Harnessing Vision-Language Models for Time Series Anomaly Detection

Zelin He[1], **Sarah Alnegheimish**[2], **Matthew Reimherr**[1,3*]

[1]Pennsylvania State University
[2]Massachusetts Institute of Technology
[3]Amazon
zbh5185@psu.edu, smish@mit.edu, mlr36@psu.edu

Abstract

Time-series anomaly detection (TSAD) has played a vital role in a variety of fields, including healthcare, finance, and sensor-based condition monitoring. Prior methods, which mainly focus on training domain-specific models on numerical data, lack the visual–temporal understanding capacity that human experts have to identify contextual anomalies. To fill this gap, we explore a solution based on vision language models (VLMs). Recent studies have shown the ability of VLMs for visual understanding tasks, yet their direct application to time series has fallen short on both accuracy and efficiency. To harness the power of VLMs for TSAD, we propose a two-stage solution, with (1) ViT4TS, a vision-screening stage built on a relatively lightweight pre-trained vision encoder, which leverages 2-D time series representations to accurately localize candidate anomalies; (2) VLM4TS, a VLM-based stage that integrates global temporal context and VLM's visual understanding capacity to refine the detection upon the candidates provided by ViT4TS. We show that without any time-series training, VLM4TS outperforms time-series pre-trained and from-scratch baselines in most cases, yielding a 24.6% improvement in F1-max score over the best baseline. Moreover, VLM4TS also consistently outperforms existing language model-based TSAD methods and is on average 36 × more efficient in token usage.

Code — https://github.com/ZLHe0/VLM4TS

Introduction

Time series anomaly detection (TSAD) has long been an important task for maintaining safety and efficiency in many domains, such as cloud computing, industrial monitoring, and web services (Lavin and Ahmad 2015). One critical challenge is that time series signals usually exhibit diverse temporal scales and dynamic behaviors, demanding deep temporal understanding to distinguish true anomalies from benign fluctuations. For example, a sudden increase in spacecraft telemetry readings may be benign if it has recurred frequently in historical records, whereas a gradual drift that deviates from an established trend could be an anomaly (Hundman et al. 2018). However, most existing

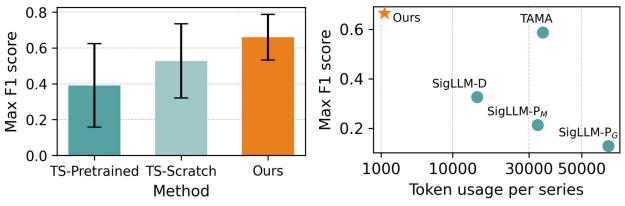

Figure 1: (Left) max F1 score averaged over all benchmarks, comparing VLM4TS to the best time-series-pretrained and from-scratch baselines. Error bars indicate standard deviation across datasets. (Right) max F1 score and token usage of VLM4TS versus language-model-based baselines.

TSAD models are built based on domain-specific assumptions and are trained on numerical data, limiting them to detecting surface-level anomalies without rich visual-temporal understanding capacities that human experts have (Kong et al. 2025).

Recent breakthroughs in multimodal foundation models have demonstrated human-level understanding across multiple modalities such as images, audio, and video (Wu et al. 2023). However, there is not yet a well-developed time-series-language model for time series anomaly detection, largely because time series–text corpora are scarce (Kong et al. 2025). Vision–language models (VLMs) have become a good alternative for two main reasons. First, their vision transformer architecture mimics a human-expert visual inspection mechanism. Compared with LLM-based approaches (Alnegheimish et al. 2024), a vision-based mechanism is especially effective at examining contextual anomalies that don't include extreme values but show deviation from normal signal patterns, see Figure 2-(a). Second, pretraining on massive image–text datasets equips VLMs with strong reasoning capabilities that generalize across domains (Nagar, Jaiswal, and Tan 2024). By encoding time series as visual representations, we can leverage VLMs' rich visual–textual understanding to perform TSAD tasks.

Recently, few pioneering works have explored VLM-based TSAD by rendering raw time series as line graphs with x-axis tick marks, then prompting the model to return anomaly intervals by referencing those tick labels (Zhuang et al. 2024; Zhou and Yu 2024; Xu et al. 2025). In prac-

*Work unrelated to Amazon.

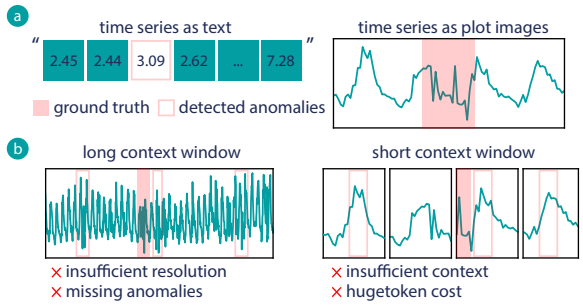

Figure 2: (a) Presenting time series as textual input for LLMs can obscure real anomalies and increase incorrect detection, whereas visualizing time series as line plots makes contextual anomalies, such as distortions, readily apparent. (b) The resolution–context dilemma in VLM-based TSAD: a global plot preserves long-range context but compresses details for detection (left), while small-window views maintain high resolution but provide limited context and incur high token cost (right).

tice, however, such a naive application of VLMs faces a major resolution-context dilemma when selecting the window size, as demonstrated in Figure 2-(b). With short context windows, we face challenges with **high cost, latency, and limited context**. To not only keep tick marks legible but also satisfy VLM input constraints, long series must be split into many small rolling window plots, leading to high token cost and inference latency, which is untenable for real-time monitoring across thousands of sensors (Alnegheimish et al. 2025). Furthermore, each plot captures only a brief temporal segment, thus preventing understanding over long-range dependencies. On the other hand, with long context windows, we face challenges with **limited resolution, poor localization, and missed anomalies**. Plotting long series as one single image reduces token usage but significantly degrades the image resolution. The VLM cannot pinpoint the exact interval boundaries via crowded x-axis ticks, even if an anomalous behavior is identified. Moreover, the volume of visual content overwhelms the attention of the model, causing it to overlook some of the anomalies.

In this paper, our goal is to harness the power of VLMs for TSAD by addressing the aforementioned resolution–context dilemma in a cost-effective manner. We propose a novel framework to decompose the anomaly detection task into two components: localization and verification. The first step localizes anomalies via a lightweight pretrained vision encoder (ViT4TS), while the second step verifies anomalies with a heavier yet more powerful VLM (VLM4TS). We emphasize that both stages of our framework work in a zero-shot setting—that is, neither ViT4TS nor VLM4TS is fine-tuned on the in-domain time series data. ViT4TS uses off-the-shelf, image-pretrained weights to screen windows purely via visual pattern matching, and VLM4TS relies on the pretrained cross-modal understanding of a VLM, showing great generalizability to a broad range of TSAD tasks. Figure 1 showcases the gain in detection performance and token efficiency of our approach. To our knowledge, this is the first VLM-based TSAD solution to achieve both superior detection accuracy and practical computational and token efficiency. To summarize, our main contributions are listed as follows.

- Motivated by human diagnostic workflows, we explore a solution that casts a 1-D anomaly detection as a 2-D visual understanding problem with VLMs. To resolve the resolution–context dilemma in VLM application, we explore decomposing the problem into sequential localization and verification stages.
- For the localization task, we propose ViT4TS, which leverages rich multi-scale cross-patch comparison to accurately localize anomaly candidates; then, for the verification task, we propose VLM4TS, that produce a final detection by filtering and refining on the detected anomalous intervals raised by ViT4TS with a deep temporal understanding on global temporal context.
- We demonstrate that our approach generalizes across domains and achieves state-of-the-art performance on multiple benchmark datasets without dataset-specific tuning, and requiring much less token usage than existing language model-based methods.

Related Work

In-domain TSAD. Early in-domain TSAD methods relied on statistical and distance-based techniques—e.g., threshold detectors and ARIMA residual analysis (Pena et al. 2013), which demand clean, uncontaminated training data and extensive domain expertise. Later on, many deep learning-based methods were introduced, applying structures like RNNs and sequence autoencoders to detect anomalies via prediction or reconstruction error (Hundman et al. 2018; Malhotra et al. 2016; Wong et al. 2022; He et al. 2024). Some works also explored VAE- and GAN-based frameworks to model normal behavior probabilistically (Geiger et al. 2020; Li et al. 2020; Kieu et al. 2019; Yin et al. 2020). More recently, Transformer-based models have shown the capacity to capture more complex temporal patterns for detection (Xu et al. 2021; Tuli, Casale, and Jennings 2022). For a comprehensive survey, see (Zamanzadeh Darban et al. 2024). Although highly effective in-domain, these approaches require large training sets and may not generalize well beyond the conditions they were trained on.

Foundation Model-based and LLM-based TSAD. Recent efforts pretrain general-purpose time series encoders (Gao et al. 2024; Zhang et al. 2024) or build large forecasting foundation models (Rasul et al. 2023; Ansari et al. 2024; Das et al. 2024; Goswami et al. 2024; Shi et al. 2024) that can be adapted for anomaly detection. However, these models usually excel at forecasting the next few time steps or encoding a short time window, but lack the capacity to perform human-like understanding over long-range temporal context, making them inefficient for detecting contextual anomalies across extended horizons. Prompting LLMs directly on numerical sequences has also been explored (Alnegheimish et al. 2024), yet they have shown that naive prompting underperforms some of the classical TSAD methods and comes with huge token costs.

Vision and VLM-based TSAD. Treating time series as images has shown promise across a range of tasks (Ni et al. 2025), including classification (Costa, Ribeiro, and Souza 2024; Kaewrakmuk and Srinonchat 2024; Li, Li, and Yan 2023) and forecasting (Yang, Fan, and Zhang 2023; Zeng et al. 2023; Semenoglou, Spiliotis, and Assimakopoulos 2023). However, the application in unsupervised anomaly detection remains relatively underexplored. There are few works exploring vision-based TSAD approaches using spectrograms or RP transformations (Namura and Ichikawa 2024; Lin et al. 2024), yet these approaches works purely with the vision modality. As a result, these pure vision-based approaches do not possess the multimodal reasoning and temporal understanding capabilities. Recent prototypes like TAMA (Zhuang et al. 2024) and related methods (Zhou and Yu 2024; Xu et al. 2025) prompt VLMs on small rolling-window plots to assign anomaly scores, but they incur prohibitive token costs and offer only limited temporal context. Consequently, they cannot scale to real-world tasks or capture contextual anomalies based on long-range temporal understanding required for robust anomaly detection.

Methodology

To overcome the resolution–context dilemma and token inefficiency of naive VLM-based TSAD, we introduce a unified, two-stage framework (Figure 3). For the first stage, we propose ViT4TS, which leverages a lightweight, pretrained vision transformer on sliding-window plots to rapidly screen the entire series and generate anomaly candidates via cross-patch comparisons. In the second stage, we propose VLM4TS, which takes each candidate proposal, renders it at a larger temporal scale, and applies an LLM's cross-modal understanding to verify and precisely localize anomalies across extended horizons.

Problem Formulation. Consider a univariate time series $\boldsymbol{x} = (x_1, ..., x_T) \in \mathbb{R}^T$, where x_t represents the value sampled at timestamp t. Assume there exists a set of anomalies of varied length $\mathbf{A} = \{(t_s, t_e)^i \mid 1 \le t_s < t_e \le T\}_{i=1}^m$ that is unknown a priori, our goal is to find a set of m anomalous time intervals, where t_s and t_e represent the start and end time points of an anomalous interval. In this paper, we mainly consider the univariate time series to focus on key method development; a discussion on extending the method for multivariate time series is provided in the Appendix.

ViT4TS: Visual Times Series Anomaly Screening

Time Series as Line Graphs. We convert 1-D time series into 2-D images by rendering them as clean line plots, a representation that aligns with both human intuition and the visual pretraining of vision encoders. Line graphs preserve temporal ordering and relative amplitude changes, enabling our ViT4TS module to perform meaningful cross-patch comparisons. As illustrated in Figure 3, to minimize non-informative visual artifacts, we omit axis ticks, legends, and other decorative elements.

Rolling Windows and Image Creation. Pretrained vision encoders usually require square inputs (e.g. 224×224 or 336×336 pixels), yet raw series $\boldsymbol{x}$ often span far more time steps than a single image width. To achieve precise, local-scale anomaly localization, we extract overlapping windows of length L_w—chosen to match the image width—using a stride $L_s = \lfloor L_w/4 \rfloor$. Each segment is rendered as a clean line plot on an $L_w \times L_w$ canvas, where each time tick maps to one pixel column. To make images of different time segments directly comparable, we use the same y-axis limits set to $[\min_t x_t, \max_t x_t]$ for every image. This process produces $N \approx (T - L_w)/L_s + 1$ images $\mathbf{I}_i \in \mathbb{R}^{L_w \times L_w \times 3}$ (replicated to three channels for compatibility), which are then fed into vision encoders. For brevity, we omit explicit window subscripts in $\mathbf{I}_i$ in later sections.

Multi-Scale Embedding Extraction. For image encoding, we adopt the pretrained CLIP vision encoder f_{clip} (Schuhmann et al. 2021), as it is a standard VLM vision encoder backbone (Liu et al. 2023, 2024) and thus can align our visual screening results with the subsequent VLM4TS stage, yet any other vision model can also be used within this framework. To capture fine-grained time series anomalies, following the standard practice in image segmentation (Jeong et al. 2023; Mishra et al. 2021), for each image we extract the full patch-level penultimate feature map $\mathbf{F} = f_{\text{clip}}(\mathbf{I}) \in \mathbb{R}^{P \times P \times D}$, where P is the patch size and D is the embedding dimension. In this way, anomalies like narrow spikes or brief dips can be localized at patch resolution.

At the same time, many anomalies, such as an extended jump or a drift that distorts the temporal patterns over dozens of time steps, will span multiple patches and demand broader context. To capture features at varying spatial scales, we generate pooled feature maps $\{\mathbf{F}^{(k)}\}_{k \in \mathcal{K}}$ by applying average pooling with kernel size k and stride 1 over the base patch feature map $\mathbf{F} \in \mathbb{R}^{P \times P \times D}$. For each $k \in \mathcal{K}$ and spatial location (i, j),

$$\mathbf{F}^{(k)}_{i,j,d} = \frac{1}{k^2} \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} \mathbf{F}_{i+u,\, j+v,\, d}.$$

Here $i, j = 1, \ldots, P - k + 1$ and $d = 1, \ldots, D$. Each $\mathbf{F}^{(k)} \in \mathbb{R}^{(P-k+1)\times(P-k+1)\times D}$ aggregates overlapping $k \times k$ neighborhoods, trading fine spatial detail for broader contextual information. These scale-specific feature maps are collected for later aggregation. Another advantage of such a multi-scale embedding is to enable rich cross-window comparisons, which in turn allows a larger rolling-window stride L_s, and thus improving efficiency without sacrificing localization accuracy.

Cross-Patch Comparison. As we don't have any ground-truth normal reference, we leverage the rarity of anomalies by matching each window's patch embeddings against those of all other windows at multiple scales. Concretely, for each sliding-window index i and scale $k \in \mathcal{K}$, let $\mathbf{L}_i^{(k)} \in \mathbb{R}^{P^2 \times D}$ denote the flattened embedding grid. To score patch p in window i, we first compute its cosine dissimilarity to every patch r in each other window $j \neq i$, then aggregate across windows via the median, that is, we obtain a cross-patch ref-

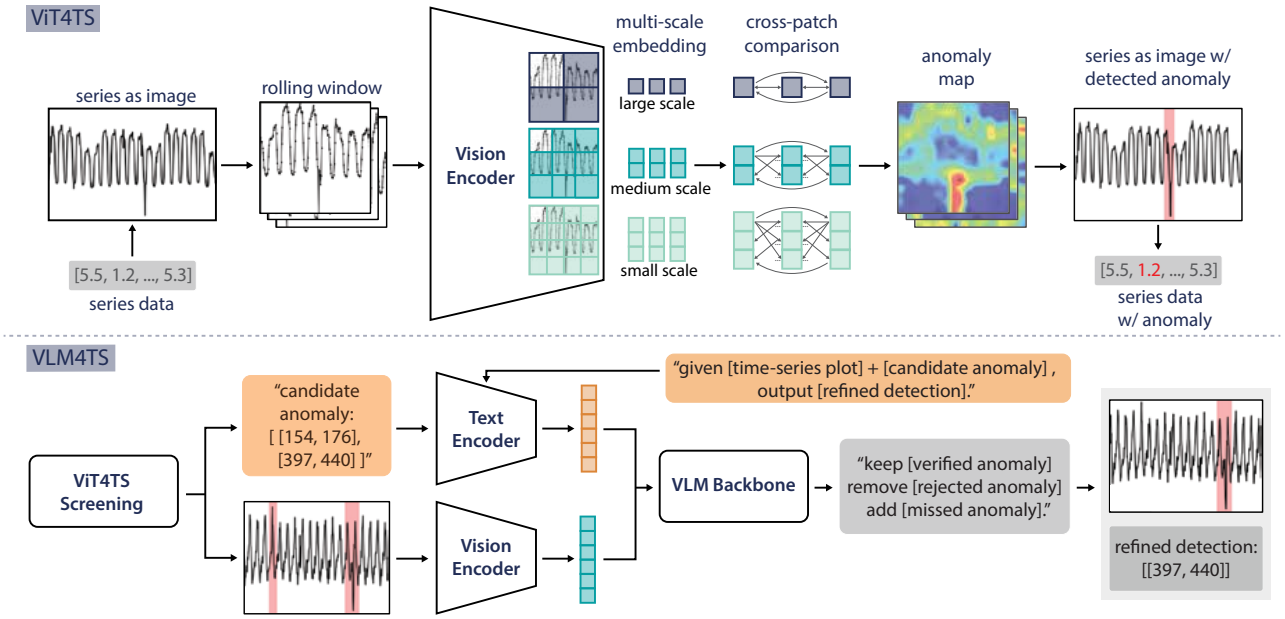

Figure 3: Overview of ViT4TS/VLM4TS (upper/lower pane). In ViT4TS, the raw time series is sliced into windows, and each window is transformed into an image and then embedded into multi-scale feature vectors. By comparing each of these features to others, ViT4TS localizes potentially anomalous regions, and outputs a set of candidate anomaly intervals. Then in VLM4TS, a VLM is then prompted to integrate global temporal context to refine the detection.

erence map

$$\mathbf{S}_i^{(k)}[p] = \text{median}_{j \neq i} \boldsymbol{U}_{i,j}^{(k)}[p], \text{ with}$$
$$\boldsymbol{U}_{i,j}^{(k)}[p] = \min_r \big[1 - \cos(\mathbf{L}_i^{(k)}[p], \mathbf{L}_j^{(k)}[r])\big].$$

Such a cross-patch matching can flexibly capture local pattern correspondences, regardless of spatial position, making it capable of detecting rare patterns while being robust to irregular seasonal shifts, trends, or periodic motifs. To reduce memory usage, we also evaluate a median-reference variant: we first compute a single reference map and then score each test patch, that is,

$$\tilde{\mathbf{S}}_i^{(k)}[p] = \min_r \big[1 - \cos(\mathbf{L}_i^{(k)}[p], \mathbf{V}^{(k)}[r])\big], \text{with}$$
$$\mathbf{V}^{(k)}[r] = \text{median}_j \mathbf{L}_j^{(k)}[r].$$

The median-reference variant drastically reduces memory usage, while the all-pairs variant remains more sensitive to rare anomalies. In our main experiments, we adopt the median-reference approach for its efficiency and report the min-reference results in the Appendix. Finally, to produce a single, fine-grained anomaly map, we upsample each scale-specific map $\{\mathbf{S}_i^{(k)}\}_{k \in \mathcal{K}}$ to the base patch resolution and fuse them via harmonic averaging at every patch location. This multi-scale fusion combines the pinpoint sensitivity of small-scale scores with the broader context captured at larger scales. Examples of the resulting aggregated patch-level anomaly maps are shown in Figure 3.

Anomaly Scoring and Localization. To assemble a final anomaly score per time step, we first map each fused patch-level score back to its original time index and average overlapping contributions to form a 2-D anomaly map $\mathbf{M} \in \mathbb{R}^{P \times T}$, where P is the patch resolution and T the series length. We then collapse $\mathbf{M}$ to a 1-D score $s \in \mathbb{R}^T$ by taking the q-th quantile across all rows: $s(t) = \text{quantile}_q\big(\mathbf{M}_{:,t}\big)$. Lower q (e.g. 0.1) increases sensitivity to subtle distortions, while higher q (e.g. 0.5) emphasizes large spikes; we set $q = 0.25$ for a balanced trade-off (See the Appendix for details). Finally, we choose a threshold τ at the $(1-\alpha)$-Gaussian quantile of s and extract all contiguous intervals $\hat{\mathbf{A}} = \{(t_s, t_e)^i \mid s(t) > \tau, \forall t_s \leq t \leq t_e\}_{i=1}^m$ as our detected anomalies from ViT4TS.

VLM4TS: Anomaly Verification and Refinement

Once ViT4TS produces a set of accurate candidate intervals, we send the full series image and these candidate proposals to VLM4TS for verification under the global context. ViT4TS excels at precise, local detection but may flag benign fluctuations or miss extended anomalies; VLM4TS uses cross-modal understanding along with the global temporal context to resolve these cases. In our main experiments, we employ GPT-4o, and evaluate alternatives in ablation studies.

Visual Input. We render each complete time series as an ordinary line graph—with evenly spaced x-axis ticks and y-axis value labels—so that the VLM can perceive trends, seasonality, and drifts at a glance. Because ViT4TS has already provided precise endpoints, we feed this single, full-length image per signal into the VLM's vision channel, avoiding further windowing and enabling understanding over the entire horizon.

Textual Input. In parallel, we supply a prompt that (1)

lists the initial proposals $\hat{\mathbf{A}}$ and remind the model these were generated from local shape matching; (2) instructs it to confirm only intervals that truly deviate from the global pattern; (3) rejects any false positives consistent with overall behavior; and (4) suggests additional intervals exhibiting clear statistical or visual irregularities that ViT4TS may have missed. We further ask the VLM to assign each interval a confidence score from 1 (low) to 3 (high). See the Appendix for the full prompt. The VLM returns a JSON object containing the refined anomaly set $\hat{\mathbf{A}}_{final} = \{(t_s, t_e)^i\}_{i=1}^{\hat{m}}$, per-interval confidence ratings, and a brief natural-language justification for each decision. We discard any interval with confidence = 1 to produce our final detection and diagnosis.

Experiments

We perform an array of experiments to evaluate the performance of ViT4TS and VLM4TS on a variety of benchmarks on industrial anomaly detection. We also conduct an extensive ablation study to validate the individual effectiveness of our proposed components. Detailed experimental setups, including data preprocessing, evaluation metrics are provided in the Appendix.

Experimental Setup

Benchmark Datasets and Baseline Methods. For unsupervised anomaly detection, we follow the standard evaluation protocol to test ViT4TS and VLM4TS on 11 widely used benchmark datasets in time series anomaly detection research (Lavin and Ahmad 2015; Hundman et al. 2018), spanning various domains from sensor data like astronomy sensory, web monitoring data like production traffic, to web metric data like volume of Twitter mentions, to evaluate the models' generalizability and adaptability. For baseline models, we compare our method against several anomaly detection approaches, from *statistical baselines* like `ARIMA` (Pena et al. 2013), to deep learning models currently considered state-of-the-art, including a forecasting-based LSTM model (`LSTM-DT`) (Hundman et al. 2018), (variational) reconstruction-based models like `LSTM-AE` (Malhotra et al. 2016), `VAE` (Park et al. 2018), and `TadGAN` (Geiger et al. 2020), hybrid models like `AER` (Wong et al. 2022), transformer-based models like Anomaly Transformer (`ATrans`) (Xu et al. 2021), and pre-trained time series foundation models like `UniTS` (Gao et al. 2024) and `TimesFM` (Das et al. 2024), as well as LLM-based approaches such as prompt-based detectors ($\texttt{SigLLM-P}_{\text{G}}$ on GPT, $\texttt{SigLLM-P}_{\text{M}}$ on Mistral) and LLM prediction-based models `SIGLLM-D` (Alnegheimish et al. 2024). We also report `TAMA` (Zhuang et al. 2024), a naive rolling-window VLM prompting baseline.

Evaluation. For any method that produces continuous anomaly scores, we first smooth the raw outputs with an exponentially weighted moving average. We then apply a Gaussian-based threshold of the form $\mu + k\sigma$, where μ and σ are the mean and standard deviation of the smoothed scores, sweeping k to compute the unweighted contextual F1 score (Geiger et al. 2020; Wong et al. 2022). Max F1 score (F1-max) results appear in the main text; full F1 results are provided in the Appendix. Methods yielding only binary labels are evaluated at their default settings and reported by raw F1. See Appendix for the formal definition of these metrics. All experiments run on an NVIDIA V100 GPU if not otherwise specified. Baselines are implemented via the Orion framework (Alnegheimish 2022) when available; otherwise, we follow each original implementation's setup and prompt.

Performance Evaluation

Overall Detection Performance. Table 1 reports the F1-max scores versus trained-from-scratch and time-series-pretrained baselines on all 11 benchmark datasets. Our two-stage VLM4TS framework achieves the highest average F1-max, outperforming competing methods on 9 out of 11 datasets, achieving a 24.6 % improvement in average F1-max score over the second-best baseline `LSTM-DT`. These results underscore that purely vision-driven screening, when coupled with powerful VLM understanding, can exceed state-of-the-art time series-based TSAD models in most cases. Furthermore, even the first-stage visual screening module ViT4TS ranks second overall, securing the top-two position on 6 datasets, showing the effectiveness of detecting time series anomalies from a visual perspective. In Table 2, we further compare against existing language model-based baseline methods, where VLM4TS also consistently outperform baselines by a large margin, achieving 13.3% performance gain over the VLM prompting-based method `TAMA` and $\times 2$ improvement over the strongest LLM-based method `SigLLM-D`. Moreover, we observe that both VLM-based methods substantially surpass LLM prompting frameworks, confirming that casting time series as images unlocks more effective anomaly detection than text-only representations.

Task-Specific Detection Performance. Compared with time-series-based methods in Table 1, VLM4TS delivers its strongest gains on real-world datasets dominated by contextual anomalies (e.g., the NAB dataset group), where it outperforms all baselines by a large margin. Its advantage stems from the ability to localize anomalies in a 2-D representation and then apply global-context verification to boost precision without sacrificing recall (see discussion on precision and recall in the Appendix). In contrast, on a few synthetic datasets, such as A3 and A4, where anomalies are densely and synthetically injected (average anomalies per time series: A3 = 9.39, A4 = 8.37 versus A1 = 2.66, A2 = 2.00), we see forecasting-based methods (e.g., `AER` and `LSTM-DT`) excelling compare to our method. Because VLM4TS assumes anomalies are rare, it adopts a more conservative behavior in high-density scenarios: visually similar, tightly packed anomaly points are interpreted as fluctuations within a broader pattern rather than as anomalies. This conservativeness results in fewer selected anomalies, leading to lower F1 scores on A3 and A4 (see Appendix for further discussion). Compared with language-based methods shown in Table 2, such as `TAMA`, VLM4TS delivers the strongest gains in datasets that demand long-range context, most notably the NAB dataset group (26.5%). This advantage arises from our two-stage design: ViT4TS supplies high-resolution, localized proposals, allowing VLM4TS to verify and refine anomalies over a much larger temporal horizon without los-

Type	Method	NAB					NASA		YAHOO				
		Art	AWS	AdEx	Traf	Tweets	MSL	SMAP	A1	A2	A3	A4	$\mu \pm \sigma$
Trained From Scratch	ARIMA	0.387	0.263	0.500	0.344	0.179	<u>0.585</u>	0.750	<u>0.650</u>	0.771	0.502	0.336	0.479±0.187
	AER	0.338	0.244	0.518	0.404	0.178	0.553	<u>0.753</u>	0.618	0.866	**0.711**	**0.614**	0.527±0.207
	TadGAN	0.338	0.196	0.385	0.421	0.205	0.612	0.593	0.492	0.667	0.135	0.109	0.378±0.189
	LSTM-DT	0.368	0.273	0.444	0.451	0.190	0.615	0.724	0.639	0.877	<u>0.704</u>	0.534	0.529±0.200
	LSTM-AE	0.231	0.244	0.400	0.416	0.232	0.487	0.673	0.583	0.853	0.584	0.201	0.446±0.203
	ATrans	0.262	0.168	0.200	0.365	0.147	0.454	0.567	0.263	0.554	0.437	0.394	0.346±0.142
	VAE	0.000	0.248	0.345	0.323	0.237	0.515	0.695	0.557	0.845	0.524	0.189	0.407±0.234
Time-series Pretrained	UniTS	0.182	0.246	0.326	0.479	0.167	0.561	0.723	0.605	0.760	0.126	0.110	0.390±0.233
	TimesFM	0.234	0.243	0.400	0.467	0.198	0.564	0.686	0.554	0.694	0.120	0.107	0.388±0.209
	TimesFM2	0.255	0.233	0.364	0.386	0.171	0.556	0.676	0.594	0.692	0.202	0.181	0.392±0.194
Ours	**ViT4TS**	<u>0.545</u>	<u>0.400</u>	<u>0.615</u>	<u>0.615</u>	<u>0.597</u>	0.543	0.726	0.614	<u>0.892</u>	0.614	<u>0.565</u>	<u>0.612±0.116</u>
	VLM4TS	**0.714**	**0.488**	**0.727**	**0.632**	**0.686**	**0.619**	**0.773**	**0.733**	**0.901**	0.497	0.474	**0.659±0.127**

Table 1: Detection performance (F1-max) of ViT4TS and VLM4TS versus trained-from-scratch and time-series-pretrained baselines on benchmark datasets. Each entry reports the maximum F1 score across all evaluated thresholds; the best score is shown in bold, and the second-best is underlined. Definition of the F1-max score, full F1 results and elapsed (wall-clock) time comparison are provided in the Appendix.

Type	Method	NAB			NASA			YAHOO			μ		
		F1-max	Tokens	Time	F1-max	Tokens	Time	F1-max	Tokens	Time	F1-max	Tokens	Time
LLM-based	SIGLLM-D	0.353	<u>11153</u>	<u>83.77</u>	0.232	<u>13157</u>	89.50	0.393	<u>21508</u>	62.20	0.326	<u>15273</u>	<u>78.49</u>
	SigLLM-P$_{\text{M}}$	0.206	14191	613.28	0.157	45050	2107.96	0.276	45082	984.45	0.213	34774	1235.23
	SigLLM-P$_{\text{G}}$	0.162	25006	2258.78	0.080	78187	2614.87	0.143	83207	2852.52	0.128	62133	2575.39
VLM-based	TAMA	<u>0.513</u>	40009	110.47	<u>0.631</u>	31563	<u>83.49</u>	<u>0.616</u>	27324	68.99	<u>0.587</u>	32965	87.65
Ours	**VLM4TS**	**0.649**	**1219**	**16.71**	**0.696**	**1213**	**20.97**	**0.651**	**1204**	**6.36**	**0.665**	**1212**	**14.68**

Table 2: Performance and efficiency comparison of VLM4TS versus language model–based baseline methods on benchmark datasets. "Tokens" reports the average number of tokens consumed per time series; "Time" indicates the average elapsed (wall-clock) time to generate detections per time series. Efficiency metrics have been adjusted to account for the methods' window and step size difference.

ing localization accuracy.

Computational Time and Token Cost. Table 2 shows the F1-max score, per-series elapsed time, and token consumption for VLM4TS and language model-based baseline methods. We also report per-series elapsed time comparison with time-series-based method in the Appendix. Due to our two-stage design, VLM4TS requires substantially less cost, reducing token usage by an average of 30× compared to pure LLM- and VLM-based detectors that encode numerical data into text or image for every rolling window. When compared against time series–only approaches—both pretrained foundation models and models trained from scratch, VLM4TS achieves comparable end-to-end runtime while achieving better overall detection accuracy. We also evaluate CPU-only operation of VLM4TS (with VLM running on the API) with the default ViT-B/16 backbone. On series spanning thousands of time points, ViT4TS screening completes in seconds, demonstrating feasibility for large-scale deployment with low token cost.

Additional Analysis

Ablation Study. Table 3 evaluates the contribution of each ViT4TS component and the necessity of visual screening for VLM4TS. We have the following observations:

- *Patch-level embedding:* replacing the detailed grid of patch embeddings with a single global summary embedding ([CLS] token) per window leads to a substantial drop by 11.94% on average on the benchmark datasets. This confirms that fine-grained, patch-level representations are critical for accurately localizing distortions in time series plots.
- *Cross-patch matching*: compared to position-aligned patches comparison (No cross-patch comparison) and row-wise patches comparison (No column-wise comparison), cross-patch matching leads to an substantial 18.76% and 30.54% improvement on YAHOO dataset group. This is because a flexible matching across both spatial dimensions is especially important on YAHOO dataset group, where seasonal and trend patterns can hide anomaly patterns.
- *Multi-scale embedding:* ablating multi-scale embedding extraction reduces performance, especially on NASA dataset group, which can lead to a 8.35% drop in F1-max, as multi-scale embeddings enhance detection to extended contextual anomalies that cannot be captured in a patch-level, particularly in domains like spacecraft telemetry.
- *Visual screening:* omitting the ViT4TS proposal stage and applying VLM4TS directly to full-series images causes a dramatic F1-max drop, especially on datasets with more dense anomalies like those in the YAHOO dataset group.

This shows the importance of a high-recall, local screening step; without it, VLM cannot reliably isolate multiple anomalies within complex temporal backgrounds.

Method	NAB	NASA	YAHOO
w/o patch-level embedding	0.519	0.578	0.541
w/o cross-patch comparison	0.504	0.613	0.514
w/o column-wise comparison	0.523	0.624	0.565
w/o multi-scale embedding	0.534	0.582	**0.677**
ViT4TS (ours)	**0.555**	**0.635**	0.671
w/o ViT4TS	0.539	0.517	0.292
VLM4TS (ours)	**0.649**	**0.696**	**0.651**

Table 3: Ablation study of ViT4TS and VLM4TS, reporting F1-max scores for each configuration.

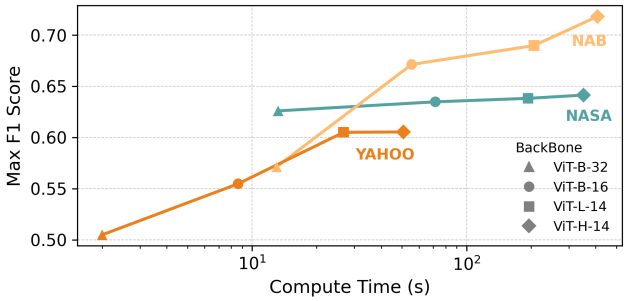

Figure 4: F1-max score and elapsed (wall-clock) time for ViT4TS with different backbones.

Backbone Scalability and Efficiency. To evaluate the impact of vision encoder capacity and patch resolution on screening performance, we replace our default ViT-B/16 backbone with other backbones within the ViT4TS pipeline and measure both F1-max and inference time (Figure 4). To accommodate real-world industrial environments where GPU resources may be limited, here we evaluate ViT4TS's elapsed time on CPU (Xeon E5, 32 GB RAM). Compared to the default backbone, coarsening the patch grid to 32×32 in ViT-B/32 yields much lower F1-max on NAB (–14.89%) and YAHOO (–8.99%), showing the importance of fine patch-level representations for accurate localization. Conversely, increasing model depth and hidden dimension improves detection on volatile series such as those in the YAHOO dataset group. However, these gains also come at a steep computational cost: For the YAHOO dataset group, ViT-B/32 completes screening in under 2s per series on CPU, whereas ViT-H/14 exceeds 50s on average.

Qualitative Results. Figure 5 illustrates anomaly localization on two NASA telemetry signals. In both cases, VLM4TS delivers the most precise detections: it refines ViT4TS's candidate proposals by comparing each candidate against the full-series context, correctly isolating the true anomalous intervals. By contrast, the "VLM-Long" ablation (prompting the VLM on the entire series without prior screening) misaligns its anomaly boundaries. Meanwhile, "VLM-Short" (`TAMA`), which prompts on every rolling window, generates numerous false positives due to its narrow context and incurs prohibitively high token usage.

Additional Analyses. In the Appendix, we provide further analysis of the framework, including patch-level embedding and anomaly map visualizations, comparisons across different VLM backbones, alternative visualizations such as spectrogram baselines, and a study of how window size and patch scale affect fine-grained detection, among other results and discussions.

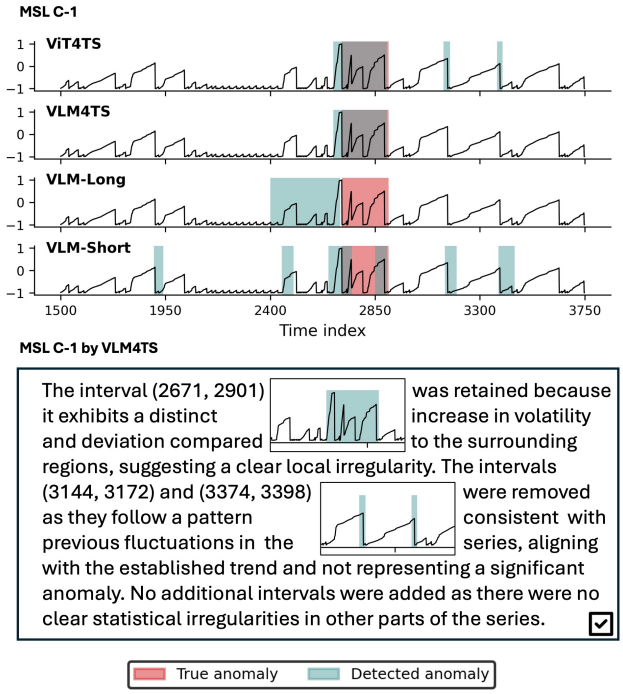

Figure 5: Qualitative results on MSL C-1, illustrating how VLM4TS refines the initial local anomaly proposals from ViT4TS. "VLM-Long" refers to VLM prompting on the full-series image (ablation without ViT4TS screening); "VLM-Short" refers to VLM prompting on the rolling-windows image. Only representative segments are shown for clarity.

Conclusion

We introduce a novel two-stage framework for unsupervised time-series anomaly detection that reframes 1-D time series as 2-D visual inputs and leverages pretrained vision and vision–language models without any in-domain training. In the first stage, ViT4TS applies multi-scale cross-patch comparisons to generate anomaly candidates in high resolution, and in the second stage, VLM4TS verifies these candidate proposals via temporal understanding with long context. Our method achieves state-of-the-art performance on diverse benchmarks while using many fewer tokens.

Acknowledgments

We thank Dr. Runze Li for valuable discussions and suggestions that contributed to the methodological formulation.

References

Alnegheimish, S. 2022. *Orion–a machine learning framework for unsupervised time series anomaly detection*. Ph.D. thesis, Massachusetts Institute of Technology.

Alnegheimish, S.; He, Z.; Reimherr, M.; Chandrayan, A.; Pradhan, A.; and D'Angelo, L. 2025. M2AD: Multi-Sensor Multi-System Anomaly Detection through Global Scoring and Calibrated Thresholding. In *International Conference on Artificial Intelligence and Statistics*, 4384–4392. PMLR.

Alnegheimish, S.; Nguyen, L.; Berti-Equille, L.; and Veeramachaneni, K. 2024. Can Large Language Models be Anomaly Detectors for Time Series? In *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*, 1–10. IEEE.

Ansari et al., A. F. 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Costa, H. V.; Ribeiro, A. G.; and Souza, V. M. 2024. Fusion of image representations for time series classification with deep learning. In *International Conference on Artificial Neural Networks*, 235–250. Springer.

Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2024. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*.

Gao, S.; Koker, T.; Queen, O.; Hartvigsen, T.; Tsiligkaridis, T.; and Zitnik, M. 2024. UniTS: A unified multi-task time series model. *Advances in Neural Information Processing Systems*, 37: 140589–140631.

Geiger et al., A. 2020. TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks. In *2020 IEEE International Conference on Big Data (Big Data)*, 33–43. IEEE.

Goswami, M.; Szafer, K.; Choudhry, A.; Cai, Y.; Li, S.; and Dubrawski, A. 2024. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*.

He, Z.; Reimherr, M.; Alnegheimish, S.; and Chandrayan, A. 2024. Weakly-supervised multi-sensor anomaly detection with time-series foundation models. *NeurIPS Workshop on Time Series in the Age of Large Models.*

Hundman et al., K. 2018. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 387–395.

Jeong, J.; Zou, Y.; Kim, T.; Zhang, D.; Ravichandran, A.; and Dabeer, O. 2023. WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19606–19616.

Kaewrakmuk, T.; and Srinonchat, J. 2024. Multi-Sensor Data Fusion and Time Series to Image Encoding for Hardness Recognition. *IEEE Sensors Journal*.

Kieu, T.; Yang, B.; Guo, C.; and Jensen, C. S. 2019. Outlier detection for time series with recurrent autoencoder ensembles. In *Ijcai*, 2725–2732.

Kong, Y.; Yang, Y.; Wang, S.; Liu, C.; Liang, Y.; Jin, M.; Zohren, S.; Pei, D.; Liu, Y.; and Wen, Q. 2025. Position: Empowering Time Series Reasoning with Multimodal LLMs. *arXiv preprint arXiv:2502.01477*.

Lavin, A.; and Ahmad, S. 2015. Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, 38–44. IEEE.

Li, L.; Yan, J.; Wang, H.; and Jin, Y. 2020. Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. *IEEE transactions on neural networks and learning systems*, 32(3): 1177–1191.

Li, Z.; Li, S.; and Yan, X. 2023. Time series as images: Vision transformer for irregularly sampled time series. *Advances in Neural Information Processing Systems*, 36: 49187–49204.

Lin, C.; Du, B.; Sun, L.; and Li, L. 2024. Hierarchical context representation and self-adaptive thresholding for multivariate anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 36(7): 3139–3150.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024. Improved Baselines with Visual Instruction Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 26296–26306.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 34892–34916. Curran Associates, Inc.

Malhotra et al., P. 2016. LSTM-based Encoder-Decoder for Multi-Sensor Anomaly Detection. *arXiv preprint arXiv:1607.00148*.

Mishra, P.; Verk, R.; Fornasier, D.; Piciarelli, C.; and Foresti, G. L. 2021. VT-ADL: A Vision Transformer Network for Image Anomaly Detection and Localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, 01–06.

Nagar, A.; Jaiswal, S.; and Tan, C. 2024. Zero-shot visual reasoning by vision-language models: Benchmarking and analysis. In *2024 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Namura, N.; and Ichikawa, Y. 2024. Training-Free Time-Series Anomaly Detection: Leveraging Image Foundation Models. *arXiv preprint arXiv:2408.14756*.

Ni, J.; Zhao, Z.; Shen, C.; Tong, H.; Song, D.; Cheng, W.; Luo, D.; and Chen, H. 2025. Harnessing Vision Models for Time Series Analysis: A Survey. *arXiv preprint arXiv:2502.08869*.

Park et al., D. 2018. A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-based Variational Autoencoder. *IEEE Robotics and Automation Letters*, 3(3): 1544–1551.

Pena et al., E. H. 2013. Anomaly Detection Using Forecasting Methods ARIMA and HWDS. In *2013 32nd International Conference of the Chilean Computer Science Society (sccc)*, 63–66. IEEE.

Rasul, K.; Ashok, A.; Williams, A. R.; Ghonia, H.; Bhagwatkar, R.; Khorasani, A.; Bayazi, M. J. D.; Adamopoulos, G.; Riachi, R.; Hassen, N.; et al. 2023. Lag-llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*.

Schuhmann, C.; Vencu, R.; Beaumont, R.; Kaczmarczyk, R.; Mullis, C.; Katta, A.; Coombes, T.; Jitsev, J.; and Komatsuzaki, A. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Semenoglou, A.-A.; Spiliotis, E.; and Assimakopoulos, V. 2023. Image-based time series forecasting: A deep convolutional neural network approach. *Neural Networks*, 157: 39–53.

Shi, X.; Wang, S.; Nie, Y.; Li, D.; Ye, Z.; Wen, Q.; and Jin, M. 2024. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*.

Tuli, S.; Casale, G.; and Jennings, N. R. 2022. Tranad: Deep transformer networks for anomaly detection in multivariate time series data. *arXiv preprint arXiv:2201.07284*.

Wong et al., L. 2022. AER: Auto-Encoder with Regression for Time Series Anomaly Detection. In *2022 IEEE International Conference on Big Data (Big Data)*, 1152–1161. IEEE.

Wu, J.; Gan, W.; Chen, Z.; Wan, S.; and Yu, P. S. 2023. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, 2247–2256. IEEE.

Xu, J.; Wu, H.; Wang, J.; and Long, M. 2021. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*.

Xu, X.; Wang, H.; Liang, Y.; Yu, P. S.; Zhao, Y.; and Shu, K. 2025. Can Multimodal LLMs Perform Time Series Anomaly Detection? *arXiv preprint arXiv:2502.17812*.

Yang, L.; Fan, X.; and Zhang, Z. 2023. Your time series is worth a binary image: machine vision assisted deep framework for time series forecasting. *arXiv preprint arXiv:2302.14390*.

Yin, C.; Zhang, S.; Wang, J.; and Xiong, N. N. 2020. Anomaly detection based on convolutional recurrent autoencoder for IoT time series. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(1): 112–122.

Zamanzadeh Darban, Z.; Webb, G. I.; Pan, S.; Aggarwal, C.; and Salehi, M. 2024. Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1): 1–42.

Zeng, Z.; Kaur, R.; Siddagangappa, S.; Balch, T.; and Veloso, M. 2023. From pixels to predictions: Spectrogram and vision transformer for better time series forecasting. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, 82–90.

Zhang, X.; Teng, D.; Chowdhury, R. R.; Li, S.; Hong, D.; Gupta, R.; and Shang, J. 2024. UniMTS: Unified Pretraining for Motion Time Series. *Advances in Neural Information Processing Systems*, 37: 107469–107493.

Zhou, Z.; and Yu, R. 2024. Can LLMs Understand Time Series Anomalies? *arXiv preprint arXiv:2410.05440*.

Zhuang, J.; Yan, L.; Zhang, Z.; Wang, R.; Zhang, J.; and Gu, Y. 2024. See it, Think it, Sorted: Large Multimodal Models are Few-shot Time Series Anomaly Analyzers. *arXiv preprint arXiv:2411.02465*.