

A Related Work

A.1 Time-series Anomaly Detection

Problem Formulation. Let $\mathbf{x}_{1:T} = (x_1, \dots, x_T) \in \mathbb{R}^{T \times d}$ be a time series sampled at a regular interval. Time-series anomaly detection seeks a function $f : \mathbb{R}^{T \times d} \rightarrow (\mathbf{s}_{1:T}, \mathbf{y}_{1:T})$ that, given the entire sequence, returns a vector of continuous anomaly scores $\mathbf{s}_{1:T} \in \mathbb{R}^T$ and, optionally, a binary label vector $\mathbf{y}_{1:T} \in \{0, 1\}^T$ obtained through a rule $y_t = \mathbb{I}[s_t > \tau]$. Each score s_t quantifies how strongly the statistical properties of observation x_t diverge from those expected under the normal generative process—after accounting for trend, seasonality and stochastic noise—while the label flags the corresponding time point (or minimal enclosing interval) as anomalous when the divergence exceeds a threshold τ . The objective is thus to isolate individual timestamps or contiguous segments whose behavior departs significantly from the normal regime, signaling faults, rare events, or structural changes in the underlying system.

Within this formulation, we primarily consider two TSAD settings based on assumptions about label availability: unsupervised, where no label information is available beyond the input time series, and semi-supervised, where only data known to be normal is accessible. While certain existing methods [43, 85] incorporate anomaly information and thus deviate from these settings, we concentrate on the more realistic unsupervised and semi-supervised scenarios. Approaches in this area have evolved from classical statistical techniques to modern deep learning and foundation models.

Statistical and Classical Models. Traditional methods such as STL decomposition [16], Matrix Profile [80], and statistical thresholding [27] detect anomalies by explicitly modeling components like trend and seasonality. These approaches are efficient and interpretable but often struggle with non-stationary and multivariate data.

Machine Learning-based Models. Classical unsupervised models include Isolation Forest [42], Subspace PCA [2], and clustering-based methods [78]. While these algorithms are lightweight and interpretable, they often rely on handcrafted features and are limited in their ability to capture temporal dependencies.

Deep Learning-based Models. Deep models such as AutoEncoders [57], USAD [7], and Omni-Anomaly [60] reconstruct normal patterns to identify anomalies. More advanced architectures like the Anomaly Transformer [73] and TimesNet [71] leverage attention mechanisms to model long-term dependencies. These models achieve superior performance but require extensive training and tuning, and they often lack interpretability.

Time-Series Foundation Models (TS-FMs). Recent efforts like Chronos [6], TimesFM [17], MOMENT [23], and Lag-Llama [52] pretrain large models on diverse time-series datasets to enable zero-shot anomaly detection. These models show promise in generalization and transferability but still rely on scalar anomaly scores and thresholding for detection, limiting interpretability and precise localization.

Comparison Across Models. Table 4 summarizes key differences:

Table 4: Comparison of unsupervised TSAD model families in terms of modeling capacity, training requirements, and output type.

Model Type	Temporal Modeling	Component Awareness	Domain Training	Output
Statistical	Low	Explicit	None	Score
ML-based	Moderate	Implicit	Light (generic heuristics)	Score
DL-based	High	Implicit	Required	Score
TS-FM	High	Learned	Pretrained	Score
LLM-based (ours)	Moderate	Structured (via prompt)	Pretrained	Label

Toward LLM-based TSAD. In contrast, Large Language Models (LLMs) offer a novel paradigm for TSAD by generating structured, interpretable anomaly reports through prompt-based natural language reasoning [4, 19, 85]. Rather than producing scalar anomaly scores, LLMs can identify and explain anomalous intervals in plain text. However, existing LLM-based TSAD methods struggle

with precise localization and temporal abstraction, especially when relying on visual inputs [75] or unstructured token prompts [33, 41].

Our approach addresses these challenges by combining statistical preprocessing (e.g., de-seasonalization) with index-aware prompting to improve both detection and localization. This structured fusion of classical time-series principles and generative language reasoning establishes a new direction for interpretable and flexible unsupervised TSAD.

A.2 LLM-based Time-series Anomaly Detection

Recent advances have explored the potential of using LLMs for TSAD, particularly in zero-shot or few-shot settings without domain-specific training. This line of research centers on a core question:

Can LLMs detect time-series anomalies without training? Studies [85, 19, 4, 75, 79, 43] have evaluated LLMs’ ability to detect anomalies directly from raw or lightly preprocessed sequences via prompting. While some setups succeed in identifying simple anomalies, key limitations persist—namely, low detection accuracy, sensitivity to prompt variations, and hallucinations leading to false positive cases. These investigations largely assess outcomes without examining the underlying causes of failure.

Vision-based anomaly detection with MLLMs. To bypass the difficulties LLMs face in capturing temporal patterns from textual input, some works [75, 85] propose converting time series into images and applying MLLMs to interpret the visualizations. This approach shows robustness to missing data and gross anomalies. However, it remains ineffective at detecting subtle irregularities hidden within complex, overlapping components. The transformation from sequence to image can also introduce distortions, particularly in high-dimensional settings, leading to inaccurate or unstable predictions.

Context-aware and interpretable detection. Other efforts [79, 43] investigate context-aware strategies that incorporate in-context learning, AnoCoT, and domain priors to improve both accuracy and interpretability. Although such methods enhance detection performance and support reasoning, they often rely on labeled anomaly examples during inference, which constrains their applicability to unsupervised or semi-supervised TSAD scenarios.

While earlier research has focused on whether LLMs can perform anomaly detection, our work shifts attention to understanding why they frequently fail. We identify two major barriers: (1) insufficient temporal abstraction and comprehension of latent components, and (2) structural misalignment between the input format and the task of localizing anomalies. Furthermore, we argue that visual representations alone are inadequate for precise localization, especially when positional indexing is essential. Instead, we advocate for structured textual representations that maintain explicit reference to index positions. Finally, unlike prior methods that leverage known anomalies to boost performance, we show that excluding such priors and instead using contextual cues better aligns with realistic, unsupervised TSAD conditions. All evaluations are conducted on the AnomLLM benchmark [85] to ensure consistency and fair comparison.

B Benchmark Details

In this paper, we use two benchmark datasets. The first is the synthetic AnomLLM benchmark, and the second is the real-world TSB-AD benchmark dataset. In this section, we describe the characteristics and statistical properties of each dataset.

B.1 AnomLLM benchmark

This appendix provides detailed information about the AnomLLM benchmark [85] datasets used in the anomaly detection study. The original dataset consists of 8 datasets in total: point anomalies, range anomalies, trend anomalies, frequency anomalies, noisy point anomalies, noisy trend anomalies, noisy frequency anomalies, and flat trend anomalies.

Among these, the first 4 are basic data forms, and the latter 4 are noisy variant forms. We only used the first 4 datasets, and we will provide an explanation of these datasets. As shown in the first three rows of Table 6, the four datasets differ in the presence or absence of the three components. In addition, [85] specifies appropriate values for minimum anomaly duration, normal duration, and

Table 5: Descriptions of the 21 prompt variants used in the AnomLLM benchmark. Prompts vary by supervision level (0-shot vs 1-shot), modality (text vs image), and auxiliary strategies such as CoT, statistical prefixing (PaP), arithmetic cues, and input formatting.

Code	Variant	Description
A	1shot-vision-cot	One-shot prompt with visual input and chain-of-thought (CoT) reasoning to guide step-by-step anomaly detection.
B	1shot-vision-calc	One-shot visual input with a correct arithmetic example to test numeracy-based reasoning.
C	1shot-vision-dyscalc	One-shot visual input with an incorrect arithmetic example to impair numeric reasoning.
D	1shot-vision	Basic one-shot visual prompt without CoT or arithmetic guidance.
E	0shot-vision-cot	Zero-shot visual input with CoT prompting to induce explicit anomaly reasoning.
F	0shot-vision-calc	Zero-shot visual input with a correct arithmetic example included.
G	0shot-vision-dyscalc	Zero-shot visual input with an incorrect arithmetic example to test robustness to misleading signals.
H	0shot-vision	Basic zero-shot visual prompt with only the time series image.
I	1shot-text-s0.3-cot	One-shot text input (subsamped to s0.3) with a CoT reasoning trace.
J	1shot-text-s0.3	One-shot prompt with s0.3 subsampled time series as plain text without reasoning.
K	0shot-text-s0.3-tpd	Zero-shot text with Token-per-Digit formatting to aid digit-level modeling.
L	0shot-text-s0.3-pap	Subsampled text with Prompt-as-Prefix: statistical summaries (mean, trend, etc.) precede the sequence.
M	0shot-text-s0.3-dyscalc	Subsampled text with a misleading arithmetic example to degrade arithmetic reasoning.
N	0shot-text-s0.3-csv	Text input formatted as CSV (index, value) to test structured data handling.
O	0shot-text-s0.3-cot-tpd	Combines CoT prompting with tokenized digit input for fine-grained reasoning.
P	0shot-text-s0.3-cot-pap	Combines CoT reasoning with Prompt-as-Prefix statistical context.
Q	0shot-text-s0.3-cot-csv	Structured CSV input with CoT reasoning to examine logical behavior over tabular text.
R	0shot-text-s0.3-cot	Basic CoT prompt with subsampled text input (no format enhancements).
S	0shot-text-s0.3-calc	Subsampled text with a correct arithmetic example for numeracy-augmented detection.
T	0shot-text-s0.3	Plain zero-shot text prompt using s0.3-length series in space-separated format.
U	0shot-text	Full-length zero-shot text prompt with raw values (space-separated).

other parameters for each dataset. These details are presented in Table 6. Also, AnomLLM provided 21 prompt variants including CoT. The descriptions for each variant are in Table 5. And also, the results of experiments using these variants are in Table 9.

B.1.1 Point Anomalies.

Normal data consists of a periodic sine wave between -1 and 1. Anomalies present as noisy and unpredictable deviations from the normal periodic pattern, with frequency 0.03, normal duration rate 800.0, anomaly duration rate 30.0, minimum anomaly duration 5, minimum normal duration 200, and anomaly standard deviation 0.5. The example series is illustrated in Figure 9.

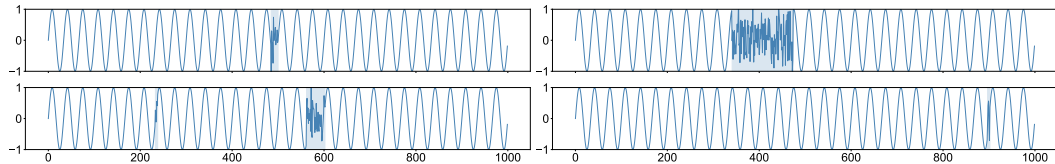


Figure 9: Example time series from the Point Anomalies dataset, with anomalies regions highlighted in blue.

Table 6: Comparison of anomaly statistics across different synthetic anomaly types.

Statistic	Point	Range	Trend	Frequency
Trend component	✗	✗	✓	✗
Seasonality component	✓	✗	✓	✓
Noise component	✗	✓	✗	✗
# Time series	400	400	400	400
# Samples per time series	1000	1000	1000	1000
Minimum anomaly duration	5	5	50	7
Minimum normal duration	200	10	800	20
Average anomaly ratio	0.0320	0.0236	0.0377	0.0341
# Time series without anomalies	117 (29.25%)	121 (30.25%)	230 (57.50%)	40 (10.00%)
Average anomalies per series	1.17	1.20	0.42	2.16
Maximum anomalies per series	4	5	1	7
Average anomaly length	27.26	19.73	88.61	15.77
Maximum anomaly length	165.0	113.0	200.0	111.0

B.1.2 Range Anomalies

Normal data comprises Gaussian noise with mean 0. Anomalies manifest as sudden spikes with values much further from 0 than the normal noise, with normal duration rate 800.0, anomaly duration rate 20.0, minimum anomaly duration 5, minimum normal duration 10, and anomaly size range (0.5, 0.8). The example series is illustrated in Figure 10.

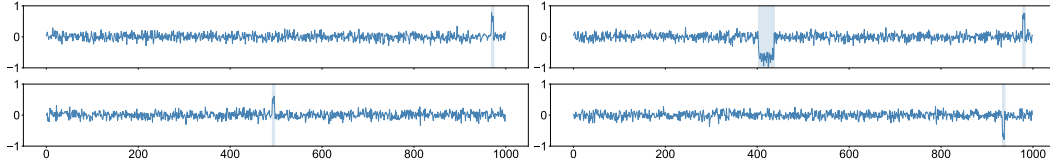


Figure 10: Example time series from the Range Anomalies dataset, with anomalies regions highlighted in blue.

B.1.3 Trend Anomalies.

Normal data follows a steady but slowly increasing trend from -1 to 1. Anomalies appear as sections where the data increases much faster or decreases, deviating from the normal trend, with trend negation probability 50%, frequency 0.02, normal duration rate 1700.0, anomaly duration rate 100.0, minimum anomaly duration 50, minimum normal duration 800, normal slope 3.0, and abnormal slope range (6.0, 20.0). The example series is illustrated in Figure 11.

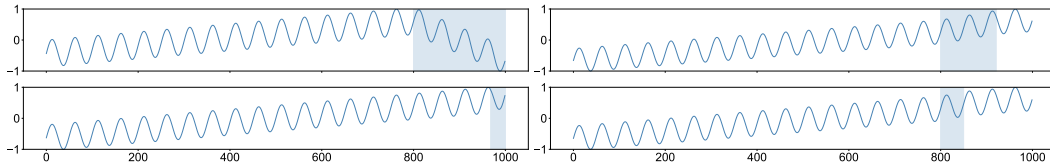


Figure 11: Example time series from the Trend Anomalies dataset, with anomalies regions highlighted in blue.

B.1.4 Frequency Anomalies.

Normal data is characterized by a periodic sine wave between -1 and 1. Anomalies occur as sudden changes in frequency, producing irregular periods between peaks, with frequency 0.03, normal duration rate 450.0, anomaly duration rate 15.0, minimum anomaly duration 7, minimum normal duration 20, and frequency multiplier 3.0. The example series is illustrated in Figure 12.

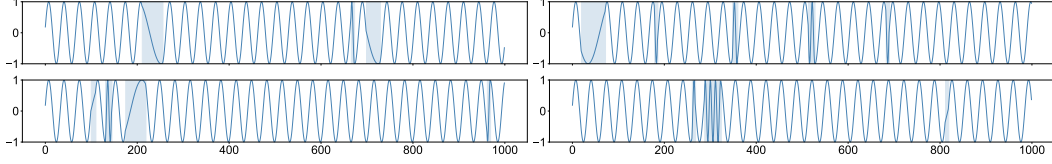


Figure 12: Example time series from the Frequency Anomalies dataset, with anomalies regions highlighted in blue.

B.2 TSB-AD Benchmark

This section describes the real-world dataset, TSB-AD benchmark [44], used in Section 4. The benchmark initially collected 13 univariate and 20 multivariate datasets. Following the curation process within TSB-AD, they obtained a total of 23 univariate and 17 multivariate datasets. Internally, datasets with a dimension of 1 were referred to as TSB-AD-U, among which we used the evaluation subset for our experiments.

To reduce the overall experimental cost, we restrict the evaluation set by selecting time series from eight categories (highlighted in green in Table 7) within the TSB-AD-U benchmark, focusing on those with relatively shorter lengths.

Table 7: TSB-AD-U evaluation benchmark dataset statistics. The total length is computed as the product of the average time series length and the number of series (Count).

Dataset	Count	Dim	Total Len	Avg. # Anomaly	Avg. Len	Anomaly Ratio	Category
NEK [59]	8	1	8,584	2.9	51.1	8.0%	P&Seq
TAO [48]	2	1	20,000	838.7	1.1	9.4%	P&Seq
MSL [30]	7	1	23,111	1.3	130.0	5.8%	Seq
Power [34]	1	1	35,040	4.0	750.0	8.5%	Seq
Daphnet [8]	1	1	38,774	6.0	384.3	5.9%	Seq
YAHOO [37]	30	1	45,270	5.5	2.5	0.6%	P&Seq
SED [11]	2	1	59,998	14.7	64.0	4.1%	Seq
TODS [36]	13	1	65,000	97.3	18.7	6.3%	P&Seq
NAB [3]	23	1	114,758	1.6	370.1	10.6%	Seq
Stock [67]	8	1	130,000	1246.9	1.1	9.4%	P&Seq
SMAP [30]	17	1	133,770	1.2	210.1	2.8%	Seq
CATSV2 [20]	1	1	300,000	19.0	778.9	4.9%	Seq
WSD [82]	20	1	348,639	5.1	25.4	0.6%	Seq
SWaT [45]	1	1	419,919	27.0	1876.0	12.1%	Seq
OPP [56]	27	1	449,396	1.4	653.4	6.4%	Seq
MGAB [66]	8	1	780,000	9.7	20.0	0.2%	Seq
SMD [60]	33	1	791,393	2.4	173.7	2.0%	Seq
LTDB [22]	8	1	800,000	127.5	144.5	18.6%	Seq
IOPS [31]	15	1	1,048,682	25.6	48.7	1.3%	Seq
Exathlon [32]	30	1	1,324,295	3.1	1577.3	11.0%	Seq
SVDB [24]	20	1	3,016,800	36.4	292.5	3.6%	Seq
UCR [72]	70	1	3,806,932	1.0	198.9	0.6%	P&Seq
MITDB [22]	7	1	4,400,000	68.7	451.9	4.2%	Seq

C Model details

C.1 LLMs

To evaluate whether our proposed strategies generalize across different LLMs, we conduct experiments using two open-source LLMs and two commercial API-based LLMs. For the open-source models, we employ InternVL2-Llama3-76B and Qwen2.5-VL-72B-Instruct. For the API-based models, we use Gemini-1.5-Flash by Google and GPT-4o by OpenAI.

The open-source models are hosted on an A100 4-GPU machine using the lmdeploy library, and queries are issued locally through this setup. For Gemini-1.5-Flash and GPT-4o, we directly send

requests to their respective APIs without using multi-threading, in order to accurately measure per-query latency.

Intern-VLM [15, 14], also known as InternVL2, is an open-source multimodal LLM designed to close the performance gap between open and commercial models in multimodal understanding. It combines a powerful vision encoder, InternViT, with support for dynamic high-resolution processing (up to 4K), and a bilingual training corpus. The language component is initialized with Hermes-2-Theta-LLaMA3-70B. InternVL2 achieves state-of-the-art results on 8 out of 18 benchmarks, outperforming some proprietary models in tasks like chart understanding.

In the AnomLLM experiments, performance with and without visual input was assessed using a trivial image. The model achieved an MMLU-Pro [68] score of 52.95 without the image and 53.26 with the image. This suggests that visual input does not negatively affect language performance. It is also noted that the slightly lower MMLU score is attributable to the Hermes-based language backbone, which underperforms compared to Meta’s official LLaMA3-70B-Instruct.

Qwen [9], specifically Qwen2.5-VL-72B-Instruct, is a large-scale multimodal language model developed by Alibaba Group. It builds upon the Qwen2.5 architecture and integrates a visual encoder to support image-text understanding tasks. The model features a 72-billion parameter transformer-based language backbone and leverages a high-resolution vision module. Pretrained on a mixture of web-scale bilingual corpora and diverse vision-language data, Qwen2.5-VL is optimized for instruction-following scenarios with multimodal inputs.

Qwen2.5-VL-72B-Instruct demonstrated strong reasoning capabilities, maintaining consistent performance both with and without visual input. For example, it achieved an MMLU-Pro score of 71.2 [10], compared to GPT-4o’s 72.6. On the MathVision_FULL benchmark, Qwen scored 38.1 while GPT-4o scored 30.4—suggesting a slight advantage for Qwen when leveraging multimodal context. These results highlight the robustness of Qwen’s visual-language integration and training pipeline.

Gemini-1.5-Flash [65] is a proprietary multimodal language model developed by Google, optimized for high-throughput and cost-efficient inference. It supports long-context processing and performs well across a wide range of multimodal tasks such as image understanding, classification, summarization, and content generation from visual, audio, or video inputs. Despite its speed-oriented design, it maintains competitive quality comparable to other Gemini Pro models while significantly reducing operational cost.

In the AnomLLM study, a small white image (10×10 pixels) was added to text prompts to test whether including visual inputs would affect model performance. The MMLU-Pro score remained stable—59.12 without image and 59.23 with image—indicating that the same model backend is likely used regardless of input modality, and that vision input does not degrade text reasoning capabilities.

GPT-4o [49], specifically the gpt-4o-2024-08-06 version, is OpenAI’s latest flagship multimodal model that unifies vision and text understanding. Unlike earlier versions that relied on separate encoders for different modalities, GPT-4o adopts a fully integrated transformer capable of jointly processing and reasoning across modalities with native support.

In the AnomLLM experiments, GPT-4o showed robust performance on both textual and multimodal benchmarks. Its MMLU-Pro score was 72.6 and GPQA score was 53.6. GPT-4o stands out for its balanced strength across diverse tasks and its ability to seamlessly integrate visual context without requiring explicit prompting strategies or auxiliary modules.

C.2 Our strategies

Statistical time-series decomposition. To compare with LLM-based TSAD approaches, we conducted experiments using conventional statistical decomposition techniques. Specifically, we employed the `seasonal_decompose` function from the `statsmodels` package, which decomposes a univariate time series into three components: trend, seasonality, and residual. This method assumes that the observed signal consists of systematic patterns combined with random noise, and supports both additive and multiplicative models. Following prior work [44], we adopt the additive model, defined as:

$$Y[t] = T[t] + S[t] + e[t],$$

where $Y[t]$ is the observed value at time t , $T[t]$ is the trend component, $S[t]$ is the seasonal component, and $e[t]$ is the residual.

The decomposition process uses moving averages to isolate the seasonal component, with the period either inferred from the time index or specified manually. When `two_sided=True`, a centered moving average is used; when `False`, the filter is causal, relying only on past values.

This method yields a structured separation of the time series into long-term trends, recurring seasonal patterns, and residuals. Anomalies are detected by analyzing the residuals, which capture deviations unexplained by the expected structure.

Our prompts. In our experiments, we conducted tests using both 0shot-Text and 0shot-Text-Vision while providing the corresponding indexes. We used the prompt shown in Figure 13 for these experiments.

<p>Our (0shot-Text-Vision)</p> <hr/> <p>I will provide you with time-series data recorded at hourly intervals, along with a plotted time-series image. Here is time-series data in (index, value) format:</p> <pre><history> (index, value) (index, value) (index, value) ... </history></pre> <p>Assume there are up to 5 anomalies.</p> <p>Detect ranges of anomalies in this time series, in terms of the index of time-series data, considering the plotted image. List one by one, in JSON format. If there are no anomalies, answer with an empty list []. Do not say anything other than the answer.</p> <hr/> <p>Response: [(start, end), (start, end), ...]</p>
<p>Our (0shot-Text)</p> <hr/> <p>I will provide you with time-series data recorded at regular intervals. Here is time-series data in (index, value) format:</p> <pre><history> (index, value) (index, value) (index, value) ... </history></pre> <p>Assume there are up to 5 anomalies.</p> <p>Detect ranges of anomalies in this time series, in terms of the index of time-series data. List one by one, in JSON format. If there are no anomalies, answer with an empty list []. Do not say anything other than the answer.</p> <hr/> <p>Response: [(start, end), (start, end), ...]</p>

Figure 13: The prompts used in the experiments: 0shot Text-Vision and 0shot Text.

C.3 Conventional non-LLM-based TSAD Methods in Section 5.1

(i) ML baseline

- **SR** [54] first applies the Fourier Transform to the input data and then computes the spectral residual from the log-amplitude spectrum. This modified spectrum is converted back to the time domain using the Inverse Fourier Transform, resulting in a saliency map. Anomaly scores are derived by measuring how much each value in the saliency map deviates from its moving average.
- **IForest** [42] builds a binary tree where the length of the path from the root to a given node indicates the likelihood of an anomaly—shorter paths imply a higher chance that the point is anomalous.
- **Sub-PCA** [2] projects subsequences into a lower-dimensional subspace, and large deviations from this projection space are treated as anomalies due to the breakdown of linear assumptions.
- **KMeans-AD** [78] assigns each subsequence to the nearest cluster centroid using k-means and then calculates anomaly scores as the distance between the subsequence and its assigned centroid.
- **MatrixProfile** [80] detects anomalies by identifying subsequences with unusually large distances to their nearest neighbors, thus uncovering patterns that differ from the rest of the time series.

(ii) DL baseline

- **CNN** [47] trains a convolutional network to predict future time steps from recent observations, and the anomaly score is derived from the prediction error at each point.
- **USAD** [7] combines reconstruction and adversarial loss from two autoencoders, where discrepancies between input and reconstruction, amplified by a discriminator, yield the anomaly score.
- **AnomalyTransformer** [73] introduces an Anomaly-Attention mechanism that captures temporal dependencies and assigns anomaly scores based on deviations in attention-based associations.
- **OmniAnomaly** [60] uses a stochastic recurrent neural network with variational inference and normalizing flows to learn representations of normal patterns, identifying anomalies via reconstruction probabilities.
- **AutoEncoder** [57] projects vector to the lower-dimensional latent space and reconstruct it through the encoding-decoding phase, where anomalies are typically characterized by evident reconstruction deviations.
- **TimesNet** [71] adopts a general-purpose block for multivariate time series that adaptively learns multiple periodicities via a hierarchical temporal convolutional structure.
- **FITS** [76] interpolates time series into the complex frequency domain to detect anomalies, leveraging the efficiency of frequency-domain manipulation with minimal parameters.

(iii) Foundation Model-based Method

- **Chronos** [6] tokenizes time series values through scaling and quantization into discrete tokens and trains a T5 model using a standard cross-entropy loss for anomaly detection.
- **TimesFM** [17] pretrains a decoder-style attention model using input patching on a large-scale time series corpus to learn general representations for various forecasting and detection tasks.
- **Lag-Llama** [52] introduces a decoder-only transformer architecture that conditions on lagged inputs, making it suitable for probabilistic forecasting and anomaly detection on univariate time series.
- **MOMENT** [23] is a T5-based encoder model pre-trained using masked time-series modeling to reconstruct masked values, enabling fine-tuning for downstream anomaly detection tasks.

(iv) Thresholding Method

- **Precentile** [6] selects a fixed quantile (e.g., top 5%) of the anomaly scores as the threshold to label outliers, assuming the score distribution reflects normal vs. abnormal points.

- **MAD** [27] computes the median absolute deviation to identify outliers, offering robustness against extreme values and non-Gaussian distributions.
- **EVT-POT** [58, 62] fits the tail of the anomaly score distribution with the Generalized Pareto Distribution via Peaks Over Threshold, enabling dynamic anomaly thresholding.

D Experiment details

D.1 Understanding experiments in Section 2.1

Determining whether anomaly exists
Determine whether this time series contains any anomalies. Respond with exactly 'True' if anomalies are present, or 'False' otherwise. Output only 'True' or 'False'.
Response: 'True' or 'False'

Figure 14: A prompt that determines the presence of anomalies and outputs a binary result.

To evaluate how well LLMs understand different types of anomalies in the TSAD setting, we design an experiment using datasets that cover four representative anomaly types: trend shift, frequency change, context-deviating point, and out-of-range anomalies, as proposed in AnomLLM. The dataset consists of 1,600 time series samples, with 400 instances per anomaly type. We convert the original interval-based labels into binary labels by marking whether any anomaly interval is present in a given time series, effectively framing the task as instance-level TSAD without requiring localization. Accordingly, the prompt output format is modified from interval prediction to a binary decision, as illustrated in Figure 14.

The experimental setup follows the same configuration as the TSAD task in AnomLLM, with the only modification being the binary output format. We adopt the F1-Macro score to fairly assess performance across both normal and anomalous classes. As a lower-bound reference, we include a constant classifier baseline to delineate the threshold for meaningful predictions. The experiment leverages the 21 prompt variants proposed in AnomLLM, with detailed descriptions provided in Table 5, and full dataset statistics presented in Table 6.

D.2 Localization experiments in Section 2.2

A 0shot-Vision prompt for image sequences with ground truth labels
Detect ranges of anomalies, highlighted by blue-colored regions, in this time series, in terms of the x-axis coordinate.
List one by one, in JSON format. If there are no anomalies, answer with an empty list [].
Output template: [{"start": number, "end": number }, ...]
Response: [{"start": number, "end": number }, ...]

Figure 15: The 0shot-vision prompt for localization-only experiment.

This section provides detailed information about the experiment designed to evaluate how well the model localizes anomalies during TSAD using AnomLLM. The dataset and evaluation protocol follow the AnomLLM benchmark. To minimize performance degradation caused by failures in understanding, we explicitly provide the ground-truth anomaly segments during the detection process for each sample as shown in Figure 9.

Specifically, this experiment uses the 0-shot vision-based prompt, which demonstrated strong performance in the original AnomLLM benchmark. When presenting the image, the ground-truth

anomaly segments are visually highlighted within the image to clearly indicate the anomalous regions. TSAD is then performed using a prompt (Figure 15) that explicitly instructs the model to utilize this information.

D.3 Counting experiments in Section 3.2

Counting prompt
Below is a numerical sequence: {selected_series}
Count the total number of elements in this sequence. Please return only the count as a single integer without any additional text or explanation.
Response: integer

Figure 16: The prompt used in the experiment to count elements in a given sequence.

This section offers additional details on the experiment introduced earlier. First, we conduct a sequence-length counting task by randomly sampling 100 sequences from the AnomLLM benchmark. Sequence lengths range from 1 to 1,000 and are binned into ten 100-step intervals (1–100, 101–200, ..., 901–1,000), with exactly ten sequences drawn from each interval to ensure a uniform distribution. We test four large language models and four different separator tokens used to delimit values within a sequence. Each model is instructed to return a single integer representing the sequence length. As shown in Figure 16, The prompt template is identical to the one adopted in 0shot-text variant of AnomLLM, except that the desired output is replaced with the sequence length.

D.4 Time-series decomposition experiments in Section 4.2

This section provides further details on our evaluation of LLMs’ ability to perform time-series decomposition without relying on external tools. We consider two subtasks: (1) identifying whether a time-series contains trend and/or seasonality components, and (2) generating the corresponding component sequences.

For the detection task, we constructed prompts that directly asked whether the given time-series contains trend or seasonal structure. Example prompts are provided in Figure 17. We used synthetic sequences with known decomposition structure sampled from AnomLLM as ground truth. As a baseline, we apply statistical decomposition and use a threshold to determine whether each component is present in a time series. The algorithm is shown in Algorithm 1. Specifically, a component is considered present if its amplitude exceeds a certain percentage of the original series’ value range. For example, with a threshold set to 10%, we compute the range of the original time series (i.e., max minus min), and if the amplitude of the seasonal component exceeds 10% of this range, we classify the series as containing seasonality. This threshold-based rule transforms the continuous decomposition output into a binary detection result. A comparison of F1 scores across models is presented in Figure 6(a)

Specifically, we used the following hyperparameters.

- thresh_trend=0.57
- thresh_seasonal=0.1
- thresh_resid=0.15

For the generation task, models were asked to output the trend and seasonality as separate numerical sequences given the original input. Prompt examples for this task are shown in Figure 17. While some responses preserved global shape, most models failed to accurately reproduce the true components. In particular, errors increased with sequence length, and artifacts such as baseline drift and amplitude attenuation were frequently observed. A comparison of mean absolute errors across models is presented in Figure 6(b), and representative examples are illustrated in Figure 6(c).

<p>Component detection prompt</p> <hr/> <p>Time series data can typically be decomposed into three main components such as Trend, Seasonality, and Residuals. In this analysis, Residuals are further divided into Noise and Anomalies.</p> <p>The four components exhibit the following characteristics: Trend: A long-term directional pattern or consistent upward/downward movement. Seasonality: Regular and predictable cycles repeating at consistent intervals. Residuals: The remaining noise or anomalies in the data.</p> <p>Below is the given time series data: {time_series}</p> <p>Analyze the provided time series carefully. Determine if each of these components (Trend, Seasonality, Residuals) is present (1) or absent (0).</p> <p>Provide your answer strictly in the following JSON format: { "Trend": 0 or 1, "Seasonality": 0 or 1, "Residuals": 0 or 1 }</p> <hr/> <p>Response: { "Trend": 0 or 1, "Seasonality": 0 or 1, "Residuals": 0 or 1 }</p>
<p>Component generation prompt</p> <hr/> <p>Time series data can typically be decomposed into three main components: Trend, Seasonality, and Residual.</p> <p>The three components exhibit the following characteristics: Trend: A long-term directional pattern or consistent upward/downward movement. Seasonality: Regular and predictable cycles repeating at consistent intervals. Residual: The remaining noise or anomalies in the data.</p> <p>Below is the given time series data: {sampled_series}</p> <p>CRITICAL INSTRUCTION: Extract ONLY the trend component from this time series.</p> <p>EXACT LENGTH REQUIREMENT: Your output MUST contain EXACTLY {num_samples} values one trend value for EACH value in the original time series.</p> <p>DECIMAL PRECISION: Maintain the same level of precision as the original data (2 decimal places). Example: If original values are like "-0.63", "-0.57", trend values should also have 2 decimal places like "-0.62", "-0.61".</p> <p>Count the values in the original data carefully. The original has exactly {num_samples} values, so your trend component must also have exactly num_samples values, no more and no less.</p> <p>Return your answer of {component} as space-separated numbers in the following JSON format: { "{component}": "value1 value2 value3 ... value{num_samples}" }</p> <p>DO NOT abbreviate or shorten the output. Include EVERY single trend value with 2 decimal places.</p> <hr/> <p>Response: "{component}": "value1 value2 value3 ... value{num_samples}"</p>

Figure 17: Component detection and generation prompts used in the decomposition experiment.

D.5 Index-free vs. Index-aware experiments in Section 4.2

Prior works have often favored the index-free approach, as it requires only a simple list of values that compose the sequence, resulting in a short and concise format. This simplicity can make it easier for LLMs to understand the sequence, potentially contributing positively to task performance. However,

Algorithm 1 Component Detection in Time Series

Require: Time series sequence, seasonality period, Threshold values $thresh_{trend}$, $thresh_{seasonal}$, $thresh_{resid}$

Ensure: Binary indicators for trend, seasonal, and residual components

```
1: function DETECTCOMPONENTS( $seq, period, thresh_{trend}, thresh_{seasonal}, thresh_{resid}$ )
2:    $decomposition \leftarrow$  SeasonalDecompose( $seq, model = \text{"additive"}, period = period$ )
3:    $tr \leftarrow decomposition.trend$ 
4:    $se \leftarrow decomposition.seasonal$ 
5:    $re \leftarrow decomposition.resid$ 
6:    $p_{tr} \leftarrow \mathbf{1}(\max(|tr|) \geq thresh_{trend})$   $\triangleright$  Binary indicator for trend
7:    $p_{se} \leftarrow \mathbf{1}(\max(|se|) \geq thresh_{seasonal})$   $\triangleright$  Binary indicator for seasonal
8:    $p_{re} \leftarrow \mathbf{1}(\max(|re|) \geq thresh_{resid})$   $\triangleright$  Binary indicator for residual
9:   return  $p_{tr}, p_{se}, p_{re}$ 
10: end function
```

TSAD with GT labels with text sequence only (Index-aware)

... (968,-0.03), (969,-0.1), (970,0.12), (971,0.1), (972,0.24), (973,0.11), (974,0.03), (975,-0.02), (976,0.09), (977,0.14), (978,-0.03), (979,-0.01), (980,-0.18), (981,0.06), (982,-0.0), (983,0.14), (984,0.74), (985,0.73), (986,0.78), (987,0.56), (988,0.98), (989,0.78), (990,0.7), (991,0.8), (992,0.64), (993,0.82), (994,0.69), (995,0.55), (996,0.51), (997,0.9), (998,0.78), (999,0.58)

Assume there are up to 5 anomalies. Detect the ranges of anomalies in this time series based on the x-axis coordinate, where the x-axis coordinate refers to the x value in (x, y) data. The patterns of anomalies are as follows.

anomaly data 1: 0.07 -0.4 -0.71 -0.8 -0.58 -0.46

anomaly data 2: 0.12 0.46 0.75 0.7 0.82 0.65

anomaly data 3: 0.14 0.74 0.73 0.78 0.56 0.98 0.78 0.7 0.8 0.64 0.82 0.69 0.55 0.51 0.9 0.78 0.58

List one by one, in JSON format. If there are no anomalies, answer with an empty list [].

Output template: [{"start": ..., "end": ...}, {"start": ..., "end": ...}]

Response: json [{"start": 414, "end": 460}, {"start": 215, "end": 221}, {"start": 984, "end": 999}]

Figure 18: A successful case in querying index-aware prompts for text-only TSAD with ground-truth labels.

in TSAD, the absence of index information introduces a significant drawback: when models are required to output anomaly intervals, they must internally perform a counting task to determine positions.

In contrast, the index-aware approach explicitly includes index-value pairs, requiring additional separator tokens and leading to longer and more complex input formats. This added complexity can potentially degrade task accuracy, especially when the context window becomes too long.

To evaluate the trade-offs between these two approaches, we conducted an experiment designed to isolate the effect of index formatting. To control for other variables, we only used text-based sequences without any accompanying image sequences. Similar to the localization experiment in Section 2.2, we provided ground-truth anomaly labels and compared localization performance under two conditions: with and without index information.

When using only text sequences, we included the ground-truth anomaly segments in the prompt and designed the task such that the model had to identify these anomalies in the sequence as shown in Figure 18.

As a result, consistent with our earlier discussion, we found that even though the index-aware prompts were longer and more complex, they outperformed index-free prompts. The benefit of eliminating the need for internal counting outweighed the cost of increased input length. In the case of Qwen,

Unnecessary code generation and reasoning outputs when using index-free prompts.

Response: To detect anomalies, we compute the mean (μ) and standard deviation (σ) of the series, and declare any point more than 3σ away from μ an anomaly.

Steps:

1. Compute the mean (μ).
2. Compute the standard deviation (σ).
3. Mark points further than 3σ from μ as anomalies.

Let's calculate these values and identify the anomalies.

Calculations:

1. **Mean (μ)**: The average value of the dataset.
2. **Standard Deviation (σ)**: A measure of the amount of variation or dispersion of a set of values.

```
python
import numpy as np

data = [0.05, -0.02, -0.06, -0.01, 0.14, 0.12, -0.04, 0.05, -0.04, 0.1, 0.03, 0.02, 0.1, -0.0, 0.1, -0.06,
-0.01, 0.09, -0.1, ...

mean = np.mean(data)
std_dev = np.std(data)
anomaly_threshold = 3 * std_dev

anomalies = []
for i, value in enumerate(data):
    if abs(value - mean) > anomaly_threshold:
        anomalies.append(i)

# Grouping anomalies into ranges
anomaly_ranges = []
if anomalies:
    start = anomalies[0]
    for i in range(1, len(anomalies)):
        if anomalies[i] != anomalies[i - 1] + 1:
            anomaly_ranges.append("start": start, "end": anomalies[i - 1])
            start = anomalies[i]
    anomaly_ranges.append({"start": start, "end": anomalies[-1]})

print(anomaly_ranges)
### Output: [{"start": 18, "end": 18}, {"start": 25, "end": 25}, {"start": 27, "end": 27}, ...
```

Figure 19: A failure case in querying index-free prompts for text-only TSAD with ground-truth labels. The quantitative evaluation results of the index-free and index-aware prompts are presented in Section 4.2.

index-free prompts occasionally resulted in the generation of unnecessary code or reasoning steps (Figure 19), suggesting that the inherent difficulty of the task without index cues may be responsible for such verbose outputs.

D.6 Context-aided TSAD experiments in Section 5.2

Prompt and Response of ETTh2 example
<p>I will provide you with time-series data recorded at hourly intervals, along with a plotted time-series image. Here is time-series data in (timestamp, value) format:</p> <pre><history> (2016-08-20 01:00:00, 46) (2016-08-20 02:00:00, 45) (2016-08-20 03:00:00, 44) ... (2016-09-11 08:00:00, 34) (2016-09-11 09:00:00, 35) </history></pre> <p>Here is context to understand the time series.</p> <p>The following event occurred in given period. A time-series pattern that can be explained by this event not considered an anomaly. * A flat time-series pattern may simply result from handling missing values caused by system shutdowns.</p> <p>Assume there are up to 3 anomalies.</p> <p>Detect ranges of anomalies, which are not explained by context in this time series, in terms of the timestamp of time-series data, considering the plotted image.</p> <p>Step1) Present possible anomalies with reason.</p> <p>Step2) Remove explainable anomalies by context.</p> <p>Step3) List one by one, in JSON format. If there are no anomalies, answer with an empty list []. Do not say anything other than the answer.</p> <p>Output template: [{"start timestamp": ..., "end timestamp": ...}, {"start timestamp": ..., "end timestamp": ...}]...</p> <p>Please answer to all steps.</p>
<p>Response:</p> <p>Step1) Present possible anomalies with reason.</p> <ol style="list-style-type: none">1. From 2016-08-31 00:00:00 to 2016-08-31 23:00:00, the time series shows a flat pattern at a value of 33, which could be considered an anomaly due to its deviation from the usual pattern of fluctuations.2. From 2016-09-06 22:00:00 to 2016-09-07 08:00:00, the time series shows a sudden drop to a lower value range, which could be considered an anomaly due to its deviation from the usual pattern. <p>Step2) Remove explainable anomalies by context.</p> <ol style="list-style-type: none">1. The flat pattern from 2016-08-31 00:00:00 to 2016-08-31 23:00:00 can be explained by the context provided, which mentions that flat patterns may result from handling missing values due to system shutdowns. Therefore, this is not considered an anomaly.2. The sudden drop from 2016-09-06 22:00:00 to 2016-09-07 08:00:00 does not have an explanation provided by the context, so it remains a potential anomaly. <p>Step3) List one by one, in JSON format.</p> <pre>““json [{"start timestamp": "2016-09-06 22:00:00", "end timestamp": "2016-09-07 08:00:00"}]““</pre>

Figure 20: The prompt and response of qualitative result on (b) potential range anomalies in ETTh2.

Conventional TSAD has relied heavily on numerical sequences, using statistical or neural network-based models. While there have been attempts to incorporate textual information into these models, such approaches often depend on synthetic datasets due to the difficulty of collecting paired text

and time-series data. As a result, they fall short in terms of generality and reasoning capabilities compared to LLMs. Thus, without other modality, these conventional models generally depend on an assumption in which anomalies are defined by sparsity or abrupt changes in the time-series patterns.

However, in real-world applications, not all sparse or abrupt patterns are of actual concern. For instance, in e-commerce platforms, promotional events can cause a sudden surge in product sales. Traditional models may flag such a change as an anomaly, despite it being an expected and well-known outcome. While it is possible to exclude known time windows using rule-based filters, this quickly becomes labor-intensive and inflexible when many events or patterns need to be handled. In contrast, simply describing these contexts in natural language would be a far more scalable and user-friendly solution.

In this situation, the reasoning ability of LLMs to interpret natural language context offers a valuable control interface for TSAD. To explore this capability, we conducted a study using real-world datasets—ETTh2 and Electricity. We first ran eight conventional anomaly detection models on these datasets and extracted time segments that were commonly detected as anomalies. We then generated domain-informed contextual descriptions for each of these segments, simulating plausible scenarios such as a known event causing a sudden drop, or a specific pattern being aligned with external factors. These contexts provided time-related or value-specific cues in natural language.

Using the prompt as shown in Figure 20, we evaluated the anomaly decisions of the LLM both with and without these contextual inputs. Our results show that the LLM could effectively integrate temporal cues, numerical patterns, and contextual knowledge to suppress false positives—i.e., anomalies detected by conventional models that users would not consider problematic.

Unlike conventional models that rigidly detect anomalies based on pattern shifts alone, LLM-based context-aided TSAD can reinterpret these shifts in light of the user’s intent and domain knowledge. This allows for “justifiable detection,” where the LLM not only detects changes but reasons whether those changes are relevant anomalies given the context. The LLM thus transforms anomaly detection into an interactive, user-aligned process that prioritizes interpretability and flexibility.

This paradigm shift points to several promising directions for future work: improving generalization across domains, ensuring real-time applicability of context-aware detection, and defining minimal requirements for effective contextual prompts. Notably, because the meaning of “anomaly” can vary depending on the user’s objective or operational context, LLMs open the door to a new class of objective-aware anomaly detection systems—systems that adapt dynamically to changing definitions of what matters, rather than adhering to a fixed statistical criterion.

Table 8: TSAD results on the evaluation set of TSB-AD-U benchmark. The summarized results are presented in Section 5.1.

Method	Thresholding	Standard Metrics			Affiliation Metrics			Inference Time
		Prec.	Recall	F1	Prec.	Recall	F1	
AnomLLM (GPT-4o)	–	12.79	11.96	9.32	39.59	30.54	31.59	25.52
AnomalyTransformer	Percentile	15.21	40.35	13.91	54.04	86.67	64.13	8.49
FITS	MAD	13.88	38.54	16.54	56.03	86.48	65.73	4.05
TimesNet	MAD	14.40	39.63	17.71	56.87	85.48	66.02	5.91
MatrixProfile	MAD	19.73	28.82	18.29	61.87	59.30	54.80	33.58
MOMENT-ZS	Percentile	26.07	27.35	20.42	65.33	71.04	63.82	8.90
OmniAnomaly	MAD	27.91	30.81	20.46	63.14	83.35	68.08	2.66
KMeansAD-U	Percentile	31.58	25.28	22.14	66.57	50.01	50.77	1.28
AutoEncoder	MAD	27.12	27.44	22.27	66.21	60.82	57.25	3.16
Lag-Llama	MAD	24.17	45.67	23.54	65.20	92.70	74.54	118.85
Sub-PCA	MAD	26.55	28.00	23.72	62.37	54.31	54.74	0.18
USAD	MAD	23.92	34.54	24.46	59.84	61.60	55.24	1.57
TimesFM	MAD	25.47	52.18	24.98	66.32	94.65	75.72	130.09
Chronos	MAD	23.58	<u>55.65</u>	25.08	64.87	96.33	<u>75.90</u>	69.10
IForest	MAD	<u>33.99</u>	26.62	26.79	<u>68.24</u>	68.44	64.72	0.52
SR	MAD	32.61	40.77	30.57	66.80	<u>95.11</u>	75.94	0.02
Our (GPT-4o)	–	47.60	31.88	34.20	76.10	81.49	74.64	140.63

E Full benchmark results

Gemini-1.5-Flash			Standard			Affiliation		
Datasets	Code	Prompt	Prec.	Recall	F1	Prec.	Recall	F1
Trend	A	1shot-vision-cot	41.07	40.88	39.55	50.91	53.18	51.82
	B	1shot-vision-calc	63.33	63.63	62.75	67.09	67.43	67.25
	C	1shot-vision-dyscalc	62.01	62.22	61.86	64.12	64.37	64.24
	D	1shot-vision	57.59	57.73	56.75	62.41	62.81	62.59
	E	0shot-vision-cot	57.93	57.94	57.93	57.99	58.00	57.99
	F	0shot-vision-calc	59.70	59.44	59.48	60.15	60.20	60.18
	G	0shot-vision-dyscalc	59.08	59.03	58.95	59.43	59.47	59.45
	H	0shot-vision	59.77	59.57	59.60	60.17	60.22	60.19
	I	1shot-text-s0.3-cot	5.64	7.96	6.32	19.15	24.30	21.23
	J	1shot-text-s0.3	2.03	2.15	1.88	11.72	17.19	13.70
	K	0shot-text-s0.3-tpd	0.00	0.00	0.00	3.03	5.81	3.97
	L	0shot-text-s0.3-pap	0.00	0.00	0.00	3.37	7.25	4.56
	M	0shot-text-s0.3-dyscalc	0.00	0.00	0.00	3.90	7.85	5.20
	N	0shot-text-s0.3-csv	0.00	0.00	0.00	3.15	7.16	4.31
	O	0shot-text-s0.3-cot-tpd	3.25	3.25	3.25	7.83	10.62	8.88
	P	0shot-text-s0.3-cot-pap	0.25	0.25	0.25	3.26	5.66	4.07
	Q	0shot-text-s0.3-cot-csv	0.50	0.50	0.50	5.51	9.70	6.88
	R	0shot-text-s0.3-cot	0.52	1.01	0.63	7.83	13.90	9.94
	S	0shot-text-s0.3-calc	0.00	0.00	0.00	4.59	8.98	6.06
	T	0shot-text-s0.3	2.12	2.58	2.07	12.28	17.76	14.31
	U	0shot-text	0.00	0.00	0.00	3.99	7.79	5.27
	Our	0shot-text	65.38	61.87	62.52	67.47	70.49	68.28
	Our	0shot-text-vision (w/o value)	70.87	71.07	70.30	88.41	91.37	89.62
	Our	0shot-text-vision (w/o index)	74.95	75.70	74.38	85.63	86.40	85.95
	Our	0shot-text-vision (w/o deseason)	58.76	58.45	58.47	59.69	61.47	60.27
	Our	0shot-text-vision	82.90	81.81	81.28	90.08	92.70	91.11
	A	1shot-vision-cot	14.29	16.89	13.80	54.19	41.68	43.93
	B	1shot-vision-calc	23.10	20.50	20.73	34.76	24.87	27.46
	C	1shot-vision-dyscalc	23.73	21.83	21.61	36.35	24.37	27.57

	D	1shot-vision	21.78	21.31	20.52	35.84	25.89	28.46
	E	0shot-vision-cot	12.65	12.71	12.45	14.16	12.52	12.98
	F	0shot-vision-calc	17.82	19.68	18.14	23.65	17.80	19.39
	G	0shot-vision-dyscalc	16.71	17.84	16.83	21.44	16.58	17.88
	H	0shot-vision	13.56	13.77	13.54	15.47	13.15	13.76
	I	1shot-text-s0.3-cot	12.81	17.30	12.31	56.67	50.63	51.30
	J	1shot-text-s0.3	10.20	14.73	10.45	56.37	52.78	52.61
	K	0shot-text-s0.3-tpd	2.57	1.57	1.63	39.54	29.61	32.22
	L	0shot-text-s0.3-pap	1.97	0.73	0.94	33.36	22.71	25.54
	M	0shot-text-s0.3-dyscalc	3.74	3.24	2.91	36.95	28.19	30.32
	N	0shot-text-s0.3-csv	2.77	2.21	2.08	46.48	33.66	37.03
	O	0shot-text-s0.3-cot-tpd	2.61	2.37	1.97	25.96	19.14	21.00
	P	0shot-text-s0.3-cot-pap	3.92	2.35	2.61	29.46	19.21	22.12
	Q	0shot-text-s0.3-cot-csv	2.55	2.34	1.94	30.81	23.37	25.37
	R	0shot-text-s0.3-cot	3.46	4.78	3.36	21.76	19.93	19.99
	S	0shot-text-s0.3-calc	3.29	2.61	2.51	36.76	27.58	29.91
	T	0shot-text-s0.3	11.28	14.24	10.91	58.42	52.67	53.59
	U	0shot-text	4.01	4.30	3.63	39.24	31.84	33.65
	Our	0shot-text	37.92	24.04	25.75	62.71	52.37	54.95
	Our	0shot-text-vision (w/o value)	30.17	50.64	33.71	<u>80.53</u>	70.80	73.25
	Our	0shot-text-vision (w/o index)	30.20	45.50	32.56	77.49	67.14	69.69
	Our	0shot-text-vision (w/o deseason)	48.87	33.96	<u>36.87</u>	55.99	41.00	45.47
	Our	0shot-text-vision	<u>46.23</u>	<u>47.13</u>	39.98	81.57	<u>70.16</u>	<u>73.03</u>
Point	A	1shot-vision-cot	43.09	67.37	49.31	82.62	83.34	82.61
	B	1shot-vision-calc	55.95	77.36	61.54	94.88	94.55	94.17
	C	1shot-vision-dyscalc	54.29	78.90	60.44	95.18	96.15	95.25
	D	1shot-vision	53.54	78.76	59.96	93.42	93.83	93.19
	E	0shot-vision-cot	51.87	84.03	59.20	91.78	89.43	89.75
	F	0shot-vision-calc	51.56	87.56	59.74	94.28	95.88	94.63
	G	0shot-vision-dyscalc	51.40	<u>88.72</u>	59.53	94.18	95.87	94.52
	H	0shot-vision	52.35	90.69	60.90	94.22	95.46	94.38
	I	1shot-text-s0.3-cot	8.19	9.11	8.05	45.60	43.79	43.26
	J	1shot-text-s0.3	3.63	4.20	3.61	43.49	41.13	40.93
	K	0shot-text-s0.3-tpd	1.83	1.76	1.77	22.66	20.71	20.94
	L	0shot-text-s0.3-pap	0.00	0.00	0.00	15.58	14.10	14.28
	M	0shot-text-s0.3-dyscalc	0.20	0.10	0.12	18.22	18.78	18.08
	N	0shot-text-s0.3-csv	3.81	2.85	2.74	34.27	29.22	30.44
	O	0shot-text-s0.3-cot-tpd	3.89	3.79	3.76	18.41	16.85	17.14
	P	0shot-text-s0.3-cot-pap	3.31	3.51	3.30	17.68	16.21	16.44
	Q	0shot-text-s0.3-cot-csv	2.88	2.86	2.19	24.03	20.64	21.46
	R	0shot-text-s0.3-cot	2.66	3.72	2.54	19.37	20.22	19.12
	S	0shot-text-s0.3-calc	0.10	0.05	0.06	18.67	19.37	18.53
	T	0shot-text-s0.3	2.75	3.60	2.83	43.01	41.02	40.64
	U	0shot-text	1.09	1.06	0.93	23.84	24.38	23.35
	Our	0shot-text	67.99	35.56	43.39	69.75	67.17	67.79
	Our	0shot-text-vision (w/o value)	56.52	81.42	62.76	95.86	95.95	95.52
	Our	0shot-text-vision (w/o index)	57.21	76.78	62.52	93.78	96.26	94.54
	Our	0shot-text-vision (w/o deseason)	<u>87.85</u>	88.36	85.44	98.91	97.12	97.64
	Our	0shot-text-vision	90.89	74.93	<u>78.41</u>	<u>98.45</u>	<u>96.81</u>	<u>97.11</u>
	A	1shot-vision-cot	22.85	50.51	28.87	69.16	71.62	70.05
	B	1shot-vision-calc	36.28	64.25	42.40	81.41	84.05	82.40
	C	1shot-vision-dyscalc	35.52	61.50	41.15	81.49	83.06	81.82
	D	1shot-vision	33.77	63.17	40.02	80.05	82.67	81.07
	E	0shot-vision-cot	28.58	61.32	34.67	75.91	73.62	73.75
	F	0shot-vision-calc	20.96	60.42	28.02	68.04	70.94	69.07
	G	0shot-vision-dyscalc	26.94	66.70	34.00	74.17	77.29	75.34
	H	0shot-vision	32.70	74.46	40.19	79.35	83.16	80.95

I	1shot-text-s0.3-cot	12.04	14.57	12.31	46.38	49.43	46.57
J	1shot-text-s0.3	7.12	13.71	8.42	47.06	56.95	50.69
K	0shot-text-s0.3-tpd	2.30	2.63	2.13	29.22	28.56	27.96
L	0shot-text-s0.3-pap	1.97	0.88	1.08	30.66	27.80	28.09
M	0shot-text-s0.3-dyscalc	2.20	1.95	1.85	33.94	36.51	34.22
N	0shot-text-s0.3-csv	5.00	4.52	4.62	35.17	33.31	33.28
O	0shot-text-s0.3-cot-tpd	2.87	3.03	2.86	16.75	14.98	15.26
P	0shot-text-s0.3-cot-pap	6.59	6.17	6.17	25.46	22.12	23.01
Q	0shot-text-s0.3-cot-csv	2.70	2.24	2.34	16.85	15.47	15.75
R	0shot-text-s0.3-cot	4.28	4.90	4.18	20.54	22.27	20.84
S	0shot-text-s0.3-calc	2.24	1.92	1.81	35.13	36.94	34.99
T	0shot-text-s0.3	7.15	12.76	8.34	46.77	56.31	50.25
U	0shot-text	2.92	4.53	2.98	32.52	38.30	34.27
Our	0shot-text	59.97	53.65	<u>54.81</u>	64.74	67.29	65.47
Our	0shot-text-vision (w/o value)	<u>38.54</u>	<u>67.64</u>	<u>44.81</u>	83.56	86.21	84.57
Our	0shot-text-vision (w/o index)	<u>20.27</u>	<u>49.77</u>	<u>26.97</u>	61.38	68.83	64.52
Our	0shot-text-vision (w/o deseason)	<u>54.54</u>	61.60	55.60	66.94	68.64	67.60
Our	0shot-text-vision	<u>53.15</u>	<u>59.37</u>	<u>53.57</u>	66.36	68.09	66.96

InternVL2-Llama3-76B			Standard			Affiliation		
Datasets	Code	Prompt	Prec.	Recall	F1	Prec.	Recall	F1
Trend	A	1shot-vision-cot	44.90	44.90	44.29	46.92	47.32	47.08
	B	1shot-vision-calc	33.76	39.43	32.67	40.33	44.13	41.61
	C	1shot-vision-dyscalc	39.61	43.23	38.34	48.48	51.02	49.37
	D	1shot-vision	38.98	42.27	37.93	47.54	50.00	48.43
	E	0shot-vision-cot	<u>51.13</u>	51.34	<u>51.06</u>	<u>53.27</u>	54.16	<u>53.60</u>
	F	0shot-vision-calc	41.31	44.80	41.82	46.76	52.22	48.64
	G	0shot-vision-dyscalc	36.48	45.76	36.67	45.49	54.89	48.67
	H	0shot-vision	27.27	45.87	27.82	37.77	49.50	41.68
	I	1shot-text-s0.3-cot	37.44	36.65	35.74	42.01	42.32	42.14
	J	1shot-text-s0.3	30.18	31.05	29.77	36.14	37.59	36.68
	K	0shot-text-s0.3-tpd	2.06	4.73	2.30	5.75	9.78	7.09
	L	0shot-text-s0.3-pap	46.08	46.75	46.14	47.07	48.13	47.43
	M	0shot-text-s0.3-dyscalc	11.38	13.71	11.63	14.30	17.62	15.41
	N	0shot-text-s0.3-csv	0.50	0.50	0.50	5.45	9.52	6.83
	O	0shot-text-s0.3-cot-tpd	10.76	11.00	10.77	11.39	12.00	11.60
	P	0shot-text-s0.3-cot-pap	20.40	20.84	20.42	21.31	22.32	21.65
	Q	0shot-text-s0.3-cot-csv	4.52	4.75	4.54	8.09	9.73	8.66
	R	0shot-text-s0.3-cot	13.26	13.50	13.27	15.42	17.52	16.12
	S	0shot-text-s0.3-calc	11.20	12.58	11.34	12.87	14.77	13.50
	T	0shot-text-s0.3	33.58	32.66	31.95	38.68	39.79	39.07
	U	0shot-text	19.21	23.60	19.56	22.80	26.84	24.16
Our	0shot-text		32.32	<u>52.20</u>	35.72	47.85	<u>54.93</u>	50.59
Our	0shot-text-vision		56.90	66.60	58.83	68.36	74.21	70.60
	A	1shot-vision-cot	2.76	8.41	3.58	35.98	42.04	36.75
	B	1shot-vision-calc	4.55	13.45	5.19	35.04	38.20	33.78
	C	1shot-vision-dyscalc	4.11	12.28	5.37	35.10	38.41	33.73
	D	1shot-vision	4.31	15.22	4.92	38.93	44.67	38.43
	E	0shot-vision-cot	4.97	9.25	5.37	30.18	27.70	27.27
	F	0shot-vision-calc	4.68	15.55	5.41	26.42	29.19	25.80
	G	0shot-vision-dyscalc	7.16	18.74	8.88	32.78	33.72	31.01
	H	0shot-vision	6.51	17.82	8.81	35.94	35.14	33.37
	I	1shot-text-s0.3-cot	3.05	2.69	2.40	30.54	26.81	26.86
	J	1shot-text-s0.3	4.24	4.93	3.22	37.50	32.79	32.66

Freq

	K	0shot-text-s0.3-tpd	4.01	7.60	4.01	24.88	21.46	21.45
	L	0shot-text-s0.3-pap	8.36	9.29	8.38	13.77	12.68	12.77
	M	0shot-text-s0.3-dyscalc	7.15	8.20	7.22	16.55	13.84	14.39
	N	0shot-text-s0.3-csv	3.65	8.97	3.62	37.52	31.89	32.68
	O	0shot-text-s0.3-cot-tpd	3.13	3.73	3.15	10.65	8.46	8.98
	P	0shot-text-s0.3-cot-pap	3.79	3.68	3.71	6.45	5.57	5.78
	Q	0shot-text-s0.3-cot-csv	2.60	4.52	2.69	19.44	17.81	17.71
	R	0shot-text-s0.3-cot	6.51	7.71	6.72	12.89	11.98	12.06
	S	0shot-text-s0.3-calc	8.18	9.48	8.30	14.36	13.50	13.45
	T	0shot-text-s0.3	3.38	7.18	3.33	37.12	33.84	33.16
	U	0shot-text	5.66	12.39	6.04	15.82	17.30	15.53
	Our	0shot-text	20.35	48.90	19.72	60.36	63.00	57.04
	Our	0shot-text-vision	13.42	46.70	17.22	57.58	73.33	62.47
	A	1shot-vision-cot	3.87	12.65	4.61	33.66	45.65	37.30
	B	1shot-vision-calc	10.01	29.92	12.26	44.34	57.93	48.86
	C	1shot-vision-dyscalc	8.98	31.81	11.90	42.77	59.47	48.65
	D	1shot-vision	9.97	29.84	12.99	44.76	57.62	49.26
	E	0shot-vision-cot	4.35	12.95	5.47	31.51	39.00	33.61
	F	0shot-vision-calc	22.97	58.75	28.77	66.27	76.14	70.44
	G	0shot-vision-dyscalc	17.31	54.36	23.36	59.49	70.15	63.79
	H	0shot-vision	14.26	53.10	20.64	56.18	67.83	60.77
	I	1shot-text-s0.3-cot	10.20	9.98	9.43	30.63	30.69	29.49
	J	1shot-text-s0.3	8.69	9.89	7.70	32.37	32.95	31.19
	K	0shot-text-s0.3-tpd	7.00	7.00	7.00	14.89	15.08	14.88
	L	0shot-text-s0.3-pap	25.26	25.50	25.27	27.14	27.26	27.16
	M	0shot-text-s0.3-dyscalc	18.89	19.66	18.94	23.35	23.80	23.46
	N	0shot-text-s0.3-csv	11.92	13.91	11.20	40.05	38.01	37.78
	O	0shot-text-s0.3-cot-tpd	9.21	9.82	8.81	26.12	24.75	24.70
	P	0shot-text-s0.3-cot-pap	10.58	10.83	10.58	11.56	11.76	11.62
	Q	0shot-text-s0.3-cot-csv	9.00	9.00	9.00	11.69	11.68	11.63
	R	0shot-text-s0.3-cot	15.82	16.50	15.86	18.51	19.07	18.68
	S	0shot-text-s0.3-calc	21.42	23.25	21.52	25.62	26.72	25.93
	T	0shot-text-s0.3	10.21	11.34	9.96	33.19	34.47	32.60
	U	0shot-text	19.26	22.48	19.47	25.08	27.32	25.84
	Our	0shot-text	37.76	53.21	37.08	59.81	65.15	60.64
	Our	0shot-text-vision	28.86	60.61	34.92	56.23	69.93	61.22
	A	1shot-vision-cot	3.57	5.90	3.66	31.10	34.97	31.58
	B	1shot-vision-calc	19.32	35.23	21.91	60.17	65.54	61.65
	C	1shot-vision-dyscalc	17.24	34.25	20.14	59.85	65.48	61.53
	D	1shot-vision	17.79	33.39	20.35	58.16	61.04	58.53
	E	0shot-vision-cot	8.25	12.05	8.64	34.21	36.28	33.99
	F	0shot-vision-calc	27.58	56.64	32.82	72.83	78.02	74.93
	G	0shot-vision-dyscalc	4.39	13.78	5.03	32.84	39.45	34.03
	H	0shot-vision	27.08	56.67	32.58	73.52	79.03	75.84
	I	1shot-text-s0.3-cot	8.70	8.90	8.48	28.66	28.08	27.47
	J	1shot-text-s0.3	8.61	9.95	8.37	27.64	26.60	25.94
	K	0shot-text-s0.3-tpd	7.83	12.72	8.05	22.23	23.44	21.88
	L	0shot-text-s0.3-pap	25.43	27.34	25.51	28.87	29.71	29.00
	M	0shot-text-s0.3-dyscalc	7.56	16.99	7.92	21.17	25.37	22.00
	N	0shot-text-s0.3-csv	7.83	8.24	7.49	38.79	34.59	35.52
	O	0shot-text-s0.3-cot-tpd	6.40	6.46	6.34	11.74	11.09	11.12
	P	0shot-text-s0.3-cot-pap	12.58	12.40	12.37	18.20	17.09	17.31
	Q	0shot-text-s0.3-cot-csv	4.83	5.71	4.75	19.71	18.08	18.24
	R	0shot-text-s0.3-cot	9.85	10.45	9.94	16.89	17.87	17.13
	S	0shot-text-s0.3-calc	10.49	13.55	10.29	20.98	21.34	20.46
	T	0shot-text-s0.3	7.45	8.89	7.48	25.94	25.18	24.59
	U	0shot-text	8.95	27.30	9.39	27.16	36.32	29.81

	Our	0shot-text	20.27	54.87	24.33	49.22	59.77	51.89
	Our	0shot-text-vision	20.82	63.50	27.68	50.37	69.70	57.33

GPT-4o			Standard			Affiliation		
Datasets	Code	Prompt	Prec.	Recall	F1	Prec.	Recall	F1
Trend	A	1shot-vision-cot	57.50	57.50	57.50	57.50	57.50	57.50
	H	0shot-vision	57.50	57.50	57.50	57.50	57.50	57.50
	U	0shot-text	6.67	5.91	6.04	14.85	22.53	17.48
	Our	0shot-text	83.47	68.97	70.84	87.48	87.93	87.41
	Our	0shot-text-vision (w/o value)	58.13	57.82	57.90	58.23	58.21	58.22
	Our	0shot-text-vision (w/o index)	64.25	64.40	63.80	71.40	71.87	71.63
	Our	0shot-text-vision (w/o deseason)	57.50	57.50	57.50	57.50	57.50	57.50
	Our	0shot-text-vision	<u>79.12</u>	75.48	76.27	<u>79.66</u>	<u>79.43</u>	<u>79.54</u>
	A	1shot-vision-cot	15.66	17.93	16.12	28.10	22.09	23.65
	H	0shot-vision	13.68	15.69	14.27	19.02	16.13	16.98
Freq	U	0shot-text	15.01	10.54	11.20	34.05	24.71	27.40
	Our	0shot-text	50.38	27.15	29.34	72.56	60.06	62.88
	Our	0shot-text-vision (w/o value)	16.27	20.95	16.82	37.87	31.69	33.40
	Our	0shot-text-vision (w/o index)	16.80	20.87	17.06	52.02	41.44	44.39
	Our	0shot-text-vision (w/o deseason)	37.86	19.01	22.13	39.54	27.41	30.80
	Our	0shot-text-vision	<u>45.42</u>	36.09	35.16	<u>65.85</u>	<u>51.44</u>	<u>55.66</u>
	A	1shot-vision-cot	30.12	46.21	33.57	72.69	73.24	72.08
	H	0shot-vision	45.20	68.00	50.48	87.71	87.70	87.09
	U	0shot-text	33.49	32.11	32.31	73.21	73.64	72.87
	Our	0shot-text	72.50	36.66	44.00	74.48	74.86	73.52
Point	Our	0shot-text-vision (w/o value)	46.67	63.10	50.96	87.29	85.85	85.87
	Our	0shot-text-vision (w/o index)	44.10	44.11	43.30	86.06	82.98	83.65
	Our	0shot-text-vision (w/o deseason)	98.87	90.33	93.57	99.96	97.76	98.54
	Our	0shot-text-vision	<u>90.51</u>	<u>73.14</u>	<u>78.71</u>	<u>92.66</u>	<u>91.48</u>	<u>91.79</u>
	A	1shot-vision-cot	22.65	32.35	24.52	67.43	70.52	68.34
	H	0shot-vision	41.76	61.19	45.87	84.52	86.62	85.15
	U	0shot-text	23.58	21.90	21.47	63.27	61.17	61.18
	Our	0shot-text	72.56	66.75	67.40	76.20	77.96	76.43
	Our	0shot-text-vision (w/o value)	42.21	58.69	45.99	83.33	83.35	82.81
	Our	0shot-text-vision (w/o index)	41.77	46.07	42.70	84.21	83.73	83.44
Range	Our	0shot-text-vision (w/o deseason)	96.39	88.93	91.67	97.75	96.20	96.72
	Our	0shot-text-vision	<u>92.70</u>	88.95	<u>90.00</u>	<u>94.25</u>	<u>93.05</u>	<u>93.45</u>

Qwen2.5-VL-72B-Instruct			Standard			Affiliation		
Datasets	Code	Prompt	Prec.	Recall	F1	Prec.	Recall	F1
Trend	A	1shot-vision-cot	54.59	55.00	54.66	54.94	55.00	54.97
	B	1shot-vision-calc	43.78	44.79	42.79	48.13	48.69	48.37
	C	1shot-vision-dyscalc	42.98	46.53	42.82	49.49	50.76	50.03
	E	0shot-vision-cot	55.93	55.89	55.81	56.19	56.22	56.20
	F	0shot-vision-calc	61.30	60.87	60.69	62.07	62.06	62.06
	G	0shot-vision-dyscalc	60.75	60.74	60.43	61.56	61.62	61.58
	H	0shot-vision	60.19	59.95	59.79	60.64	60.64	60.64
	J	1shot-text-s0.3	45.52	45.51	45.52	47.48	47.99	47.69
	L	0shot-text-s0.3-pap	40.50	40.50	40.50	40.76	40.84	40.79
	M	0shot-text-s0.3-dyscalc	10.25	10.25	10.25	10.98	11.38	11.13

Freq	N	0shot-text-s0.3-csv	16.75	16.75	16.75	19.08	20.10	19.44
	P	0shot-text-s0.3-cot-pap	18.75	18.75	18.75	19.01	19.15	19.06
	S	0shot-text-s0.3-calc	8.81	8.80	8.81	10.03	10.88	10.33
	T	0shot-text-s0.3	46.81	46.77	46.78	48.76	49.48	49.02
	U	0shot-text	56.00	56.00	56.00	56.08	56.14	56.11
	Our	0shot-text	86.06	83.24	82.00	91.96	93.96	92.69
	Our	0shot-text-vision (w/o value)	61.38	60.66	60.77	62.70	62.86	62.77
	Our	0shot-text-vision (w/o index)	66.82	68.73	66.34	81.77	84.62	82.88
	Our	0shot-text-vision (w/o deseason)	60.80	59.36	59.68	61.37	61.27	61.31
	Our	0shot-text-vision	85.61	81.49	81.09	89.95	91.19	90.20
	A	1shot-vision-cot	5.16	7.29	5.35	16.12	14.46	14.67
	B	1shot-vision-calc	10.98	11.84	10.00	28.72	23.05	23.95
	C	1shot-vision-dyscalc	10.15	11.80	9.17	31.71	26.34	26.81
	E	0shot-vision-cot	9.33	13.26	9.93	17.26	16.61	16.33
	F	0shot-vision-calc	14.29	15.52	14.15	31.78	24.87	26.59
	G	0shot-vision-dyscalc	16.40	18.10	16.39	31.99	24.42	26.38
	H	0shot-vision	15.47	18.62	15.75	29.05	26.55	26.64
	J	1shot-text-s0.3	16.39	16.31	15.45	63.17	50.72	54.66
	L	0shot-text-s0.3-pap	6.75	6.75	6.75	7.98	7.36	7.52
	M	0shot-text-s0.3-dyscalc	3.72	3.56	3.50	19.20	14.15	15.51
	N	0shot-text-s0.3-csv	11.16	9.81	9.98	33.34	23.73	26.37
Point	P	0shot-text-s0.3-cot-pap	4.81	4.82	4.81	5.26	5.06	5.12
	S	0shot-text-s0.3-calc	5.51	4.96	4.76	19.25	13.76	15.27
	T	0shot-text-s0.3	19.30	18.79	18.14	64.73	52.41	56.38
	U	0shot-text	11.45	10.11	10.20	18.80	14.19	15.43
	Our	0shot-text	56.91	30.77	33.57	80.90	62.84	67.54
	Our	0shot-text-vision (w/o value)	23.87	28.34	22.97	61.32	51.76	53.90
	Our	0shot-text-vision (w/o index)	21.97	27.84	21.22	63.38	53.50	55.76
	Our	0shot-text-vision (w/o deseason)	42.81	28.58	31.05	47.43	37.21	40.30
	Our	0shot-text-vision	49.41	41.14	37.88	79.18	64.95	68.57
	A	1shot-vision-cot	18.47	22.57	18.98	35.89	38.85	36.88
	B	1shot-vision-calc	41.02	43.32	40.04	73.97	79.19	75.82
	C	1shot-vision-dyscalc	37.43	42.19	37.49	70.08	75.13	71.84
	E	0shot-vision-cot	25.04	32.84	26.86	44.71	46.66	45.33
	F	0shot-vision-calc	46.25	51.14	46.36	82.45	83.61	82.77
	G	0shot-vision-dyscalc	47.30	53.39	47.98	82.86	84.52	83.41
	H	0shot-vision	60.09	67.56	61.62	94.93	96.19	95.34
	J	1shot-text-s0.3	33.19	31.96	32.18	70.90	65.03	66.95
	L	0shot-text-s0.3-pap	16.25	16.25	16.25	18.21	17.68	17.83
	M	0shot-text-s0.3-dyscalc	12.62	12.19	12.27	26.96	23.93	24.80
	N	0shot-text-s0.3-csv	18.06	16.77	17.03	39.32	34.06	35.67
	P	0shot-text-s0.3-cot-pap	8.50	8.50	8.50	8.85	8.70	8.75
Range	S	0shot-text-s0.3-calc	6.61	6.36	6.41	18.27	15.88	16.57
	T	0shot-text-s0.3	29.66	29.23	29.16	68.38	62.84	64.63
	U	0shot-text	30.33	29.85	29.83	62.56	54.68	57.09
	Our	0shot-text	94.52	56.07	63.16	98.31	91.50	93.78
	Our	0shot-text-vision (w/o value)	56.83	58.55	55.57	92.67	90.89	91.23
	Our	0shot-text-vision (w/o index)	52.61	54.97	52.19	89.04	89.42	88.50
	Our	0shot-text-vision (w/o deseason)	90.61	95.63	91.74	99.36	99.53	99.36
	Our	0shot-text-vision	94.19	83.80	85.98	99.01	96.94	97.70
	A	1shot-vision-cot	30.14	31.44	29.92	51.12	51.95	51.00
	B	1shot-vision-calc	49.75	51.22	48.58	94.28	91.52	92.25
	C	1shot-vision-dyscalc	49.84	51.61	48.97	94.03	90.73	91.68
	E	0shot-vision-cot	36.46	43.52	37.61	61.41	61.55	61.27
	F	0shot-vision-calc	46.03	55.76	48.02	92.50	91.52	91.57
	G	0shot-vision-dyscalc	46.97	58.48	49.17	91.06	91.02	90.67

H	0shot-vision	47.35	57.86	49.43	93.36	92.56	92.49
J	1shot-text-s0.3	14.26	16.71	14.76	50.94	55.36	52.01
L	0shot-text-s0.3-pap	9.65	9.30	9.39	22.67	20.76	21.09
M	0shot-text-s0.3-dyscalc	2.35	2.30	2.31	8.47	7.66	7.84
N	0shot-text-s0.3-csv	9.77	9.14	9.20	32.94	28.68	29.93
P	0shot-text-s0.3-cot-pap	3.94	3.33	3.40	6.23	5.32	5.52
S	0shot-text-s0.3-calc	0.69	0.56	0.59	3.37	2.92	3.03
T	0shot-text-s0.3	11.42	13.58	11.85	46.77	51.10	47.91
U	0shot-text	6.19	6.68	5.93	45.32	44.74	43.80
Our	0shot-text	52.40	46.27	45.78	60.49	65.31	61.84
Our	0shot-text-vision (w/o value)	47.70	53.44	48.04	91.94	89.46	90.06
Our	0shot-text-vision (w/o index)	13.26	22.41	15.17	53.77	61.26	56.37
Our	0shot-text-vision (w/o deseason)	61.49	<u>66.94</u>	62.43	70.08	73.02	71.26
Our	0shot-text-vision	<u>59.79</u>	67.18	<u>61.21</u>	69.91	73.02	71.03

NeurIPS Paper Checklist

1. Claims

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

Answer: [\[Yes\]](#)

Justification: The abstract and introduction clearly describe the method, challenges addressed, and empirical results (Section 1).

Guidelines:

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

2. Limitations

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

Answer: [\[Yes\]](#)

Justification: Section 5 describes inference latency and limitations in real-time deployment.

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [\[NA\]](#)

Justification: This work is empirical and does not include formal theoretical results or proofs.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [\[Yes\]](#)

Justification: All settings, datasets, and prompts are detailed in Section 4 and Appendix B, D.

Guidelines:

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

5. Open access to data and code

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

Answer: [Yes]

Justification: Public datasets are used, and anonymized code is provided in the supplementary material.

Guidelines:

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

6. Experimental setting/details

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

Answer: [Yes]

Justification: Data splits, prompts, LLM versions, and ablation setups are described in Section 4 and Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

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

Answer: [Yes]

Justification: The paper reports precision, recall, F1 scores, and includes ablation studies with clear performance breakdowns (Tables 1–3).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

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

Answer: [\[Yes\]](#)

Justification: Inference times and token lengths are reported in Table 3; latency limitations are discussed in Section 5.1.

Guidelines:

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

9. Code of ethics

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

Answer: [\[Yes\]](#)

Justification: The research complies with NeurIPS ethical guidelines and uses public data/-models.

Guidelines:

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

10. Broader impacts

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

Answer: [\[Yes\]](#)

Justification: Section 5.2 discusses potential positive applications and concerns related to misuse or latency.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

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

Answer: [NA]

Justification: The paper does not release any high-risk models or datasets.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: All external models and datasets are cited with proper attribution and licensing (e.g., AnomLLM, GPT-4o).

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [NA]

Justification: No new datasets or models are introduced.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

Answer: [NA]

Justification: The research does not involve any human subjects or crowd workers.

Guidelines:

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

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

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

Answer: [NA]

Justification: Not applicable as the study does not involve human participants.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer:[[Yes](#)]

Justification:The paper’s core method involves prompting and evaluating LLMs (Section 2–4), and usage is clearly described.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.