovers it. **(Right)** AnomalyCLIP accuracy vs. encoder throughput on VisA (Zou et al., 2022), sweeping the pruning ratio. Image and pixel AUROC rise above the baseline with throughput, turning over at 90% as pruning reaches anomalous tokens.

without them. The criterion acts through the scoring term: it is non-negative when the released tokens are normal, and turns negative under a classification criterion as anomalous tokens leak (Table S3). The encoding term is positive under either criterion (Fig. 2), a gain of pruning itself: the survivors become easier to separate. A plausible cause is attention dilution. Self-attention makes tokens more alike with depth (Dong et al., 2021; Wang et al., 2022; Gong et al., 2021), so an anomalous token is pulled toward its normal neighbors, and removing them may weaken this pull (Appendix A.2).

Motivated by these observations, we propose **ADoRe**, a training-free module for efficient **A**nomaly **D**etection via n**o**rmality-guided token **Re**duction. Over a band of intermediate encoder layers, every surviving token is scored against a normality reference estimated from the tokens themselves, and the ones the band agrees are most normal are released. For video, the reference is instead formed from earlier frames in the clip, so that a growing anomaly is compared against a reference that precedes it. A released patch has already been judged normal, so we assign zero at its location rather than interpolating from surviving neighbors, and the anomaly map is assembled in a single pass. By releasing normal tokens, ADoRe reduces encoder computation with little or none of the usual accuracy cost (Fig. 1, right). Moreover, at a fixed time budget, ADoRe can afford a larger encoder or more crops per frame, and surpasses the accuracy of the unpruned pipeline (Fig. 4).

Our contributions are summarized as follows:

- We identify that the token AD can release is one whose normality is settled, not one that cannot change a global prediction, and show under ground-truth labels that releasing such tokens does not trade accuracy for speed: the criterion decides the scoring term, while re-encoding the survivors adds a gain of its own.
- We propose ADoRe, a training-free token pruning method for AD that scores each patch against a trimmed normality reference, accumulates its rank across a band of layers located once per backbone, and writes a released patch straight into the anomaly map with no interpolation.
- We show that ADoRe transfers to CLIP, DINOv2, and video detectors across industrial, medical, and video benchmarks, accelerating the encoder by up to 2.47× with little or no accuracy loss, and that reinvesting the savings surpasses the unpruned pipeline by up to 2.9pp at fixed latency.

2 Related Work

2.1 Token Pruning in Vision Transformers

Token pruning speeds up ViT inference by dropping tokens judged unimportant, and methods differ mainly in how importance is estimated. Attention-based methods score each patch by the attention it

receives from the class token and remove or reorganize the lowest-scoring ones (Liang et al., 2022; Fayyaz et al., 2022; Xu et al., 2022; Wang et al., 2024). Learned methods train auxiliary predictors or halting units to decide which tokens survive, at the cost of extra parameters and training (Rao et al., 2021; Yin et al., 2022; Kong et al., 2022; Meng et al., 2022). Merging methods retain information by fusing redundant tokens rather than discarding them (Bolya et al., 2023; Zeng et al., 2025). All of these are designed around a single global prediction, which a dense task does not provide.

Methods for dense prediction differ mainly in what happens to a token after it leaves. DToP (Tang et al., 2023) exits a token once auxiliary classification heads, trained at each layer, agree on its label. GTP-ViT (Xu et al., 2024) propagates the removed token through a graph onto the survivors, and Liu et al. (2024) keep it recoverable in later layers for the detection head. The first route needs category labels that AD does not have, while the other two must still supply a feature at the exited location. Section 3 asks what should decide a token's fate when neither is available.

2.2 Anomaly detection

Modern AD methods reach strong accuracy by pairing a large frozen backbone with lightweight scoring, and they differ mainly in supervision. Zero-shot methods build on vision-language models, mostly CLIP, and turn image-text similarity into a localization signal, either by engineering prompts and aggregation around a frozen backbone (Jeong et al., 2023; Chen et al., 2025) or by optimizing abnormality prompts and adapters on auxiliary data (Zhou et al., 2024; Cao et al., 2024; Gu et al., 2024; Qu et al., 2024; Zhu et al., 2025); a separate line drops text supervision and repurposes visual foundation models such as DINOv2 (Fučka et al., 2026; Hou et al., 2026a). Few-shot methods store a handful of normal references and score patches against them, drawing on registration-, graph-, CLIP-, and DINOv2-based approaches (Huang et al., 2022; Xie et al., 2023; Jeong et al., 2023; Damm et al., 2025; Lendering et al., 2026). Video methods extend AD along the time axis: some couple frame-level scoring with temporal or language context (Sultani et al., 2018; Tian et al., 2021; Ahn et al., 2026), while others replace specialized multi-stage pipelines with a single video foundation model (Orlova et al., 2025).

Across these regimes, accuracy rests on a large frozen ViT that dominates inference. Efficiency in AD has been pursued along other axes, through lighter backbones or faster reconstruction schemes (Batzner et al., 2024; Sakai et al., 2026; Khan et al., 2026), but the ViT-based detectors that now lead accuracy are left untouched: their token-level computation has rarely been reduced, and criteria imported from classification remove the wrong tokens (Section 3.1). A concurrent work, KeepAD (Hou et al., 2026b), likewise prunes tokens for AD, protecting defect evidence with prototype-guided selection and self-distillation. We share the diagnosis but keep the module training-free, reading which tokens to release from the tokens themselves.

3 Analysis

Whether a token can be pruned is decided by the task. We make this precise for AD, show that a classification-oriented criterion removes the wrong tokens, and then ask why removing the right ones can improve accuracy.

3.1 The task gap between classification and anomaly detection

Token pruning rests on one premise, that a token may leave the computation once it can no longer change the outcome. *When* that point is reached depends on the task. Classification issues a single global prediction y, so a token whose removal leaves y unchanged never needed a label of its own, and background patches typically qualify. AD assigns a label at every location, so no token is removable in that sense, and only the timing of that label differs. Let K be the index set of patch tokens, $x_i^{(l)}$ the representation of token i at layer l of an L-layer encoder, and a_i its anomaly label. We say a_i is settled as normal at layer l if $x_i^{(l)}$ already places token i inside the normal distribution and no later layer would overturn this; carrying such a token further buys nothing. Then

$$\begin{aligned} \mathcal{B} &= \{\, i \in K : y \text{ is unchanged when token } i \text{ is removed} \,\}, \\ \mathcal{N} &= \{\, i \in K : a_i \text{ is settled as normal at some } l < L \,\}. \end{aligned} \tag{1}$$

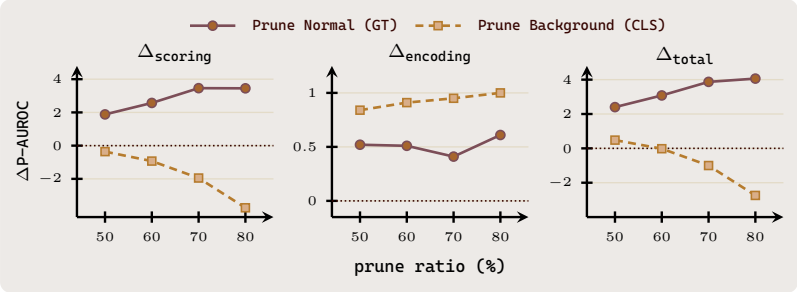


Figure 2: **Releasing the right tokens improves accuracy.** Pixel-level gain of AnomalyCLIP on VisA over the unpruned pipeline, split into Δ_{scoring} and Δ_{encoding} (Eq. 2), when releasing ground-truth normal patches or the lowest CLS-attention patches. Δ_{encoding} is positive under both criteria, while classification-based pruning loses accuracy through Δ_{scoring} as it removes more anomalous patches.

Pruning $\mathcal{B}$ deletes tokens that never needed a label, while pruning $\mathcal{N}$ releases tokens whose normality is already decided, recording zero at each released location.

Neither set is observable at inference time, so each criterion works through a proxy, attention from the CLS token for $\mathcal{B}$ and deviation from the token mean μ for $\mathcal{N}$. The proxies diverge exactly where the sets do, and the two mismatches are opposite failures. Some normal regions draw high attention because their texture or edges inform the category, even though they sit well inside the normal distribution, and attention-based pruning keeps them to the last layer, where they act as distractors and raise false positives. A small defect on a smooth surface is the reverse, since it barely affects the category and draws little attention, so attention-based pruning discards it and labels the location normal. This second error is the more damaging one, as detecting that defect is what AD is for. For the 60% pruning ratio on VisA, classification-based pruning removes 13.5% of the anomalous patches while our criterion removes 4.3% (Table S2).

3.2 Why pruning normal patches need not cost accuracy

Removing normal patches with ground-truth labels isolates the effect of releasing only normal tokens, and under it accuracy does not just hold but can rise. We split the change into two parts,

$$\Delta_{\text{total}} = \underbrace{\Delta_{\text{scoring}}}_{\geq 0 \text{ (Prop. 1)}} + \underbrace{\Delta_{\text{encoding}}}_{\text{sign unknown}}, \tag{2}$$

where Δ_{scoring} is the change that would occur if the kept tokens were encoded exactly as before, and Δ_{encoding} is the remainder from re-encoding them.

Proposition 1 *Prune only normal patches, and assume that:*

(A1) the scores of kept patches are unchanged,

(A2) the score of each pruned patch does not increase, and

(A3) the maximum score of each anomalous image lies on an anomalous patch.

Then both image- and pixel-level AUROC are non-decreasing.

The proof is in Appendix A.1: every anomalous patch keeps its score while pruned normal patches can only lose theirs, and (A3) carries this to the image level. The proposition thus assumes an ideal selector and unchanged encoding, and does not by itself justify ADoRe. Real pruning breaks (A1), since removing tokens changes how the kept tokens attend and encode, and this change is Δ_{encoding}, whose sign the proposition leaves open.

We therefore measure both terms empirically (Fig. 2). When ground-truth normal patches are pruned, Δ_{encoding} is positive alongside Δ_{scoring} across prune ratios, so the gain is not confined to the score pool: the kept anomalous tokens themselves become easier to separate once their normal

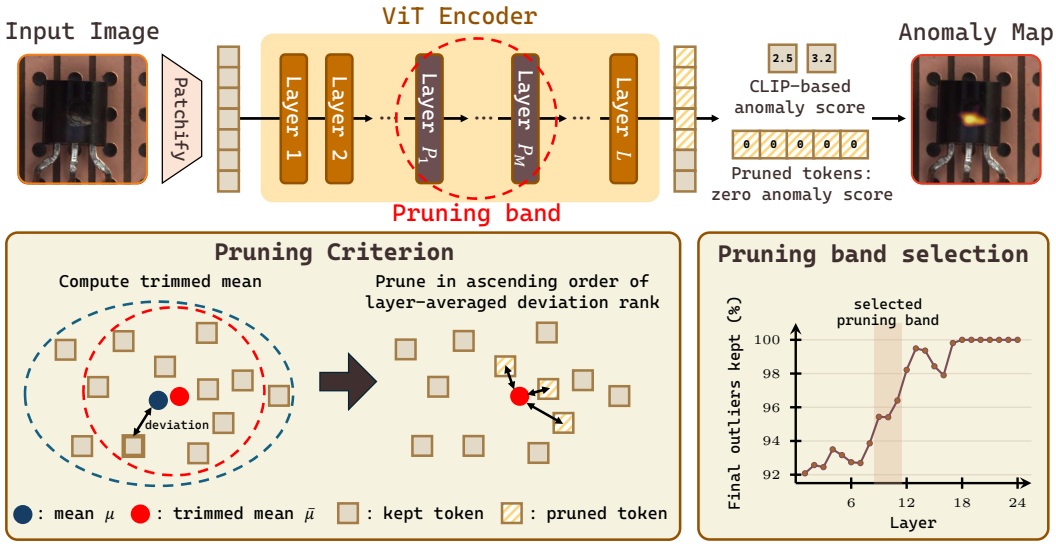


Figure 3: **Overview of ADoRe.** At each layer of a pruning band $P_1, \ldots, P_M$, every kept token is scored by its deviation from the trimmed mean $\bar{\mu}$ (Eq. 3), and the tokens ranked most normal across the band so far (Eq. 4) are pruned with an anomaly score of zero. The kept tokens are encoded to the final layer and scored by the detector as usual, and the two sets together form the anomaly map without interpolation. The band is placed where $\phi(l)$ (Eq. 5) has risen and layers remain to save. Curve shown for AnomalyCLIP on MVTec-AD.

neighbors are gone. A plausible cause is attention dilution. Self-attention tends to make tokens more alike with depth (Dong et al., 2021; Wang et al., 2022; Gong et al., 2021), and since an anomalous token attends mostly to its normal neighbors, it is pulled toward them, which shrinks the very deviation AD must detect. Removing normal tokens may weaken this pull, as Appendix A.2 illustrates for a single idealized layer. Δ_{encoding} is positive under the CLS criterion as well, so it is a gain of pruning itself, and the criterion acts through scoring.

The scoring gain is not automatic. It needs the removed tokens to be normal, and even a good criterion reverses it once pruning reaches anomalous tokens (Table S3). The criterion therefore has to find the normal tokens and spare the anomalous ones.

4 Method

We prune normal patches at intermediate layers of the ViT encoder, so that fewer tokens reach the end while the anomalous ones stay. This takes three parts: a criterion for which patches are normal (Section 4.1), where in the network to apply it (Section 4.2), and how to assemble the anomaly map once some tokens have left (Section 4.3). Fig. 3 gives an overview.

4.1 Pruning criterion

Since normal patches make up most of an image, we score each patch token by how far it deviates from the bulk of the tokens. Let $K_l \subseteq K$ be the tokens kept at layer l. Their mean $\mu^{(l)}$ is pulled toward the anomalous ones, increasingly so as an anomalous region grows. We therefore drop the τ fraction of K_l with the largest $1 - \cos(x_i^{(l)}, \mu^{(l)})$, and use the trimmed mean $\bar{\mu}^{(l)}$ of the remaining set K_l' as the reference:

$$\bar{\mu}^{(l)} = \frac{1}{|K_l'|} \sum_{j \in K_l'} x_j^{(l)}, \qquad s_i^{(l)} = 1 - \cos\left(x_i^{(l)}, \bar{\mu}^{(l)}\right). \tag{3}$$

We call $s_i^{(l)}$ the deviation of token i at layer l. For video, $\bar{\mu}^{(l)}$ is instead computed from the tokens of earlier frames in the clip, so that an anomaly in the current frame is compared against a reference that precedes it.

Pruning is applied over a band of M layers $P_1 < \cdots < P_M$. Since a pruned token never returns (Section 4.3), it should leave only once its normality is settled (Section 3.1), and a deviation read at a single layer does not guarantee this, least of all at P_1, where $\bar{\mu}^{(P_1)}$ is computed before any pruning has cleaned the token set. We therefore accumulate evidence over the band. Deviations are not comparable across layers, so we rank them within each layer and average the ranks:

$$r_i^{(l)} = \frac{\mathrm{rank}\big(s_i^{(l)}\big)}{|K_l| - 1}, \qquad R_i^{(P_m)} = \frac{1}{m}\sum_{k=1}^{m} r_i^{(P_k)}, \tag{4}$$

where rank counts from 0 in ascending order of deviation within K_l, so that $r_i^{(l)} \in [0, 1]$. At each band layer P_m, we keep the $\lfloor \rho\, |K_{P_m}| \rfloor$ tokens with the largest $R_i^{(P_m)}$ and prune the rest, where ρ is the per-layer keep ratio.

4.2 Where to prune

A pruned token receives an anomaly score of zero, so we prune only at depths where the deviation ranking is already reliable. We locate these depths once per backbone on normal images, running the encoder without pruning. Calibration uses 100 normal images from a dataset other than the one evaluated.

For each layer l, let E_l be the $|K| - \lfloor \rho |K| \rfloor$ tokens with the smallest $s^{(l)}$, i.e., the tokens a single cut at l would prune, and let $D_L = K \setminus E_L$ be the most-deviating tokens at the final layer. The preservation

$$\phi(l) = \frac{|\, D_L \setminus E_l \,|}{|\, D_L \,|} \tag{5}$$

is the fraction of D_L that a cut at l would keep.

$\phi(l)$ is low in the early layers and rises through the middle of the network as the ranking settles (Fig. 3). We place the band where $\phi(l)$ has risen but enough layers remain after it to save compute. An ablation at matched compute confirms this choice (Table 4b): shallower bands lose pixel accuracy, and deeper ones must prune far more aggressively to reach the same budget.

Moreover, we spread the cut over $M = 3$ band layers rather than making it at once. This helps the criterion twice: the clearest normal tokens leave first, so $\bar{\mu}$ at later band layers is computed from an already cleaned set, and Eq. 4 accumulates over more than one layer. The band is fixed per backbone across datasets, and ρ is left to the user as a compute budget, pruning a fraction $p \approx 1 - \rho^M$ of the patch tokens in total.

4.3 Anomaly map construction

Pruned tokens reach no final layer, so their locations have no native anomaly score. The usual fix interpolates these from kept neighbors, which assumes a pruned location resembles whatever was kept nearby. We instead use the verdict already made: the band judged a pruned token normal, so its location is assigned an anomaly score of zero, while kept locations take the detector's native score. The map is thus assembled in a single pass without interpolation.

5 Experiments

5.1 Experimental setup

Datasets and metrics. We evaluate on three groups of benchmarks. For industrial AD, we use MVTec-AD (Bergmann et al., 2019) and VisA (Zou et al., 2022), reporting image-level AUROC and AP together with pixel-level AUROC and AUPRO. For medical AD, we use HeadCT (Salehi et al., 2021), BrainMRI (Salehi et al., 2021), and Br35H (Hamada), reporting image-level AUROC and AP. For video AD, we use CUHK Avenue (Lu et al., 2013) and ShanghaiTech (Liu et al., 2018), reporting micro- and macro-averaged frame-level AUROC.

Table 1: **Token reduction performance on industrial domain.** Image and Pixel results are reported as (AUROC, AP) and (AUROC, AUPRO), respectively. *Lat* is ViT encoding latency; *Speed.* is the latency-based speedup over the unpruned model. MVTec-AD is reported at 50, 70%, and VisA is reported at 60, 80%. ADoRe has the lowest measured encoder latency among the compared methods.

Method	MVTec-AD					VisA				
	Lat.(ms)↓	GFLOPs↓	Speed.↑	Image↑	Pixel↑	Lat.(ms)↓	GFLOPs↓	Speed.↑	Image↑	Pixel↑
AnomalyCLIP	80.0	1212.6	1.00×	(91.6, 96.4)	(88.2, 80.5)	80.0	1212.6	1.00×	(82.0, 85.3)	(93.9, 83.6)
	Prune **50%** of Tokens					Prune **60%** of Tokens				
+ EViT	43.9	811.1	1.82×	(91.6, 96.3)	(82.2, 77.2)	39.9	748.0	2.01×	(82.4, 86.1)	(90.8, 79.7)
+ ToMe	43.8	790.8	1.83×	(88.9, 95.1)	(87.6, 79.6)	39.6	726.9	2.02×	(81.9, 85.5)	(94.4, 85.3)
+ ATS	60.7	811.6	1.32×	(91.2, 96.1)	(88.1, 81.2)	56.5	716.4	1.42×	(82.8, 86.3)	(94.1, 83.0)
+ GTP-ViT	61.3	797.4	1.31×	(90.9, 96.0)	(87.3, 79.0)	56.8	714.5	1.41×	(82.4, 86.1)	(90.8, 73.4)
+ ADoRe (Ours)	**43.2**	**784.6**	**1.85×**	(**92.7**, **96.8**)	(**89.7**, **82.4**)	**39.1**	**709.2**	**2.05×**	(**83.7**, **86.9**)	(**95.8**, **85.9**)
	Prune **70%** of Tokens					Prune **80%** of Tokens				
+ EViT	39.8	690.8	2.01×	(91.2, 95.9)	(78.2, 70.5)	34.4	640.3	2.33×	(82.1, 85.5)	(87.7, 75.1)
+ ToMe	38.1	662.2	2.10×	(86.4, 93.8)	(84.8, 71.9)	33.9	610.5	2.36×	(67.5, 72.8)	(88.4, 64.4)
+ ATS	53.1	643.5	1.51×	(90.8, 95.8)	(85.5, 73.7)	49.1	586.9	1.63×	(82.5, 86.2)	(92.4, 78.8)
+ GTP-ViT	54.2	640.1	1.48×	(75.7, 88.2)	(76.2, 54.5)	47.2	577.5	1.69×	(54.3, 62.1)	(86.8, 63.8)
+ ADoRe (Ours)	**36.3**	**637.6**	**2.20×**	(**92.1**, **96.7**)	(**88.6**, **81.6**)	**32.4**	**569.5**	**2.47×**	(**84.0**, **86.9**)	(**94.7**, **86.3**)
FAPrompt	80.0	1212.6	1.00×	(91.3, 95.5)	(89.4, 85.1)	80.0	1212.6	1.00×	(83.4, 86.1)	(95.3, 87.9)
	Prune **50%** of Tokens					Prune **60%** of Tokens				
+ EViT	43.9	811.1	1.82×	(90.7, 95.1)	(78.7, 73.3)	39.9	748.0	2.01×	(84.5, 86.8)	(85.1, 70.3)
+ ToMe	43.8	790.8	1.83×	(88.8, 94.1)	(83.4, 80.5)	39.6	726.9	2.02×	(83.7, 86.4)	(92.9, 84.5)
+ ATS	60.7	811.6	1.32×	(90.5, 95.0)	(82.9, 78.2)	56.5	716.4	1.42×	(84.7, 86.9)	(90.6, 77.1)
+ GTP-ViT	61.3	797.4	1.31×	(90.3, 94.9)	(82.2, 76.6)	56.8	714.5	1.41×	(81.2, 85.0)	(83.1, 62.8)
+ ADoRe (Ours)	**43.2**	**784.6**	**1.85×**	(**92.1**, **96.0**)	(**88.2**, **84.1**)	**39.1**	**709.2**	**2.05×**	(**84.8**, **87.0**)	(**95.8**, **88.7**)
	Prune **70%** of Tokens					Prune **80%** of Tokens				
+ EViT	39.8	690.8	2.01×	(89.9, 94.6)	(72.3, 60.6)	34.4	640.3	2.33×	(84.2, 86.1)	(78.0, 57.5)
+ ToMe	38.1	662.2	2.10×	(86.1, 92.5)	(79.1, 68.6)	33.9	610.5	2.36×	(70.8, 74.8)	(80.5, 53.9)
+ ATS	53.1	643.5	1.51×	(88.7, 94.2)	(78.4, 69.4)	49.1	586.9	1.63×	(83.9, 86.4)	(86.3, 70.8)
+ GTP-ViT	54.2	640.1	1.48×	(72.5, 86.4)	(57.8, 18.2)	47.2	577.5	1.69×	(55.5, 63.2)	(61.2, 15.9)
+ ADoRe (Ours)	**36.3**	**637.6**	**2.20×**	(**91.0**, **95.5**)	(**86.3**, **82.5**)	**32.4**	**569.5**	**2.47×**	(**84.8**, **86.6**)	(**94.0**, **88.3**)
SubspaceAD	247.1	6540.0	1.00×	(96.3, 98.4)	(97.4, 94.4)	247.1	6540.0	1.00×	(92.9, 92.5)	(98.2, 95.2)
	Prune **50%** of Tokens					Prune **60%** of Tokens				
+ EViT	174.5	4434.1	1.42×	(96.2, 98.4)	(96.4, 93.8)	154.6	4039.7	1.60×	(**93.8**, 93.6)	(97.9, 95.3)
+ ToMe	173.3	4489.6	1.43×	(96.1, 98.3)	(**97.4**, **94.4**)	153.5	4033.9	1.61×	(92.7, 92.8)	(98.1, 94.7)
+ ATS	199.7	4394.8	1.24×	(95.5, 97.9)	(96.9, 93.8)	185.3	4164.0	1.33×	(89.0, 88.7)	(97.8, 94.0)
+ GTP-ViT	310.2	4374.2	0.80×	(95.7, 98.2)	(97.0, 93.8)	266.8	4033.9	0.93×	(90.7, 90.4)	(97.9, 94.3)
+ ADoRe (Ours)	**172.5**	**4370.3**	**1.43×**	(**96.6**, **98.7**)	(97.2, **94.4**)	**152.9**	**3978.0**	**1.62×**	(**93.8**, **94.0**)	(**98.5**, **95.7**)
	Prune **70%** of Tokens					Prune **80%** of Tokens				
+ EViT	138.0	3622.2	1.79×	(96.2, 98.4)	(95.5, 92.6)	128.9	3309.3	1.92×	(93.4, 93.3)	(96.6, 93.3)
+ ToMe	134.7	3648.5	1.83×	(95.3, 97.7)	(**97.3**, **94.3**)	130.4	3380.7	1.89×	(92.6, 92.6)	(**98.0**, 94.6)
+ ATS	170.1	3611.3	1.45×	(93.1, 96.7)	(96.2, 92.7)	164.1	3326.0	1.51×	(82.8, 82.1)	(96.6, 90.3)
+ GTP-ViT	253.0	3757.4	0.98×	(95.2, 97.9)	(96.8, 93.4)	248.5	3540.6	0.99×	(89.9, 89.5)	(97.8, 94.0)
+ ADoRe (Ours)	**133.7**	**3596.6**	**1.85×**	(**96.5**, **98.6**)	(96.8, 93.9)	**128.1**	**3228.0**	**1.93×**	(**93.8**, **94.0**)	(96.9, **94.7**)

Baselines. We attach our module to detectors that span supervision, backbone, and modality: two zero-shot CLIP detectors (AnomalyCLIP (Zhou et al., 2024), FAPrompt (Zhu et al., 2025)), a few-shot DINOv2 detector (SubspaceAD (Lendering et al., 2026)), and a video detector (AI-VAD (Reiss & Hoshen, 2025)). On each we compare against four training-free token reduction baselines: EViT (Liang et al., 2022), ToMe (Bolya et al., 2023), ATS (Fayyaz et al., 2022), and GTP-ViT (Xu et al., 2024). Each baseline fills the locations it removes by its own mechanism: EViT by a single attention-weighted token pooled from the removed set, ToMe by the merged representative that absorbed the location, and ATS and GTP-ViT by the nearest surviving token.

5.2 Results

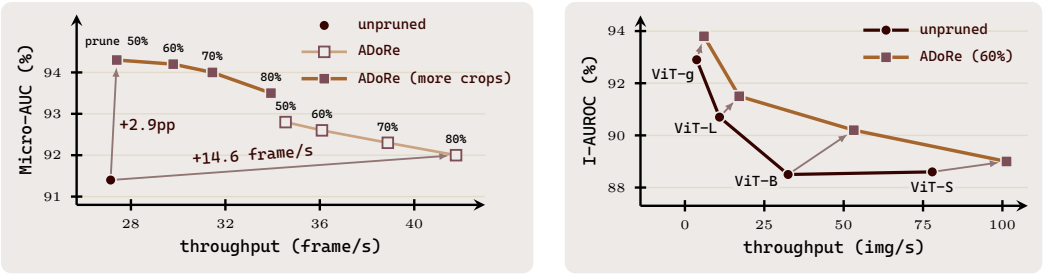
Table 1 reports accuracy at matched compute. Our criterion matches or exceeds the unpruned model on most metrics while accelerating the encoder by up to 2.47×. Existing reductions instead pay for their speed, and some collapse at high ratios. The medical datasets show the same trend (Table 2). Beyond CLIP zero-shot models, the same rule applied to SubspaceAD (DINOv2, 1-shot) and AI-VAD (video) raises detection accuracy on most metrics (Tables 1 and 3). For video, the reference comes from earlier frames rather than from the frame being judged (Section 4.1), so an anomaly growing in the current frame contaminates it less.

Table 2: **Token reduction performance on medical domain.** Each dataset result is reported as (I-AUROC, I-AP).

Method	Prune **50%** of Tokens					Prune **70%** of Tokens				
	Lat.(ms)↓	Speed.↑	HeadCT	BrainMRI	Br35H	Lat.(ms)↓	Speed.↑	HeadCT	BrainMRI	Br35H
AnomalyCLIP	80.0	1.00×	(93.1, 91.1)	(90.0, 92.1)	(94.2, 94.2)	80.0	1.00×	(93.1, 91.1)	(90.0, 92.1)	(94.2, 94.2)
+ EViT	43.9	1.82×	(92.3, 88.1)	(89.2, 90.4)	(93.6, 93.0)	39.8	2.01×	(92.1, 87.9)	(88.7, 89.9)	(93.2, 92.5)
+ ToMe	43.8	1.83×	(90.0, 86.0)	(90.5, 93.2)	(94.5, 94.4)	38.1	2.10×	(87.9, 87.1)	(88.6, 91.2)	(94.0, 93.4)
+ ATS	60.7	1.32×	(92.2, 89.2)	(89.8, 91.2)	(93.8, 93.5)	53.1	1.51×	(92.5, 88.6)	(88.9, 89.8)	(93.4, 92.9)
+ GTP-ViT	61.3	1.31×	(92.2, 89.5)	(90.3, 91.3)	(93.9, 93.4)	54.2	1.48×	(73.5, 68.9)	(89.8, 90.5)	(93.6, 92.8)
+ ADoRe (Ours)	**43.2**	**1.85×**	(**94.8**, **95.2**)	(**95.4**, **96.4**)	(**97.4**, **97.2**)	**36.3**	**2.20×**	(**92.9**, **92.4**)	(**93.5**, **93.8**)	(**96.1**, **96.1**)
FAPrompt	80.0	1.00×	(93.9, 92.2)	(94.8, 94.0)	(96.5, 95.5)	80.0	1.00×	(93.9, 92.2)	(94.8, 94.0)	(96.5, 95.5)
+ EViT	43.9	1.82×	(92.8, 89.2)	(94.3, 93.2)	(95.5, 94.3)	39.8	2.01×	(91.6, 88.2)	(93.7, 93.2)	(94.7, 93.3)
+ ToMe	43.8	1.83×	(88.4, 83.3)	(95.0, 93.6)	(95.9, 94.4)	38.1	2.10×	(84.3, 80.5)	(87.0, 90.2)	(88.4, 88.0)
+ ATS	60.7	1.32×	(93.1, 89.9)	(94.6, 93.6)	(95.5, 94.4)	53.1	1.51×	(90.6, 87.6)	(93.0, 91.4)	(93.9, 92.4)
+ GTP-ViT	61.3	1.31×	(93.1, 90.6)	(93.9, 92.2)	(95.2, 94.0)	54.2	1.48×	(66.6, 67.2)	(59.1, 72.6)	(53.3, 57.2)
+ ADoRe (Ours)	**43.2**	**1.85×**	(**94.5**, **95.1**)	(**96.2**, **97.2**)	(**97.1**, **97.4**)	**36.3**	**2.20×**	(**91.7**, **92.4**)	(**95.3**, **96.4**)	(**95.9**, **96.3**)

Table 3: **Token reduction performance on video anomaly detection.** *Lat* is ViT encoding latency per crop. GTP-ViT is omitted because it did not reduce latency. Measured with batch size 1.

Method	Avenue					ShanghaiTech				
	Lat.(ms)↓	GFLOPs↓	Speed.↑	Micro-AUC↑	Macro-AUC↑	Lat.(ms)↓	GFLOPs↓	Speed.↑	Micro-AUC↑	Macro-AUC↑
AI-VAD	11.8	381.9	1.00×	91.4	94.4	11.8	381.9	1.00×	85.2	90.0
	Prune **50%** of Tokens					Prune **60%** of Tokens				
+ EViT	8.52	294.1	1.38×	90.8	93.5	7.52	267.2	1.57×	85.1	89.7
+ ToMe	8.67	302.7	1.36×	91.4	94.7	7.79	278.5	1.51×	85.1	89.6
+ ATS	8.72	283.7	1.35×	90.9	93.7	7.87	261.8	1.50×	85.2	89.8
+ ADoRe (Ours)	**8.44**	**283.5**	**1.40×**	**92.8**	**95.4**	**7.47**	**261.4**	**1.58×**	**85.6**	**90.2**
	Prune **70%** of Tokens					Prune **80%** of Tokens				
+ EViT	7.17	253.8	1.65×	90.7	93.3	6.61	236.0	1.79×	85.1	89.6
+ ToMe	7.37	261.4	1.60×	90.7	**94.5**	6.74	238.7	1.75×	85.0	89.1
+ ATS	7.29	240.3	1.62×	90.7	94.2	6.69	220.8	1.76×	85.1	89.7
+ ADoRe (Ours)	**7.02**	**240.0**	**1.68×**	**92.3**	94.2	**6.39**	**218.3**	**1.85×**	**85.8**	**90.1**



(a) More crops: spend savings on extra crops.

(b) Larger backbone: spend savings on a bigger ViT.

Figure 4: **Reinvesting the savings at a fixed budget.** (a) Pruning at each ratio (percentages) with extra context crops on the Avenue dataset. A *crop* is the image region of a detected object that AI-VAD feeds to its ViT appearance encoder, and a *context crop* is an additional crop taken for the largest objects and fused into their appearance score. ADoRe (more crops) spends the time saved by pruning on these extra crops. Measured with batch size 32. (b) Pruning each DINOv2 backbone against the unpruned backbones of increasing size, on the VisA dataset.

5.3 Reinvesting the savings at a fixed budget

Pruning returns part of the encoder's budget, and that budget can be spent again where it buys accuracy. We hold latency fixed and reinvest along the axis each detector naturally scales on: backbone size for SubspaceAD, the only detector here released at several sizes, and the number of crops per frame for AI-VAD.

More crops. AI-VAD encodes each detected object as a crop, so what pruning saves can instead buy more views per frame. We add a context crop for the largest objects and fuse it into their appearance score with one setting fixed in advance and applied unchanged at every pruning ratio. As

(a) Pruning criterion

Accum.	Trim	Image↑	Pixel↑
✓	✓	(**84.0**, **86.9**)	(94.7, **86.3**)
✓	✗	(83.8, 86.8)	(**94.8**, 86.2)
✗	✓	(83.4, 86.1)	(94.0, 84.2)
Random		(70.6, 75.0)	(85.6, 61.8)

(b) Pruning band (matched compute)

		Center depth (%), width 3				Band width (layers)			
Detector	Data	17 / 15	29 / 28	42 / 40	54 / 52	1	2	3	4
AnomalyCLIP	VisA	-40.2	-2.0	**1.6**	-5.8	0.4	1.5	**1.6**	0.9
	MVTec-AD	-24.3	-7.2	**0.4**	-16.3	-4.0	-3.3	**0.4**	-2.3
SubspaceAD	VisA	-6.9	-0.3	**0.02**	-1.7	-3.0	-1.1	**0.02**	-0.01
	MVTec-AD	-14.8	-1.3	**-0.6**	-3.1	-1.6	-1.1	-0.6	**-0.5**

Table 4: **Ablations on the pruning criterion and the pruning band. (a)** AnomalyCLIP at 80% pruning on VisA. Image and Pixel are (AUROC, AP) and (AUROC, AUPRO). Accum. is the rank accumulation of Eq. 4 and Trim the trimmed reference of Eq. 3; Random replaces the selection entirely. **(b)** Δ P-AUROC (pp) against the unpruned pipeline at the compute of the default setting (prune 70%). Center depth is given as CLIP / DINOv2, and bold marks the best value per row within each block. Shading marks the setting we use.

shown in Fig. 4a, the reinvested pipeline runs no slower than the unpruned one and exceeds it by 2.9pp in Micro-AUC.

Larger backbone. Shrinking the encoder is the other way to spend less on it, and the two are not the same trade. A smaller backbone gives up representation, while pruning gives up patches whose verdict is already settled. Fig. 4b places both on the plane of accuracy against throughput, and the pruned curve lies above the unpruned one throughout. Interpolating the unpruned curve at the throughput of each pruned point, a pruned larger backbone gains about 1.7pp of image AUROC over the unpruned backbone that runs at the same speed.

5.4 Ablation study

Pruning criterion. Table 4a keeps our schedule and map construction fixed and varies only how tokens are chosen. Our criterion is ahead on every metric but one. Accumulating the verdict over the band accounts for most of the margin, and the gap is widest in localization. The trimmed reference adds a smaller amount on top. Random selection prunes the same number of tokens at the same layers and falls far behind, so the gain comes from which tokens are pruned rather than from releasing at these layers.

Where and how wide to cut. Table 4b sweeps the pruning band at a matched compute budget. Cutting shallower gives up pixel accuracy, because the scores patches leave on are not yet reliable; cutting deeper has to prune far more aggressively to reach the budget, because the layers ahead of the band already run in full. Between these the choice is flat, and the band chosen by $\phi(l)$ lies inside it. Splitting the cut across the band matters less than where the band sits, but a single cut still falls behind, since it leaves the reference uncleaned and gives Eq. 4 only one layer to read.

6 Conclusion

We set out to lighten the ViT encoder that dominates inference in anomaly detection. Which tokens may leave it is decided by the task. The tokens that can be released are not the ones a classification criterion would drop, since little influence on a global prediction does not make a patch uninformative about its own location. What can be released is a patch whose normality is already settled. This distinction explains why classification-oriented token reduction discards subtle defects while retaining normal yet salient regions, and it leads to ADoRe, a training-free module that accumulates a normality verdict across intermediate layers, releases the tokens those layers agree on, and records the verdict directly in the anomaly map. Across zero-shot and few-shot detectors, CLIP and DINOv2 backbones, and image and video benchmarks, ADoRe reduces encoder computation while remaining comparable to or better than the unpruned models. The ground-truth analysis suggests that releasing truly normal tokens need not cost accuracy, and ADoRe retains much of this benefit in practice. At a fixed time budget, reinvesting the savings in a larger encoder or more crops surpasses the unpruned pipeline.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing, to draft sections of the paper, and to assist with proving mathematical claims. We have not used generative AI tools for research ideation. Additionally, we used generative AI tools for retrieval and discovery of related work. All AI-assisted work was reviewed by the authors: drafted text was rewritten and verified against our own results, and retrieved references were read against the original papers. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work accelerates anomaly detection and does not introduce a new detector or a new data source. All experiments use publicly available benchmarks for industrial, medical, and video anomaly detection under their original licenses and intended research use, and no new data were collected. No human subjects were involved in this study.

REPRODUCIBILITY STATEMENT

The proof of Proposition 1 and the single-layer analysis behind the encoding term are given in Appendix A. Pruning layers and ratios, the trim fraction, the accumulation rule, detector configurations, and the settings used to match each baseline to the same compute are listed in Appendix B, together with the datasets, splits, and evaluation metrics.

REFERENCES

Sunghyun Ahn, Youngwan Jo, Kijung Lee, Sein Kwon, Inpyo Hong, and Sanghyun Park. Anyanomaly: Zero-shot customizable video anomaly detection with lvlm. In *WACV*, 2026.

Kilian Batzner, Lars Heckler, and Rebecca König. Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In *WACV*, 2024.

Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, 2019.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *ICLR*, 2023.

Yunkang Cao, Jiangning Zhang, Luca Frittoli, Yuqi Cheng, Weiming Shen, and Giacomo Boracchi. Adaclip: Adapting clip with hybrid learnable prompts for zero-shot anomaly detection. In *ECCV*, 2024.

Xuhai Chen, Jiangning Zhang, Guanzhong Tian, Haoyang He, Wuhao Zhang, Yabiao Wang, Chengjie Wang, and Yong Liu. Clip-ad: A language-guided staged dual-path model for zero-shot anomaly detection. In *Human Activity Recognition and Anomaly Detection*, 2025.

Simon Damm, Mike Laszkiewicz, Johannes Lederer, and Asja Fischer. Anomalydino: Boosting patch-based few-shot anomaly detection with dinov2. In *WACV*, 2025.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *ICML*, 2021.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *ECCV*, 2022.

Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. Anomalyvfm–transforming vision foundation models into zero-shot anomaly detectors. In *CVPR*, 2026.

Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Vision transformers with patch diversification. *arXiv preprint arXiv:2104.12753*, 2021.

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Hao Li, Ming Tang, and Jinqiao Wang. Filo: Zero-shot anomaly detection by fine-grained description and high-quality localization. In *ACM MM*, 2024.

Ahmed Hamada. Br35h :: Brain tumor detection 2020. https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection. Accessed: 2026-09-07.

Yanning Hou, Peiyuan Li, Zirui Liu, Yitong Wang, Yanran Ruan, Jianfeng Qiu, and Ke Xu. Visualad: Language-free zero-shot anomaly detection via vision transformer. In *CVPR*, 2026a.

Yanning Hou, Jingyuan Zhang, Xiaoyun Wang, Qixiang Ma, Sihang Zhou, and Ke Xu. Keep the needle, prune the haystack: Defect-preserving token pruning for efficient zero-shot anomaly detection. *arXiv preprint arXiv:2608.03681*, 2026b.

Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In *ECCV*, 2022.

Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation. In *CVPR*, 2023.

Uzair Khan, Luigi Capogrosso, Muhammad Aqeel, Francesco Setti, Michele Magno, and Marco Cristani. Lizad: A lightweight zero-shot anomaly detection framework for industrial manufacturing. *arXiv preprint arXiv:2607.01949*, 2026.

Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *ECCV*, 2022.

Camile Lendering, Erkut Akdag, and Egor Bondarau. Subspacead: Training-free few-shot anomaly detection via subspace modeling. In *CVPR*, 2026.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *ICLR*, 2022.

Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection-a new baseline. In *CVPR*, 2018.

Yifei Liu, Mathias Gehrig, Nico Messikommer, Marco Cannici, and Davide Scaramuzza. Revisiting token pruning for object detection and instance segmentation. In *WACV*, 2024.

Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *ICCV*, 2013.

Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *CVPR*, 2022.

Svetlana Orlova, Tommie Kerssies, Brunó B Englert, and Gijs Dubbelman. Simplifying traffic anomaly detection with video foundation models. In *ICCVW*, 2025.

Zhen Qu, Xian Tao, Mukesh Prasad, Fei Shen, Zhengtao Zhang, Xinyi Gong, and Guiguang Ding. Vcp-clip: A visual context prompting model for zero-shot anomaly segmentation. In *ECCV*, 2024.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, 2021.

Tal Reiss and Yedid Hoshen. An attribute-based method for video anomaly detection. In *TMLR*, 2025.

Shunsuke Sakai, Xiangteng He, Chunzhi Gu, Leonid Sigal, and Tatsuhito Hasegawa. Invad: Inversion-based reconstruction-free anomaly detection with diffusion models. In *CVPR*, 2026.

Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In *CVPR*, 2021.

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

Quan Tang, Bowen Zhang, Jiajun Liu, Fagui Liu, and Yifan Liu. Dynamic token pruning in plain vision transformers for semantic segmentation. In *ICCV*, 2023.

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*, 2021.

Hongjie Wang, Bhishma Dedhia, and Niraj K Jha. Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. In *CVPR*, 2024.

Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In *ICLR*, 2022.

Guoyang Xie, Jinbao Wang, Jiaqi Liu, Feng Zheng, and Yaochu Jin. Pushing the limits of fewshot anomaly detection in industry vision: Graphcore. In *ICLR*, 2023.

Xuwei Xu, Sen Wang, Yudong Chen, Yanping Zheng, Zhewei Wei, and Jiajun Liu. Gtp-vit: Efficient vision transformers via graph-based token propagation. In *WACV*, 2024.

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *AAAI*, 2022.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *CVPR*, 2022.

Fanhu Zeng, Deli Yu, Zhenglun Kong, and Hao Tang. Token transforming: A unified and training-free token compression framework for vision transformer acceleration. *arXiv preprint arXiv:2506.05709*, 2025.

Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. In *ICLR*, 2024.

Jiawen Zhu, Yew-Soon Ong, Chunhua Shen, and Guansong Pang. Fine-grained abnormality prompt learning for zero-shot anomaly detection. In *ICCV*, 2025.

Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *ECCV*, 2022.

Overview of the Appendix

A. Proofs

- Proposition 1: when releasing normal patches cannot lower AUROC, AP or AUPRO
- A single-layer account of attention dilution

B. Implementation Details

- Pruning layers and ratios, and how each baseline is matched to the same compute
- AD models, baselines, datasets, and metrics

C–D. More Experiments and Discussion

- How many anomalous patches each criterion removes, and the split it actually yields
- What the trimmed reference and the zero-filled map each contribute
- Measured latency, where the time goes, and speedup across batch sizes
- The spread across reference draws in the few-shot setting
- How accuracy and leakage change with defect area, and the limit of a fixed budget

Figure 5: **Overview of the appendix.** The appendix contains proofs, implementation details, and additional experimental results.

A PROOFS

A.1 PROOF OF PROPOSITION 1

The setting is the ground-truth normal pruning of Section 3.2. A fixed set of normal patches is pruned using ground-truth labels, each pruned location is assigned a value, and every kept location takes the score the detector gives it. We treat each patch as one location, *i.e.*, we work at patch resolution.

Setup and notation. Let $\mathcal{A}$ be the set of ground-truth anomalous locations and $\mathcal{A}^c$ the normal ones. Let z_i be the anomaly score at location i under the unpruned model, and z'_i the score after pruning. The proposition assumes:

(A1) $z'_i = z_i$ for every kept location i. Since only normal patches are pruned, this includes every $i \in \mathcal{A}$.

(A2) $z'_i \leq z_i$ for every pruned location i. Assigning zero, as ADoRe does, satisfies this whenever scores are non-negative.

(A3) In each anomalous image, the maximum of z_i is attained at a location in $\mathcal{A}$.

Pixel-level AUROC. Pixel-level AUROC is the probability that a randomly chosen anomalous location scores above a randomly chosen normal one,

$$\mathrm{AUROC} = \frac{1}{|\mathcal{A}|\,|\mathcal{A}^c|} \sum_{a\in\mathcal{A}} \sum_{n\in\mathcal{A}^c} \Big[\mathbf{1}\{z_a > z_n\} + \tfrac{1}{2}\,\mathbf{1}\{z_a = z_n\}\Big]. \tag{S1}$$

Fix a pair $(a, n) \in \mathcal{A} \times \mathcal{A}^c$. By (A1), $z'_a = z_a$. If n is kept, $z'_n = z_n$ by (A1) and the pair contributes as before. If n is pruned, $z'_n \leq z_n$ by (A2), so the comparison can only move in favor of a and the summand cannot decrease. Every summand in Eq. S1 is therefore non-decreasing, and so is the pixel-level AUROC. The argument holds within each category, and hence for the category average we report.

Image-level AUROC. The image score is $Z = \max_i z_i$. In a normal image every location is normal, so by (A1) and (A2) no score rises and $Z' \leq Z$. In an anomalous image, (A3) places the maximum at some $a \in \mathcal{A}$, which keeps its score by (A1), so $Z' \geq z_a = Z$. Anomalous image scores thus do not fall and normal ones do not rise, and the pairwise argument of Eq. S1 shows that image-level AUROC is non-decreasing. For an image score computed from a global embedding instead, Δ_{scoring} is zero at the image level, and any change there is Δ_{encoding}.

AP and AUPRO. The same assumptions extend the result to AP and AUPRO. By (A1), the anomalous side is unchanged at every threshold t, and by (A1) and (A2), the number of normal locations scoring at least t does not increase. Writing

$$\mathrm{AP} = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} P(z_a), \qquad P(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t) + \mathrm{FP}(t)}, \tag{S2}$$

every term of Eq. S2 is therefore non-decreasing. For AUPRO, the per-region overlap at t depends only on anomalous locations and is unchanged, while $\mathrm{FPR}(t)$ does not increase, so every point of the PRO–FPR curve moves left at a fixed height. As the curve is non-decreasing, its area up to $\mathrm{FPR} = 0.3$ cannot decrease under either step or linear interpolation. At the image level, (A1)–(A3) leave every anomalous image score unchanged, since $Z' \geq z_a = Z$ and no score rises, and raise no normal one, so image AP follows by the same argument.

The result holds by construction: (A1) fixes the anomalous side of every comparison, and (A2) can only lower the normal side. It covers $\Delta_{\mathrm{scoring}}$ alone and says nothing about how pruning changes the representations of the kept tokens, which is the term $\Delta_{\mathrm{encoding}}$ discussed next.

A.2 Attention dilution under normal-token removal

Proposition 1 covers $\Delta_{\mathrm{scoring}}$, with the kept tokens encoded as before. For the remaining term $\Delta_{\mathrm{encoding}}$, we use an idealized single-layer calculation to illustrate how removing normal tokens can sharpen the anomalous signal.

Model. Consider one self-attention layer with m tokens, one anomalous and $m-1$ normal. Normal tokens share a value v_n and the anomalous token has value v_a. We write $\delta = v_a - v_n$, assume $\delta \perp v_n$, and set $\kappa = \|\delta\|/\|v_n\|$. Attention logits are c for self and d for cross, and we assume $c > d$, so that $r = e^{c-d} > 1$. Each token carries independent noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2 \|v_n\|^2 I)$, so that σ is on the same scale as κ.

Signal. The anomalous token attends to itself with weight

$$\alpha_{aa}(m) = \frac{e^c}{e^c + (m-1)e^d} = \frac{r}{r + m - 1}, \tag{S3}$$

which grows as tokens are removed. Its output is $x_a = v_n + \alpha_{aa}\,\delta$, while a normal token, which places weight $1/(r+m-1)$ on the anomalous one, outputs $x_n = v_n + \delta/(r+m-1)$. Let θ be the angle between an output and v_n. Since $\delta \perp v_n$, $\tan\theta = \|x - v_n\|/\|v_n\|$ measures how far an output deviates from v_n, and the gap between the two outputs is

$$g(m) = \tan\theta_a - \tan\theta_n = \frac{(r-1)\,\kappa}{r + m - 1}, \tag{S4}$$

which also grows as m shrinks.

Noise. We score an output by its component along δ, $\langle x - v_n, \delta/\|\delta\| \rangle / \|v_n\|$, which equals $\tan\theta$ without noise. Fewer tokens also mean less averaging. In units of σ^2, the noise variance of the anomalous output is

$$V(m) = \alpha_{aa}^2 + \frac{(1-\alpha_{aa})^2}{m-1} = \frac{r^2 + m - 1}{(r+m-1)^2}, \tag{S5}$$

which a normal output shares, and which grows as m shrinks as well. Signal and noise rise together, so $g(m)$ alone does not decide the sign.

Signal-to-noise ratio. In the ratio the factor $r + m - 1$ cancels,

$$\mathrm{SNR}(m) = \frac{g(m)}{\sigma\sqrt{V(m)}} = \frac{(r-1)\kappa}{\sigma\sqrt{r^2 + m - 1}}, \tag{S6}$$

so $\mathrm{SNR}(m') > \mathrm{SNR}(m)$ whenever $m' < m$. Treating detection as separating an anomalous output from a normal output of another image, whose noise is independent, pixel-level AUROC is

$$\mathrm{AUROC}(m) = \Phi\left(\frac{\mathrm{SNR}(m)}{\sqrt{2}}\right) = \Phi\left(\frac{(r-1)\kappa}{\sigma\sqrt{2(r^2 + m - 1)}}\right), \tag{S7}$$

and since Φ is increasing, the encoding term $\mathrm{AUROC}(m') - \mathrm{AUROC}(m)$ is positive in this model.

Limitations. The calculation covers a single layer and does not show whether the gain accumulates across depth, where deviations can also contract (Dong et al., 2021). It assumes one anomalous token and two values, and so ignores larger or spatially contiguous anomalies. Within one image the outputs share noise terms, which the expression ignores; it describes pairs drawn from different images, which make up most pairs in pixel-level AUROC. We therefore read the result only as a mechanism, and Fig. 2 measures Δ_{encoding} on a full encoder.

B Implementation details

This section provides the experimental details omitted from the main paper. All token-reduction methods are implemented as inference-time modifications of the frozen visual encoder. They introduce no additional training and use the same detector weights and input resolution as the corresponding unpruned detector. All experiments use a single NVIDIA L40S GPU. For AI-VAD, which applies the appearance encoder to a variable number of object crops per frame, we report the resulting per-crop latency.

B.1 Implementation and baseline configurations

ADoRe. For AnomalyCLIP and FAPrompt, we use 518×518 inputs, $37 \times 37 = 1{,}369$ patch tokens, and prune blocks 9–11 of CLIP ViT-L/14. For SubspaceAD, we use 672×672 inputs, $48 \times 48 = 2{,}304$ patch tokens, and prune blocks 15–17 of the 40-block DINOv2-g/14 encoder. For AI-VAD, we use 336×336 object crops, $24 \times 24 = 576$ patch tokens, and prune blocks 9–11 of the 24-block CLIP ViT-L/14 encoder. Layer indices are one-based.

The pruning ratio p in the main tables denotes the fraction of patch tokens removed after the complete pruning band. Because pruning is distributed over three blocks, the per-block keep ratio is $\rho = (1-p)^{1/M}$ with $M = 3$, *i.e.*, 0.7937, 0.7368, 0.6694, and 0.5848 for 50%, 60%, 70%, and 80% pruning, and the kept count is rounded down at each block. The same layers and ratios are used for every category. For all experiments, a pruned patch receives an anomaly score of exactly zero at its original location, while a kept patch retains the detector's native score.

The trim fraction is $\tau = 0.2$ for every backbone and ratio, and ranks are averaged with equal weight across the band layers seen so far.

For AI-VAD, the reference $\bar{\mu}^{(l)}$ at each band layer is an exponential moving average (momentum 0.99) of the trimmed means of earlier forward passes in the same video, pooled over the object crops of each pass. It is read before the crops of the current pass are added, so the current frame never contributes to its own reference, and it is reset at every video boundary.

Band calibration and the zero-shot protocol. $\phi(l)$ (Eq. 5) is estimated from 100 normal training images, with no anomalous images, masks, or test metrics. Calibration follows the same cross-dataset rule as the detector checkpoints: the band used on VisA is calibrated on MVTec-AD, and vice versa, so no image from the evaluation dataset is seen. Both directions select the same band (blocks 9–11 for CLIP ViT-L/14; 15–17 for DINOv2-g/14), which is held fixed for the medical and video benchmarks without recalibration.

Compute matching. Different reduction methods incur different token schedules and selection overheads. Applying the same nominal keep ratio to all of them would therefore give unequal compute. For every target pruning ratio, we calibrate each baseline's native reduction parameter to match ADoRe's encoder cost as closely as possible. The resulting settings are listed in Table S1. EViT and ATS entries are per-event keep fractions; ToMe entries are the number of tokens reduced per block; and GTP-ViT entries are the number of tokens propagated per block.

EViT. EViT (Liang et al., 2022) ranks patch tokens using the attention received from the class token. At each reduction block it retains the high-attention tokens and aggregates the removed tokens into one attention-weighted representative.

ToMe. Token Merging (ToMe) (Bolya et al., 2023) reduces the sequence by bipartite soft matching: similar token pairs are identified from their features and merged through a size-aware average.

Table S1: Reduction hyperparameters used for matched-compute comparison. "CLIP image" covers AnomalyCLIP and FAPrompt. A dash denotes a method not used for that detector.

Detector family	Pruned	EViT keep	ToMe r	ATS keep	GTP-ViT n_{prop}
CLIP image	50%	0.77	56	0.95	55
	60%	0.72	66	0.93	68
	70%	0.67	77	0.91	81
	80%	0.62	88	0.89	96
DINOv2 image	50%	0.75	34	0.97	41
	60%	0.68	41	0.96	50
	70%	0.60	50	0.94	70
	80%	0.53	53	0.93	90
AI-VAD	50%	0.82	15	0.96	–
	60%	0.75	20	0.95	–
	70%	0.72	23	0.94	–
	80%	0.66	28	0.92	–

Unlike pruning methods, ToMe preserves the contribution of every input patch in a smaller set of representative tokens.

ATS. Adaptive Token Sampling (ATS) (Fayyaz et al., 2022) constructs a sampling distribution from class-token attention and the value vectors, and retains tokens by inverse-transform sampling from its cumulative distribution.

GTP-ViT. GTP-ViT (Xu et al., 2024) represents patch interactions as a graph and propagates information from removed nodes to the retained tokens. We use its graph-propagation coefficient 0.1, eight semantic neighbors, token-scale attention, and the per-block propagation counts in Table S1. For dense anomaly maps, locations without an explicit output are assigned the representation of the nearest surviving patch, as is also done for ATS; EViT and ToMe use their own aggregated or merged representatives.

B.2 Anomaly detection models

AnomalyCLIP. AnomalyCLIP (Zhou et al., 2024) is a zero-shot anomaly detector that learns object-agnostic normal and abnormal text prompts and compares them with global and patch-level CLIP features. We use CLIP ViT-L/14@336px, evaluate at resolution 518, and extract patch features from blocks 6, 12, 18, and 24. The design uses prompt depth 9, 12 context tokens, four compound text-context tokens, and the DPAM replacement from block 20. We follow the standard cross-dataset protocol: the VisA-trained checkpoint is used on MVTec-AD, while the MVTec-AD-trained checkpoint is used on VisA and the medical datasets.

FAPrompt. FAPrompt (Zhu et al., 2025) is a zero-shot CLIP detector that decomposes abnormality into complementary fine-grained prompts. Its compound abnormality prompt module models diverse abnormal patterns, while its data-dependent abnormality prior adapts the prompts to each test image. We use the authors' ViT-L/14@336px model at resolution 518 with prompt depth 9, 12 context tokens, four compound context tokens, and DPAM beginning at block 20. Checkpoints are selected with the same cross-dataset protocol as AnomalyCLIP.

SubspaceAD. SubspaceAD (Lendering et al., 2026) is a training-free few-shot detector built on DINOv2 with register tokens. It fits a category-specific PCA model to features of a small set of normal reference images and uses the PCA reconstruction residual as the anomaly score. We use facebook/dinov2-with-registers-giant, average features from blocks 23–29, and retain enough principal components to explain 99% of the reference variance. Results use the 1-shot setting. Following the official few-shot configuration, the reference image is augmented with 30 rotations (except for categories for which the official implementation disables augmentation).

AI-VAD. AI-VAD (Reiss & Hoshen, 2025) detects video anomalies by combining interpretable velocity, pose, and appearance attributes for foreground objects. Token reduction is applied only to its appearance encoder; the velocity and pose branches and their score fusion remain unchanged for all methods. We use CLIP ViT-L/14@336px for appearance features and encode 336×336 detected-object crops.

B.3 Datasets

Industrial anomaly detection. MVTec-AD (Bergmann et al., 2019) contains high-resolution images from 15 industrial object and texture categories. Its training split contains only normal samples, whereas the test split contains both normal samples and multiple defect types with pixel-level masks. VisA (Zou et al., 2022) contains 10,821 images from 12 object categories, including 9,621 normal and 1,200 anomalous images. It contains visually diverse defects and provides pixel annotations. We use the official category definitions and the standard one-class train/test split for both datasets.

Medical anomaly detection. We use three binary image-level benchmarks. HeadCT (Salehi et al., 2021) consists of 200 head CT slices, equally divided between normal cases and intracranial hemorrhage. BrainMRI (Salehi et al., 2021) contains 253 brain MR images: 98 normal images and 155 tumor images. Br35H (Hamada) contains 3,000 brain MR images, equally divided between tumor-free and tumor-bearing cases. These experiments assess anomaly classification only because the evaluated versions do not provide pixel-level masks.

Video anomaly detection. CUHK Avenue (Lu et al., 2013) contains 16 normal training videos and 21 test videos recorded in a campus avenue. Test anomalies include unusual actions and objects such as running, throwing, and bicycles. ShanghaiTech Campus (Liu et al., 2018) is a larger multi-scene benchmark with 330 normal training videos and 107 test videos collected from 13 scenes. We use the official splits and frame-level labels.

B.4 Evaluation metrics

Image-level metrics. Image AUROC measures the probability that an anomalous image receives a higher score than a normal image and is insensitive to the decision threshold. Image AP is the area under the precision–recall curve and is more sensitive to class imbalance. We first compute each metric independently for every object category and then report the unweighted mean over categories. For the medical datasets, which each form one binary task, the metric is computed over all images directly.

Pixel-level metrics. Pixel AUROC treats every annotated pixel as a binary prediction and computes the area under the pixel-wise ROC curve. AUPRO evaluates localization by averaging the per-region overlap of connected anomalous regions while varying the threshold; following the standard industrial-AD protocol, the curve is integrated up to a false-positive rate of 0.30 and normalized by this range. As above, both metrics are computed per category and macro-averaged across categories.

Video metrics. We report frame-level Micro and Macro AUROC. Micro AUROC concatenates the scores and labels of all test frames and computes one global ROC curve, so longer videos contribute more observations. Macro AUROC computes AUROC for each test video separately and averages over videos, assigning equal weight to each video. Frames for which an object detector returns multiple foreground crops receive the maximum object-level appearance score before fusion with the other AI-VAD attributes.

C Additional experiments

C.1 Anomalous-token leakage

Table S2 extends the measurement of Section 3.1 across criteria and pruning ratios. A patch counts as anomalous if it overlaps at least one ground-truth pixel.

Table S2: **Share of anomalous patches removed** (%, lower is better), pooled over the anomalous test images. Random removes anomalous patches at the overall rate by construction.

Dataset	Pruning ratio	EViT	ToMe	ATS	GTP-ViT	Ours	Random
VisA	50 %	7.3	23.2	6.9	6.3	**1.6**	49.7
VisA	60 %	13.5	31.2	13.6	13.1	**4.3**	60.1
MVTec-AD	50 %	23.3	29.6	22.0	21.0	**14.0**	50.1
MVTec-AD	60 %	31.2	39.6	30.2	29.3	**21.9**	59.9

C.2 Scoring and encoding under the actual criterion

Fig. 2 splits the gain of ground-truth normal pruning into a scoring and an encoding term. Table S3 repeats the split with the sets ADoRe actually releases.

Table S3: **Scoring and encoding under the actual criterion.** Pixel-level gain (pp) of AnomalyCLIP on VisA over the unpruned pipeline, split as in Eq. 2. The ground-truth and CLS rows repeat Fig. 2; the ADoRe rows use the sets ADoRe actually releases.

		Prune ratio (%)			
Term	Criterion	50	60	70	80
Δ_{scoring}	Prune Normal (GT)	$+1.9$	$+2.6$	$+3.5$	$+3.5$
	Prune Background (CLS)	-0.3	-0.9	-2.0	-3.7
	ADoRe (Ours)	$+0.8$	$+0.9$	$+0.6$	-0.2
Δ_{encoding}	Prune Normal (GT)	$+0.5$	$+0.5$	$+0.4$	$+0.6$
	Prune Background (CLS)	$+0.8$	$+0.9$	$+1.0$	$+1.0$
	ADoRe (Ours)	$+0.9$	$+1.0$	$+1.0$	$+1.0$
Δ_{total}	Prune Normal (GT)	$+2.4$	$+3.1$	$+3.9$	$+4.1$
	Prune Background (CLS)	$+0.5$	-0.02	-1.0	-2.7
	ADoRe (Ours)	$+1.7$	$+1.9$	$+1.6$	$+0.8$

C.3 Trim fraction

Table S4 varies the trim fraction τ of Eq. 3 at 80% pruning. Every metric stays within 0.8 points across τ from 0 to 0.4, so the method is not sensitive to this choice, and we use $\tau = 0.2$ throughout. Pixel AUROC on MVTec-AD rises steadily with τ, which fits its larger defects contaminating the reference more.

Table S4: **Trim fraction.** AnomalyCLIP at 80% pruning.

	VisA		MVTec-AD	
τ	Image	Pixel	Image	Pixel
0.0	(83.8, 86.8)	(**94.8**, 86.2)	(90.6, **95.7**)	(86.5, 80.4)
0.1	(83.8, 86.7)	(94.7, **86.3**)	(90.6, 95.6)	(86.6, 80.3)
0.2	(**84.0**, **86.9**)	(94.7, **86.3**)	(**90.8**, **95.7**)	(86.8, **80.7**)
0.3	(83.6, 86.5)	(94.7, **86.3**)	(90.4, 95.6)	(**86.9**, 80.4)
0.4	(83.4, 86.5)	(94.6, 86.0)	(90.1, 95.5)	(**87.0**, 80.4)

C.4 Map construction

Section 4.3 assigns pruned locations zero instead of interpolating their scores from surviving neighbors. Table S5 holds selection fixed and varies only this step. The direct assignment is ahead on AUPRO in every setting and on pixel AUROC in all but one.

Table S5: **Map construction.** Selection is fixed to ours, and only the treatment of released locations varies. Each cell is pixel-level (AUROC, AUPRO).

		AnomalyCLIP		FAPrompt	
Pruning ratio	Map	VisA	MVTec-AD	VisA	MVTec-AD
50 %	Interpolated	(95.3, 85.3)	(89.0, 81.1)	(95.5, 87.6)	(87.9, 81.9)
	Direct (ours)	(**95.6**, **85.8**)	(**89.7**, **82.4**)	(**95.9**, **88.7**)	(**88.2**, **84.1**)
70 %	Interpolated	(**95.5**, 86.1)	(**87.6**, 78.7)	(95.4, 86.9)	(85.8, 78.2)
	Direct (ours)	(**95.5**, **86.4**)	(**88.6**, **81.6**)	(**95.5**, **88.5**)	(**86.3**, **82.5**)

C.5 Wall-clock latency

Table S6 reports wall-clock time, broken down into the ViT encoder and the stages after it. The encoder dominates every unpruned pipeline. Table S7 breaks down the full video AD pipeline. Object detection and optical flow run once per frame, while the appearance encoder runs once for every detected object, 6.4 times per frame on average on Avenue. The encoder therefore takes a larger share of the runtime in scenes with more objects.

Table S6: **Measured latency** (ms per image) on a single NVIDIA L40S. Latency is measured with batch size 1 and averaged over 100 runs after 30 warm-ups. ViT is the encoder forward pass, Scoring the detector's own head, and Map the assembly of the anomaly map.

Detector	Setting	ViT	Scoring	Map	Total	Speedup	ViT share (%)
AnomalyCLIP	Unpruned	80.0	1.12	0.46	81.6	1.00×	98.0
	Prune 50%	43.2	1.14	0.78	45.1	1.81×	95.8
	Prune 70%	36.3	1.05	0.77	38.1	2.14×	95.3
FAPrompt	Unpruned	80.0	14.7	0.32	95.0	1.00×	84.2
	Prune 50%	43.2	14.0	0.25	57.5	1.65×	75.1
	Prune 70%	36.3	17.0	0.33	53.6	1.77×	67.7
SubspaceAD	Unpruned	247.1	16.8	1.13	265.0	1.00×	93.2
	Prune 50%	172.5	9.12	1.09	182.7	1.45×	94.4
	Prune 70%	133.7	6.14	1.11	141.0	1.88×	94.8

Table S7: **End-to-end latency of AI-VAD** on Avenue (ms per frame). Preprocessing includes object detection and optical flow. Measured with batch size 1.

Detector	Setting	Preprocessing	ViT	Scoring + Map	Total	Speedup	ViT share (%)
AI-VAD	Unpruned	50.2	75.5	0.01	125.7	1.00×	60.0
	Prune 50%	50.2	53.9	0.01	104.1	1.21×	51.8
	Prune 70%	50.2	44.8	0.01	95.0	1.32×	47.2

C.6 Batch size

The keep count $\lfloor \rho \, |K_{P_m}| \rfloor$ depends only on the number of surviving tokens and the ratio, never on image content, so every image in a batch retains the same number of tokens and the surviving set stays a rectangular index tensor. Pruning therefore needs no padding, no length bucketing and no per-image forward pass, and the speedup is similar at every batch size we tested (Table S8).

C.7 Seed variability of the few-shot detector

Among the detectors we use, only SubspaceAD is stochastic. It draws its normal reference images at random, so the 1-shot reference differs from run to run. The two CLIP detectors and AI-VAD run frozen checkpoints with deterministic inference, and our module adds no randomness of its own, so repeating those runs reproduces the same numbers. We therefore repeat every SubspaceAD run with five seeds. The SubspaceAD rows of Table 1 report the mean, and Table S9 adds standard deviation.

Table S8: **Batch size.** End-to-end speedup at a 60 % pruning ratio on VisA.

Batch size	1	2	4	8	16
AnomalyCLIP (CLIP ViT-L/14)	1.98×	2.02×	2.01×	1.99×	2.00×
SubspaceAD (DINOv2 ViT-g/14)	1.64×	1.63×	1.72×	1.71×	1.66×

Table S9: **Seed variability of SubspaceAD.** Mean over five seeds of the 1-shot reference image with standard deviation. Image and Pixel are (AUROC, AP) and (AUROC, AUPRO). Ratios and latency follow Table 1.

Method	MVTec-AD		VisA	
	Image↑	Pixel↑	Image↑	Pixel↑
SubspaceAD	$(96.3 \pm 0.83, 98.4 \pm 0.30)$	$(97.4 \pm 0.14, 94.4 \pm 0.21)$	$(92.9 \pm 0.40, 92.5 \pm 0.47)$	$(98.2 \pm 0.07, 95.2 \pm 0.23)$
	Prune **50%** of Tokens		Prune **60%** of Tokens	
+ EViT	$(96.2 \pm 0.84, 98.4 \pm 0.29)$	$(96.4 \pm 0.15, 93.8 \pm 0.20)$	$(\mathbf{93.8} \pm 0.37, 93.6 \pm 0.42)$	$(97.9 \pm 0.08, 95.3 \pm 0.17)$
+ ToMe	$(96.1 \pm 0.90, 98.3 \pm 0.31)$	$(\mathbf{97.4} \pm 0.13, \mathbf{94.4} \pm 0.21)$	$(92.7 \pm 0.34, 92.8 \pm 0.54)$	$(98.1 \pm 0.07, 94.7 \pm 0.25)$
+ ATS	$(95.5 \pm 1.16, 97.9 \pm 0.60)$	$(96.9 \pm 0.19, 93.8 \pm 0.36)$	$(89.0 \pm 1.24, 88.7 \pm 1.12)$	$(97.8 \pm 0.09, 94.0 \pm 0.32)$
+ GTP-ViT	$(95.7 \pm 1.22, 98.2 \pm 0.62)$	$(97.0 \pm 0.25, 93.8 \pm 0.43)$	$(90.7 \pm 0.72, 90.4 \pm 0.67)$	$(97.9 \pm 0.08, 94.3 \pm 0.27)$
+ ADoRe (Ours)	$(\mathbf{96.6} \pm 0.59, \mathbf{98.7} \pm 0.20)$	$(97.2 \pm 0.14, \mathbf{94.4} \pm 0.20)$	$(\mathbf{93.8} \pm 0.40, \mathbf{94.0} \pm 0.56)$	$(\mathbf{98.5} \pm 0.06, \mathbf{95.7} \pm 0.13)$
	Prune **70%** of Tokens		Prune **80%** of Tokens	
+ EViT	$(96.2 \pm 0.76, 98.4 \pm 0.29)$	$(95.5 \pm 0.28, 92.6 \pm 0.45)$	$(93.4 \pm 0.66, 93.3 \pm 0.74)$	$(96.6 \pm 0.16, 93.3 \pm 0.52)$
+ ToMe	$(95.3 \pm 0.85, 97.7 \pm 0.38)$	$(\mathbf{97.3} \pm 0.16, \mathbf{94.3} \pm 0.24)$	$(92.6 \pm 0.73, 92.6 \pm 0.82)$	$(\mathbf{98.0} \pm 0.09, 94.6 \pm 0.31)$
+ ATS	$(93.1 \pm 1.18, 96.7 \pm 0.55)$	$(96.2 \pm 0.30, 92.7 \pm 0.48)$	$(82.8 \pm 1.62, 82.1 \pm 1.79)$	$(96.6 \pm 0.25, 90.3 \pm 0.71)$
+ GTP-ViT	$(95.2 \pm 0.92, 97.9 \pm 0.37)$	$(96.8 \pm 0.22, 93.4 \pm 0.34)$	$(89.9 \pm 0.95, 89.5 \pm 1.04)$	$(97.8 \pm 0.11, 94.0 \pm 0.35)$
+ ADoRe (Ours)	$(\mathbf{96.5} \pm 0.59, \mathbf{98.6} \pm 0.26)$	$(96.8 \pm 0.19, 93.9 \pm 0.25)$	$(\mathbf{93.8} \pm 0.35, \mathbf{94.0} \pm 0.41)$	$(96.9 \pm 0.07, \mathbf{94.7} \pm 0.54)$

D DISCUSSION

Defects cover 4.7% of the image on average on MVTec-AD against 1.0% on VisA. Table S10 bins anomalous images by defect area at 70% pruning. Small defects gain and large ones lose, and the loss follows leakage. On MVTec-AD, images whose defect covers more than 20% of the image lose 3.2 and 8.8 points of pixel AUROC, with more than half of their anomalous patches removed.

Part of this loss is forced by the budget. With 410 of 1,369 patches kept, 49 MVTec-AD images have more anomalous patches than survivors, which fixes a lower bound of 12.1% on pooled leakage against the measured 31.1%. On VisA the same bound is 0.4%.

The loss comes from the fixed budget. The keep count depends only on ρ and never on image content, which keeps the surviving set a rectangular tensor and the speedup constant across batch sizes. A defect that covers most of the image, however, needs more tokens than this budget allows. A content-adaptive budget could fix this, for instance by pruning only while the released tokens stay close to the trimmed reference. Each image would then keep a different number of tokens, so batches would need padding or masking, and the uniform speedup would be lost. Since large defects are rare in both benchmarks, we keep the fixed budget and suggest a lower pruning ratio where large defects are common.

Table S10: **Performance by anomaly area.** AnomalyCLIP at 70% pruning. ΔP-AUROC is the mean per-image change in pixel AUROC from the unpruned model. Leakage is the share of anomalous patches removed, pooled within each bin. Five VisA masks that become empty after resizing are excluded.

	VisA			MVTec-AD		
Area (%)	Images	ΔP-AUROC	Leakage (%)	Images	ΔP-AUROC	Leakage (%)
(0, 1]	1003	+1.11	7.8	498	+0.66	4.6
(1, 2]	100	+1.40	17.6	219	+0.22	9.6
(2, 5]	49	+0.98	16.1	298	+0.38	13.4
(5, 10]	6	−2.18	26.0	121	+0.75	20.6
(10, 20]	25	−0.14	8.1	64	−0.25	26.0
(20, 40]	12	−1.71	37.4	18	−3.21	52.8
(40, 100]	0	–	–	40	−8.79	60.2