SAPIENZA
UNIVERSITÀ DI ROMA

UC San Diego

Department of Computer Science (Sapienza)
Department of Computer Science and Engineering (UC San Diego)

CaTS-Bench: Benchmarking Vision-Language Models for Time Series Captioning and Reasoning

MSc Thesis | **Luca Zhou** | A.Y. 2024-25

Supervised by Prof. **Fabio Galasso** and Prof. **Rose Yu**

July 2025

This thesis is based on the paper
"CaTS-Bench: Can Language Models Describe Numeric Time Series?"

Co-authored with

Pratham Yashwante, *Marshall Fisher*, *Zihao Zhou*, and Dr. Rose Yu
– **University of California San Diego**

Alessio Sampieri and *Dr. Fabio Galasso*
– **Sapienza University of Rome**

Chapter 1

Acknowledgements

I would like to express my deepest gratitude to the people who supported and guided me throughout this journey.

First and foremost, I thank my family for their unwavering support, both emotionally and financially. Their constant encouragement and stability allowed me to fully focus on my studies and personal growth.

I am immensely grateful to Professor **Rose Yu** at UC San Diego for giving me the opportunity to spend three enriching months as a visiting scholar in her lab 1.2 at the Department of Computer Science and Engineering 1.1. Working closely with her and the talented team—**Pratham Yashwante**, **Marshall Fisher**, and **Zihao Zhou**—was an incredibly rewarding experience. The academic environment at the University of California San Diego was inspiring, and beyond the productive research, I truly enjoyed my time there both professionally and personally.

I would also like to thank Professor **Fabio Galasso**, my co-advisor at Sapienza University of Rome, for supporting my collaboration with UCSD and providing valuable guidance throughout this project. A special thanks goes to **Alessio Sampieri**, a research scientist at ItalAI, who consistently mentored me with dedication and insight during our weekly meetings.

All of the individuals mentioned above are co-authors of the research paper that forms the basis of this thesis. This work would not have been possible without the contribution, support, and mentorship of all these individuals.

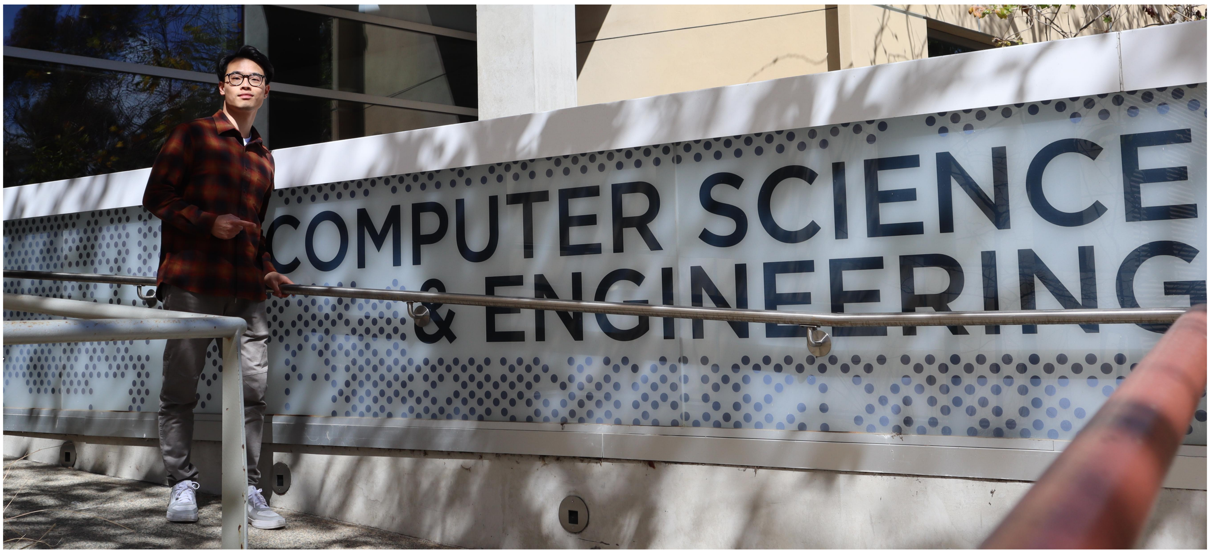


Figure 1.1. Me at CSE (UCSD), 3rd March 2025

Figure 1.2. Rose STL Lab (UCSD), 23rd May 2025

Contents

Chapter 2

Abstract

Time series data is ubiquitous across scientific, industrial, and economic domains, yet its interpretation often remains a domain-specific and expert-driven task. In recent years, the emergence of large language models (LLMs) has sparked a growing interest in their ability to reason over structured and unstructured data. This thesis investigates the role of language models in enhancing the interpretability and analysis of time series, with a particular focus on the novel task of time series captioning—automatically generating natural language descriptions of temporal patterns.

While early attempts in this domain have laid foundational work, they typically rely on synthetic datasets or overlook the rich contextual and visual cues accompanying real-world time series. To address this gap, we introduce CaTS-Bench, a large-scale, multimodal benchmark for Context-aware Time Series captioning and understanding. CaTS-Bench is constructed from 11 real-world datasets spanning various domains, reformulated as both captioning and question-answering (Q&A) tasks. Each instance in the benchmark comprises a numeric time series segment, structured metadata, a line-plot image, and a reference caption generated by an oracle LLM. Furthermore, a curated set of 460 multiple-choice questions probes deeper levels of reasoning, challenging models to understand trends, outliers, causality, and temporal dynamics.

Alongside the dataset, this thesis proposes novel evaluation metrics tailored to temporal reasoning and linguistic fidelity. We conduct a thorough empirical study of state-of-the-art vision-language models (VLMs) and LLMs, highlighting their capabilities and limitations in interpreting temporal data when given multimodal inputs. The results uncover a performance gap between general-purpose foundation models and the requirements of domain-specific time series understanding, especially in tasks demanding precise numerical reasoning and context-aware interpretation.

Beyond captioning, the thesis surveys the landscape of LLM applications in time series analysis, including anomaly detection and forecasting. We analyze how language-driven approaches are being adapted for tasks traditionally dominated by statistical and deep learning models, and discuss their potential for increasing model transparency, enabling zero-shot generalization, and integrating domain knowledge through natural language.

By bridging time series analysis with modern language-based AI, this work aims to lay the groundwork for more explainable, generalizable, and human-aligned models for temporal data understanding.

Chapter 3

Introduction

The ability to effectively interpret time series data is foundational to decision-making across diverse fields, including finance, healthcare, climate science, and industrial automation. Despite its importance, translating raw numerical sequences into concise, human-readable summaries remains a complex and time-consuming task. This process often requires a combination of domain expertise, statistical literacy, and careful visualization. Automating this translation through the task of *time series captioning* (TSC) holds the potential to accelerate insight extraction and make temporal analytics more accessible. In particular, TSC can empower non-expert users to pose natural language queries and receive meaningful interpretations without the need for coding skills or detailed chart inspection.

Recent advances in large language models (LLMs) and multimodal vision-language models (VLMs) have demonstrated impressive capabilities in text generation and visual reasoning. However, when applied to time series data, these models reveal significant limitations. LLMs, for instance, struggle with precise numerical extrapolation, temporal continuity, and the quantification of uncertainty [Tang et al., 2025, Merrill et al., 2024, Tan et al., 2024, Cao and Wang, 2024]. While VLMs have shown encouraging results in visual pattern recognition tasks—such as identifying trends and anomalies from line plots [Zhou and Yu, 2025]—their ability to perform fine-grained numerical reasoning on time series remains largely untested. These limitations are further exacerbated by the lack of comprehensive evaluation resources that reflect the complexity and variability of real-world temporal signals. As a result, model development is often unguided by the nuanced demands of practical applications.

To address these challenges, the research community has proposed time series captioning as a natural interface for foundation models to exhibit both generative and reasoning capabilities [Trabelsi et al., 2025, Jhamtani and Berg-Kirkpatrick, 2021]. However, existing benchmarks in this space are limited in scope: they are frequently based on synthetic data, simplified trend labels, and lack the integration of visual or contextual metadata. Consequently, it is difficult to assess the progress of model architectures, pretraining strategies, or fine-tuning approaches on tasks that truly mirror deployment scenarios. This slows the development and adoption of robust models in high-stakes settings where accurate temporal interpretation is essential.

In response to these gaps, I introduce **CaTS-Bench**, a large-scale, multimodal benchmark specifically designed for *context-aware* time series captioning and reasoning. In this context, "context-aware" refers to the integration of both metadata (e.g., units, domain labels, time and region information) and visual features (e.g., line plots) that provide semantic and numerical grounding to the captioning task. CaTS-Bench is constructed by mining 11 real-world datasets spanning domains such as finance, environmental monitoring, and public health. It contains approximately 570,000 time series segments, each paired with: (1) rich metadata capturing domain-specific cues and unit information [Dong et al., 2024, Wang et al., 2024]; (2) a line plot visualization of the time series, enabling the application of VLMs [Chen et al., 2024a, Zhou and Yu, 2025]; and (3) a high-quality reference caption generated by an oracle LLM.

To evaluate reasoning beyond generation, CaTS-Bench also includes a suite of 460 carefully constructed multiple-choice questions. These questions cover tasks such as time series matching, caption grounding, visual plot interpretation, and comparative reasoning. They are specifically designed to uncover model weaknesses in numerical precision, contextual comprehension, and multimodal alignment.

Furthermore, I propose new evaluation metrics that extend beyond surface-level textual similarity. These metrics prioritize numerical accuracy, trend fidelity, and the effective incorporation of metadata. Through a series of experiments involving both zero-shot and fine-tuned settings on leading VLMs, I show that while current models can generate fluent and plausible captions, they often fall short in capturing quantitative detail and contextual nuance. Notably, models tend to underutilize the visual context available during captioning. This analysis highlights key areas for improvement—such as integrating structured metadata embeddings, enhancing multimodal alignment strategies, and incorporating dedicated numeric reasoning modules—thereby paving the way for more capable foundation models in temporal data interpretation.

We summarize the contributions as follows:

1. **CaTS-Bench**: A comprehensive, multimodal benchmark for context-aware time series captioning and reasoning, incorporating time series segments, rich metadata, visual plots, and grounded reference captions.
2. **Diagnostic Q&A Suite**: A set of four multiple-choice tasks designed to assess capabilities in series matching, caption interpretation, visual reasoning, and comparative analysis.
3. **Comprehensive Evaluation**: A thorough empirical study of state-of-the-art VLMs in both zero-shot and fine-tuned settings, revealing their strengths, limitations, and directions for future development in time series understanding.

Chapter 4

Related Work

Table 4.1. Comparison of TSC benchmarks. CaTS-Bench is the first featuring multimodality, rich metadata, and additional Q&A tasks.

Dataset	# Timestamps	Time Series	Modality	Source Datasets	Metadata	Captions	Captioning	Q&A
TADACap [Fons et al., 2024]	N/A	Partially Synthetic	Visual	4	Minimal	Patterns Only	✓	✗
TRUCE [Jhamtani and Berg-Kirkpatrick, 2021]	$34k$	Partially Synthetic	Numeric	2	✗	Patterns Only	✓	✗
TACO [Dohi et al., 2025]	$2.46b$	Mostly Synthetic	Numeric	8	✗	Expressive	✓	✗
CaTS-Bench	$570k$	Real	Numeric + Visual	11	Rich	Expressive	✓	✓

4.1 Language Models for Time Series Analysis

Recent progress in large language models (LLMs) has prompted a surge of interest in adapting these models for time series tasks [Zhang et al., 2024, Liu et al., 2024a], with initial work primarily concentrating on forecasting. A variety of methods have been explored, including prompt engineering [Liu et al., 2024a, Chatzigeorgakidis et al., 2024], modality alignment [Liu et al., 2024b, Sun et al., Liu et al., 2024c, Pan et al., 2024], data discretization [Ansari et al., 2024, Jin et al., 2024], and specialized finetuning strategies [Zhou et al., 2023, Chang et al., 2023]. These studies highlight the potential of pretrained LLMs to reason over temporal data using natural language interfaces. Despite these promising results, persistent challenges remain: LLMs often underperform on tasks that require precise numerical reasoning, long-range temporal dependency tracking, or structured logic [Tang et al., 2025, Merrill et al., 2024, Tan et al., 2024, Cao and Wang, 2024, Zeng et al., 2023].

4.2 Time Series Captioning with Language Models

To better exploit the strengths of LLMs, recent research has shifted focus toward Time Series Captioning (TSC)—a task that emphasizes narrative generation over strict prediction. Several approaches have emerged. TSLM [Trabelsi et al., 2025] introduces a cross-modal encoder-decoder architecture trained on synthetic data, using retrieval-based denoising to enhance textual quality. TADACap [Fons et al., 2024] leverages time series images and retrieval-based mechanisms to produce domain-aware captions, offering the flexibility to adapt across domains without model retraining. TRUCE [Jhamtani and Berg-Kirkpatrick, 2021] proposes a truth-conditional generation framework that relies on symbolic programs to ensure factual alignment with temporal patterns.

However, existing TSC datasets are limited in multiple aspects. RealCovid and RealKnee [Fons et al., 2024], while grounded in real-world data with human annotations, are narrowly scoped and domain-specific. TRUCE remains focused on synthetic or stock market signals with only basic trend labels. Larger-scale efforts like TACO [Dohi et al., 2025] adopt a backward generation approach to build sizeable caption corpora, but their reliance on synthetic templates without metadata or visual grounding restricts the richness and variability of generated text.

4.3 Multimodal Datasets and Benchmarks

Metadata—such as temporal context, category tags, and measurement units—as well as visual cues from line plots, are often ignored in existing time series resources. Traditional datasets like UCR [Chen et al., 2015], UEA [Bagnall et al., 2018], and Monash [Godahewa et al., 2021] have long supported classification and forecasting tasks, but they are not designed for generative modeling or multimodal reasoning. Even more recent benchmarks like PISA [Xue and Salim, 2023], which emphasize prompt-based forecasting, do not incorporate auxiliary information like metadata or plot imagery.

Meanwhile, studies have begun to demonstrate the value of integrating auxiliary modalities. Research shows that incorporating metadata and visualizations can significantly improve both model interpretability and task performance in generative and predictive settings [Zhou and Yu, 2025, Dong et al., 2024, Chen et al., 2024a, Wang et al., 2024, Kim et al., 2024, Williams et al., 2024, Liu et al., 2025, Tang et al., 2023]. However, no existing benchmark unifies real-world numeric series, high-resolution plots, semantic metadata, and reference captions in a single resource that supports both LLMs and vision-language models (VLMs).

To address these limitations, CaTS-Bench introduces a large-scale, multimodal benchmark tailored for context-aware time series captioning and understanding. Each instance in the dataset contains a time series snippet, a visual line plot, structured metadata, and a detailed caption generated via an oracle LLM. This composition encourages multimodal reasoning, contextual grounding, and semantic interpretation—bringing together diverse signals from domains such as finance, public health, and environmental science. We compare our dataset against existing time series captioning datasets in table 4.1.

4.4 Evaluation of Time Series Captioning

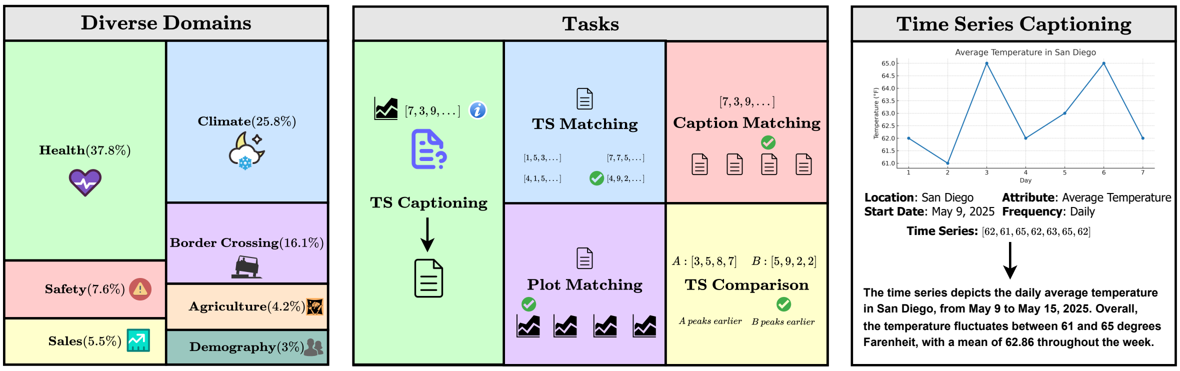
Evaluating time series captions requires metrics that go beyond surface-level text similarity. While classical metrics like BLEU [Papineni et al., 2002], ROUGE [Chin-Yew, 2004], and BERTScore [Zhang* et al., 2020] provide a useful baseline, they often fail to capture numerical correctness, trend fidelity, or alignment with contextual metadata [Zhang et al., 2023, Dohi et al., 2025]. To fill this gap, emerging metrics penalize deviations in reported values or reward coverage of key turning points. However, these approaches lack consistency across studies, making comparative analysis difficult and limiting reproducibility [Dohi et al., 2025].

CaTS-Bench provides a unified framework for evaluating both captioning and diagnostic question-answering (Q&A) tasks. It introduces specialized metrics and task formats designed to assess numerical precision, multimodal grounding, and context utilization. By combining structured evaluation protocols with diverse tasks—including time series matching, caption validation, visual alignment, and comparative reasoning—CaTS-Bench enables fine-grained error analysis and fosters more robust research on temporal language understanding.

Chapter 5

CaTS-Bench Design

This section details the complete data curation pipeline and the design of benchmark tasks in **CaTS-Bench**. The overall pipeline is visualized in Figure 5.2. While the generated examples can be directly used for Time Series Captioning (TSC) evaluation, we further extend the benchmark with a suite of multiple-choice Q&A tasks constructed from the same data. This augmentation allows for a more fine-grained assessment of models' time series reasoning capabilities. Figure 5.1 provides an overview of CaTS-Bench and the task of Time Series Captioning.



Location: San Diego **Attribute**: Average Temperature
Start Date: May 9, 2025 **Frequency**: Daily
Time Series: $[62, 61, 65, 62, 63, 65, 62]$

The time series depicts the daily average temperature in San Diego, from May 9 to May 15, 2025. Overall, the temperature fluctuates between 61 and 65 degrees Farenheit, with a mean of 62.86 throughout the week.

Figure 5.1. Overview of CaTS-Bench. It features diverse domains, provides training and benchmark data, and formulates five challenging tasks, with time series captioning as the primary one.

5.1 Data Curation

CaTS-Bench is curated from 11 diverse real-world source datasets spanning domains: climate [Jha, 2023, Ritchie, 2021], safety [of Los Angeles, n.d., of Public Health, n.d.], USA border crossing [U.S. Department of Transportation, n.d.], demography [Aziz, 1985], health [European Centre for Disease Prevention and Control, 2024, Food and Agriculture Organization of the United Nations, 2024], sales [Hassan, 2020, Chen, 2015], and agriculture [USDA Economic Research Service, 2024]. Below we report full details about each of them.

1. **Air Quality** – Hourly air pollution data from 453 Indian cities (2010–2023), covering 30+ parameters including $PM_{2.5}$, NO_x, CO, and SO_2, compiled from CPCB Jha [2023].

2. **Border Crossing** – Monthly inbound border crossing counts at U.S.-Mexico and U.S.-Canada ports, disaggregated by transport mode and collected by U.S. Customs and Border Protection

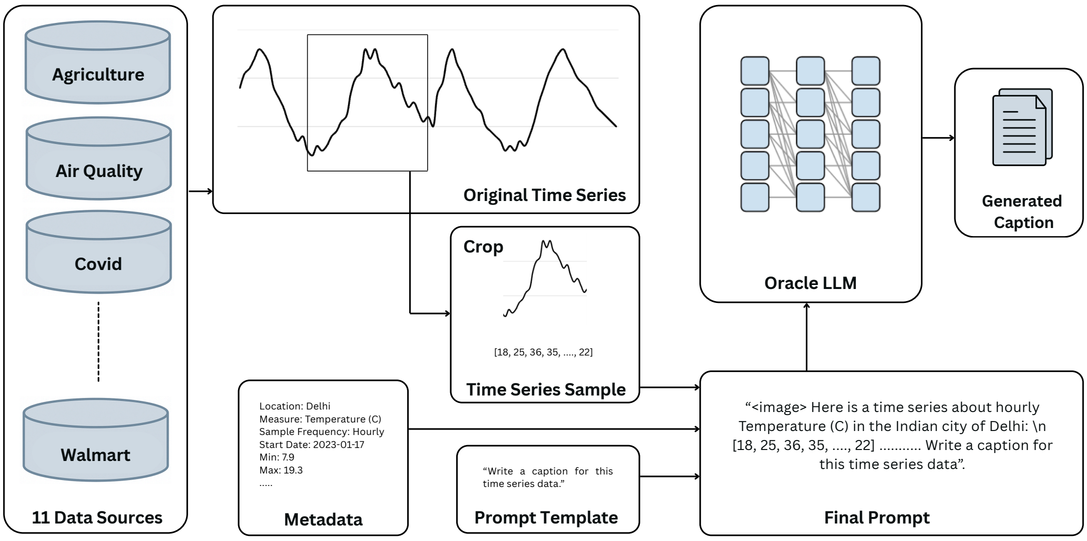


Figure 5.2. CaTS-Bench data generation pipeline. From each source dataset, a random time series window is extracted, paired with metadata and a plot, and used to generate a caption via an oracle LLM.

U.S. Department of Transportation [n.d.].

3. **Crime** – Incident-level crime reports in Los Angeles from 2020 onward, provided by LAPD OpenData and updated biweekly, including NIBRS-compliant records of Los Angeles [n.d.].

4. **Demography** – Annual global indicators from the UN and World Bank (2000–2021) covering population growth, fertility, life expectancy, death rates, and median age to assess patterns of demographic change and collapse Aziz [1985].

5. **Injury** – Annual counts of fatal and severe road traffic injuries in California (2002–2010), disaggregated by transport mode and geography, from CDPH's Healthy Communities Indicators of Public Health [n.d.].

6. **COVID** – Global daily COVID-19 case and death counts (2020), compiled by ECDC, covering over 200 countries with population-adjusted metrics European Centre for Disease Prevention and Control [2024].

7. **CO_2** – National-level per capita CO_2 emissions and GDP trends from Our World in Data, adjusted for trade (consumption-based), spanning 1990–2023 Ritchie [2021].

8. **Calories (Diet)** – Food supply and caloric intake patterns from FAO Food Balance Sheets Food and Agriculture Organization of the United Nations [2024].

9. **Walmart** – Weekly sales data from 45 Walmart stores (2010–2012), enriched with features like temperature, fuel price, CPI, unemployment rate, and holiday flags Hassan [2020].

10. **Retail** – Transactional records from a UK-based online gift retailer (2010–2011), capturing item-level purchases, cancellations, and customer behavior Chen [2015].

11. **Agriculture** – Annual agricultural total factor productivity (TFP) indices from USDA for 1961–2022, covering outputs and inputs like land, labor, capital, and materials across countries USDA Economic Research Service [2024].

From each dataset, we extract full-length time series associated with entities (e.g., countries, cities, or products). We then sample variable-length windows using a random cropping strategy. The number and size of windows are dataset-specific, governed by the total time steps available, to ensure balanced representation across domains. Each window is paired with the following:

1. A **metadata JSON** file containing contextual attributes (e.g., domain, location, start time).
2. A **line plot image** rendered with randomized styles (color, width, figure size).
3. A **ground-truth caption** generated via `Gemini 2.0 Flash`, prompted with (i) the raw numeric values of the window and (ii) metadata enriched with basic statistics (mean, std, min, max). An example of the prompt is shown in 5.1.

This format allows for rigorous evaluation of models' ability to synthesize multimodal cues—numerical, textual, and visual—into an expressive narrative that captures trends, anomalies, and context.

Ground-Truth Caption Generation Prompt

The following is an example of a prompt for generating the ground-truth caption from the source dataset *Crime*.

```
Here is a time series about the number of <sampling frequency> crimes in <town>, Los
    Angeles, from <start date> to <end date>:

<time series>

The all-time statistics of <town> until today are:
Mean: <general mean of this town>
Standard Deviation: <general standard deviation of this town>
Minimum: <general minimum of this town>
Maximum: <general maximum of this town>

And the statistics for this specific time series are:
Mean: <mean of this specific series>
Standard Deviation: <standard deviation of this specific series>
Minimum: <minimum of this specific series>
Maximum: <maximum of this specific series>

Describe this time series by focusing on trends and patterns. Discuss concrete numbers
     you see and pay attention to the dates.

For numerical values, ensure consistency with the provided time series. If making
    percentage comparisons, round to the nearest whole number. Report the dates when
    things happened.

Use the statistics I provided you for comparing this example to the normalcy.
Do not add any extra information beyond what is given.
Highlight significant spikes, dips, or patterns.

You don't have to explicitly report the numeric values of general statistics; you just
     use them for reference.
Compare the trends in this time series to global or regional norms, explaining whether
     they are higher, lower, or follow expected seasonal patterns.
```

```
When making comparisons, clearly state whether differences are minor, moderate, or
    significant.

Use descriptive language to create engaging, natural-sounding text.
Avoid repetitive phrasing and overused expressions.

Answer in a single paragraph of four sentences at most, without bullet points or any
    formatting.
```

We emphasize that while the ground-truth captions are LLM-generated, these captions are anchored in the true underlying data as we explicitly provide full contextual metadata to the oracle that is not available during evaluation and instruct it not to include any external facts beyond what is given. Thus, the task challenges models' ability to reason from the multimodal time series input, not merely to mimic the oracle. We also stress that both the time series window size and the line plot style are sampled with randomness during data generation. This design choice discourages models from overfitting to specific hyperparameters and better reflects practical conditions, where end users provide time series of varying lengths and visualizations in diverse styles. We provide a few concrete samples in the following.

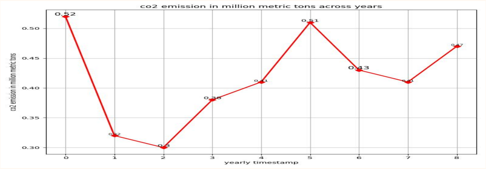
Time Series Segment

`[0.52, 0.32, 0.30, 0.38, 0.41, 0.51, 0.43, 0.41, 0.47]`

Metadata JSON

{ "attribute": "co2_emissions", "country": "Djibouti", "end year of this series": 2018, "maximum of this specific series": 0.52, "mean of this specific series": 0.42, "minimum of this specific series": 0.3, "population at the end year": 1071886.0, "population at the start year": 930251.0, "region": "Middle East & North Africa", "sampling frequency": "yearly", "standard deviation of this specific series": 0.07, "start year of this series": 2010 }

Line Plot Image



Ground-Truth Caption

Djibouti's CO2 emissions from 2010 to 2018, show fluctuations around the average of 0.42 million metric tons, with a standard deviation of 0.07. Starting at 0.52 million metric tons in 2010, emissions generally decreased to a low of 0.3 million metric tons by 2012, before experiencing some increases and decreases, ending the period at 0.47 million metric tons in 2018.

Figure 5.3. Sample 1 showing time series data, metadata, plot image, and reference caption.

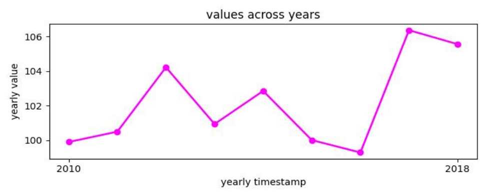
Time Series Segment

```
[99.89, 100.49, 104.22, 100.93, 102.85,
100.0, 99.28, 106.37, 105.56]
```

Metadata JSON

{ "attribute": "Agricultural output index (2015=100)", "country": "Namibia", "end year of this series": 2018, "historical max": 152.47, "historical mean": 102.05, "historical min": 69.97, "maximum of this specific series": 106.37, "mean of this specific series": 102.18, "metrics info": "The Agricultural output index (2015=100) comprises the following components: crop_output, animal_output, fish_output.", "minimum of this specific series": 99.28, "sampling frequency": "yearly", "start year of this series": 2010 }

Line Plot Image



Ground-Truth Caption

The Agricultural output index for Serbia and Montenegro shows considerable fluctuation between 2014 and 2018. Starting at 103.58 in 2014, the index dipped to 100.0 in 2015, then peaked at 113.67 in 2016, before falling to a minimum of 96.07 in 2017, and then rising to 113.25 in 2018. This volatility, with a range of 17.6 units, suggests a moderate level of instability compared to the historical mean, and the absence of a clear upward or downward trend indicates that the agricultural output did not follow expected seasonal patterns.

Figure 5.4. Sample 2 showing time series data, metadata, plot image, and reference caption.

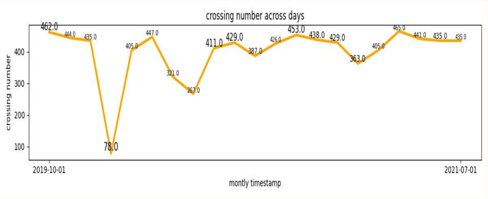
Time Series Segment

```
[462, 444, 435, 78, 405, 447, 321, 267,
411, 429, 387, 426, 453, 438, 429, 363,
405, 465, 441, 435, 435]
```

Metadata JSON

{ "border": "US-Canada Border", "end date": "2021-07-01", "general maximum in the history of this port": 1710, "general mean in the history of this port": 737.34, "general minimum in the history of this port": 0, "general standard deviation in the history of this port": 269.0, "maximum in this specific series": 465, "mean of this specific series": 398.86, "means": "Train Passengers", "minimum in this specific series": 78, "port": "Detroit", "sample frequency": "monthly", "standard deviation of this specific series": 85.49, "start date": "2019-10-01", "state": "Michigan" }

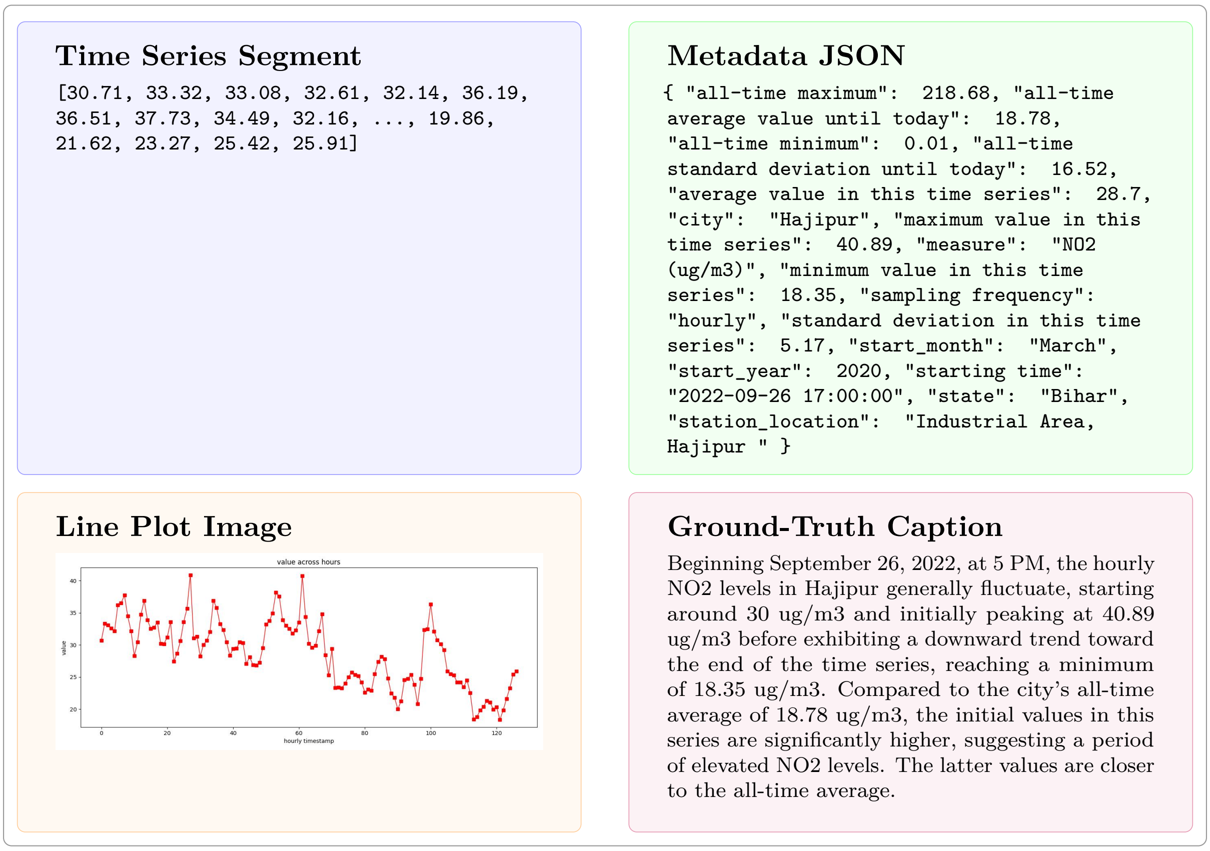
Line Plot Image



Ground-Truth Caption

The monthly train passenger volume crossing the Detroit-Windsor border, starting October 2019, displays a noticeable dip in April 2020 with only 78 passengers, a significant deviation from the series mean of 399. The passenger volume fluctuates between 321 and 465 for the remainder of the period, showing no clear upward or downward trend. Compared to the all-time mean of 737.34, this time series exhibits significantly lower passenger numbers, suggesting a notable change in border crossing patterns.

Figure 5.5. Sample 3 showing time series data, metadata, plot image, and reference caption.



Time Series Segment

```
[30.71, 33.32, 33.08, 32.61, 32.14, 36.19,
36.51, 37.73, 34.49, 32.16, ..., 19.86,
21.62, 23.27, 25.42, 25.91]
```

Metadata JSON

{ "all-time maximum": 218.68, "all-time average value until today": 18.78, "all-time minimum": 0.01, "all-time standard deviation until today": 16.52, "average value in this time series": 28.7, "city": "Hajipur", "maximum value in this time series": 40.89, "measure": "NO2 (ug/m3)", "minimum value in this time series": 18.35, "sampling frequency": "hourly", "standard deviation in this time series": 5.17, "start_month": "March", "start_year": 2020, "starting time": "2022-09-26 17:00:00", "state": "Bihar", "station_location": "Industrial Area, Hajipur " }

Line Plot Image

Ground-Truth Caption

Beginning September 26, 2022, at 5 PM, the hourly NO2 levels in Hajipur generally fluctuate, starting around 30 ug/m3 and initially peaking at 40.89 ug/m3 before exhibiting a downward trend toward the end of the time series, reaching a minimum of 18.35 ug/m3. Compared to the city's all-time average of 18.78 ug/m3, the initial values in this series are significantly higher, suggesting a period of elevated NO2 levels. The latter values are closer to the all-time average.

Figure 5.6. Sample 4 showing time series data, metadata, plot image, and reference caption.

Furthermore, to prevent information leakage, we partition each source dataset temporally before generating the samples. Specifically, the first 80% is used for generating training samples, whereas the last 20% is reserved exclusively for generating test samples. Random window cropping is applied separately to the training and test partitions. This strategy ensures that the model is evaluated on future, unseen data relative to the training set. The actual benchmark samples consist of the test split resulting from this process. We leave the training split of the data for optional training. Our final dataset contains $20k$ examples, split into roughly $16k$ training samples and $4k$ test samples. Detailed statistics across the different source datasets are reported in Table 5.1.

Table 5.1. Dataset statistics by domains. AQ: Air Quality, Border: Border Crossing, Demo: Demography, Injury: Road Injuries, Calories: Calories Consumption, Agri: Agriculture

Metric	**All**	**AQ**	**Border**	**Crime**	**Demo**	**Injury**	**COVID**	**CO$_2$**	**Calories**	**Walmart**	**Retail**	**Agri**
# Source Time Steps	287M	286M	397k	38k	14k	37k	720k	34k	234k	6k	7k	49k
# Samples Generated	20k	4.4k	3.2k	764	598	756	5.5k	732	2.1k	544	551	835
# Train Samples	16k	3.5k	2.6k	611	478	604	4.4k	585	1.7k	435	440	668
Avg. Train Sample Length	29.1	65.3	21.2	76.8	11.6	5.9	75.8	9.5	12.2	12.2	22.4	7.3
# Test Samples	4k	886	646	153	120	152	1.1k	147	422	109	111	167
Avg. Test Sample Length	26.1	66.0	21.2	76.9	5.0	3.6	73.0	8.7	5.5	11.8	8.1	7.5

5.2 Time Series Captioning (TSC)

In the TSC task, models must generate a coherent and informative caption describing the provided time series. Each evaluation instance comprises four components:

1. **Numeric Series**: Raw time-indexed values embedded as text (e.g., `[25.3, 26.1, 26.8, ...]`).
2. **Contextual Metadata**: Describes key attributes such as units, source, sampling interval, and domain tags (e.g., "Hourly temperature readings from Rome, May 2000"). Unlike the ground-truth prompt, evaluation metadata excludes computed statistics.
3. **Visual Input**: A line plot image of the time series, aiding VLMs in grounding their textual outputs in visual structure.
4. **Instruction Template**: A standardized directive prompting the model to generate a caption. An example of the prompt is provided in 5.2.1.

5.2.1 Baseline Caption Generation Prompt

```
Here is a time series about the number of <sampling frequency> crimes in <town>, Los
    Angeles, from <start date> to <end date>:

<time series>

Describe this time series by focusing on trends and patterns. Discuss concrete numbers
     you see and pay attention to the dates. For numerical values, ensure consistency
    with the provided time series. If making percentage comparisons, round to the
    nearest whole number. Report the dates when things happened.

Compare the trends in this time series to global or regional norms, explaining whether
     they are higher, lower, or follow expected seasonal patterns.

When making comparisons, clearly state whether differences are minor, moderate, or
    significant.

Use descriptive language to create engaging, natural-sounding text. Avoid repetitive
    phrasing and overused expressions.

Answer in a single paragraph of four sentences at most, without bullet points or any
    formatting.
```

5.3 Q&A Multiple-Choice Tasks

To evaluate models more comprehensively, we introduce a suite of Q&A tasks formulated as multiple-choice questions. These tasks investigate different reasoning skills related to time series understanding. All tasks are automatically constructed from the same source data used for captioning. We design four question types, as shown in Table 5.2. All questions are generated using task-specific, fixed templates.

5.3.1 Time Series Matching

Given a caption, select the correct time series from a pool of candidates. Distractors are generated by applying challenging perturbations—random shuffling, temporal reversal, and noise

Table 5.2. Q&A task breakdown

Q&A Task	**# Questions**
Time Series Matching	100
Caption Matching	100
Plot Matching	100
Time Series Comparison:	
Amplitude Comparison	40
Reach Maximum Earlier	40
Mean Comparison	40
Variance Comparison	40
Total	**460**

injection—forcing models to reason beyond surface-level trends or numeric overlap. Below, we present an example of a time series matching question 5.7.

Question

Here is a time series caption:
From 2014 to 2019, Bulgaria's Agricultural output index (2015=100) generally increased, starting at 103.4 in 2014 and reaching a peak of 109.23 in 2019, with a slight dip to 100.0 in 2015. The average output index during this period was 105.4, notably lower than the historical mean of 126.73, suggesting a period of relatively lower agricultural productivity compared to Bulgaria's long-term performance. The increase from 2015 to 2019 indicates a moderate recovery and growth phase within this specific timeframe.

What time series is best described by this caption?
(A) [109.23, 107.24, 108.45, 104.1, 100.0, 103.4]
(B) [108.45, 100.0, 104.1, 107.24, 103.4, 109.23]
(C) [103.9, 99.8, 104.1, 109.2, 106.8, 109.23]
(D) [103.4, 100.0, 104.1, 108.45, 107.24, 109.23]

You must respond only with valid JSON, and no extra text or markdown.

The JSON schema is:
`{"answer": <string>}`
`<string>` must be an answer string containing only A, B, C, or D.
Ensure your output parses as JSON with exactly one top-level object containing the answer field.

Answer

`"answer": "D"`

Figure 5.7. Example of a *time series matching* question.

5.3.2 Caption Matching

Given a time series (in numeric format), the task is to select the correct descriptive caption from a set of four candidates. The distractor captions are sampled from other series within the same domain and window size range, ensuring that all options are semantically plausible and structurally similar. This prevents models from exploiting superficial keyword mismatches and encourages genuine trend understanding. Next, we show an example of a caption matching question in 5.8 and the prompt used to generate the distractor options through semantic 5.3.2 and numeric perturbation 5.3.2. .

Question

Here is a time series:
37.00,37.00,37.00,37.00,37.00,40.57,40.57,40.57,40.57,40.57,40.57

What caption best describes to this time series?

(A) From May 1st to July 26th, 2024, the daily COVID-19 deaths in China show a fluctuating pattern, with values generally ranging between 0.29 and 2.86. There are periods of relative stability, such as the initial days of May with a consistent 0.86, interspersed with occasional spikes to 2.86, and dips to 0.29 towards the end of July. Compared to the general daily death statistics for China, where the mean is 73.0 and the maximum reaches 6812.0, this specific time series indicates a period of significantly lower daily deaths, suggesting a substantial improvement in the COVID-19 situation during this timeframe.

(B) From October 24, 2023, to November 3, 2023, the daily COVID-19 cases in Luxembourg show a relatively stable pattern, beginning at 37.3 cases and rising to 40.57 cases by October 29, 2023, where it remains for the rest of the period. Compared to the country's general statistics, where the mean is 236, the daily cases during this period are significantly lower, suggesting a period of reduced viral transmission. This trend does not follow any expected seasonal patterns, as COVID-19 case numbers are known to fluctuate unpredictably.

(C) From October 24, 2023, to November 3, 2023, the daily COVID-19 cases in Luxembourg show a relatively stable pattern, beginning at 37 cases and rising to 40.57 cases by October 29, 2023, where it remains for the rest of the period. Compared to the country's general statistics, where the mean is 236.0, the daily cases during this period are significantly lower, suggesting a period of reduced viral transmission. This trend does not follow any expected seasonal patterns, as COVID-19 case numbers are known to fluctuate unpredictably.

(D) From October 24, 2023, to November 3, 2023, the daily COVID-19 cases in Luxembourg show a relatively unstable pattern, beginning at 37 cases and decreasing to 40.57 cases by October 29, 2023, where it remains for the rest of the period. Compared to the country's general statistics, where the mean is 236.0, the daily cases during this period are significantly lower, suggesting a period of reduced viral transmission. This trend does follow expected seasonal patterns, as COVID-19 case numbers are known to fluctuate unpredictably.

You must respond only with valid JSON, and no extra text or markdown.

The JSON schema is:
`{"answer": <string>}`
`<string>` must be an answer string containing only A, B, C, or D.
Ensure your output parses as JSON with exactly one top-level object containing the answer field.

Answer

`"answer": "C"`

Figure 5.8. Example of a *caption matching* question.

Semantic Perturbation Prompt To perturb a caption so that its semantic meaning is altered while keeping numbers intact, we feed the following prompt into `Gemini 2.0 Flash`.

```
Your task is to minimally modify a time series description so that it's meaning is
    altered but the numbers are maintained.

For example, you can switch "increase" with "decrease", "upward" to "downward" or
    something more sophisticated. Keep the description structurally identical to the
    original text, you don't have to alter too much information, altering anywhere
    between 1 to 3 parts is enough. Do not edit the numbers.

Here's the description to modify:
<caption>

Give your answer in a paragraph of text as the given description, without any
    explanation and formatting.
```

Numeric Perturbation Prompt To perturb a caption so that its numbers are altered while its semantic information is preserved, we feed the following prompt into `Gemini 2.0 Flash`.

```
Your task is to slightly modify the numbers in a time series description so that its
    semantics remain the same but the numbers are slightly altered.

For example, you can replace "12" with "12.2", "45%" with "46%". Keep the description
    structurally and semantically identical to the original text; you don't have to
    alter all numbers but anywhere between 1 to 3 times is enough. Make sure that the
    altered number still makes sense and fits the scale of the phenomenon.

Here's the description to modify:

<caption>

Give your answer in a paragraph of text as the given description, without any
    explanation and formatting.
```

5.3.3 Plot Matching

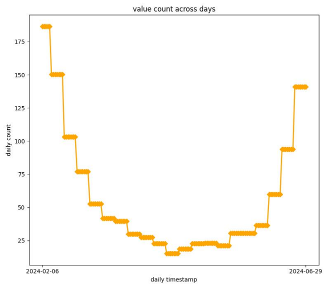
Given a numeric time series, the task is to identify the correct line plot from a pool of four images. Distractor plots correspond to unrelated series with similar length and the same domain. This task evaluates a model's ability to connect numeric input to its visual rendering, testing grounding capabilities across modalities. See a concrete question at 5.9.

Question

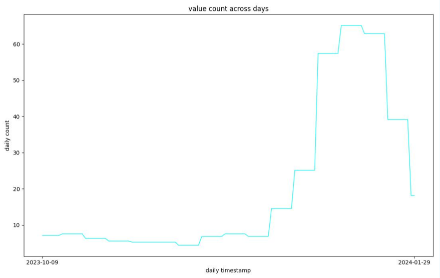
Here is a time series:
186.57, 186.57, 186.57, 186.57, 186.57, 150.29, 150.29, 150.29, 150.29, 150.29, 150.29, 150.29, 103.14, 103.14, 103.14, 103.14, 103.14, 103.14, 103.14, 77.00, 77.00, 77.00, 77.00, 77.00, 77.00, 77.00, 52.71, 52.71, 52.71, 52.71, 52.71, 52.71, 52.71, 41.71, 41.71, 41.71, 41.71, 41.71, 41.71, 41.71, 39.71, 39.71, 39.71, 39.71, 39.71, 39.71, 39.71, 29.86, 29.86, 29.86, 29.86, 29.86, 29.86, 29.86, 27.43, 27.43, 27.43, 27.43, 27.43, 27.43, 27.43, 22.57, 22.57, 22.57, 22.57, 22.57, 22.57, 22.57, 15.14, 15.14, 15.14, 15.14, 15.14, 15.14, 15.14, 18.71, 18.71, 18.71, 18.71, 18.71, 18.71, 18.71, 22.71, 22.71, 22.71, 22.71, 22.71, 22.71, 22.71, 23.14, 23.14, 23.14, 23.14, 23.14, 23.14, 23.14, 21.00, 21.00, 21.00, 21.00, 21.00, 21.00, 21.00, 30.57, 30.57, 30.57, 30.57, 30.57, 30.57, 30.57, 30.57, 30.57, 30.57, 30.57, 30.57, 30.57, 30.57, 36.29, 36.29, 36.29, 36.29, 36.29, 36.29, 36.29, 59.71, 59.71, 59.71, 59.71, 59.71, 59.71, 59.71, 93.71, 93.71, 93.71, 93.71, 93.71, 93.71, 93.71, 140.86, 140.86, 140.86, 140.86, 140.86, 140.86, 140.86

Here are four plots of different time series:

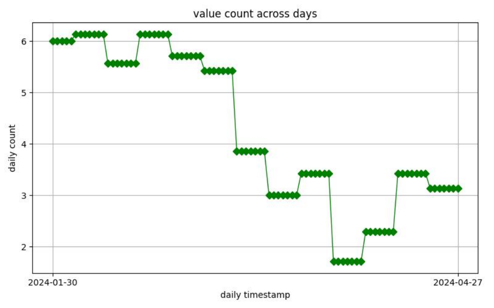
(A)



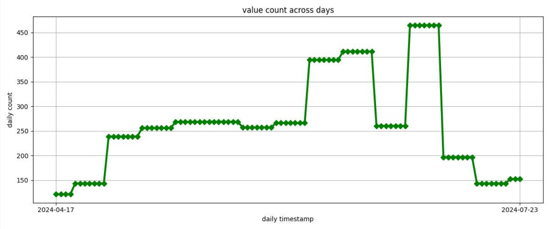
(B)



(C)



(D)



Which plot corresponds to the time series provided above?

You must respond only with valid JSON, and no extra text or markdown.

The JSON schema is:
`{"answer": <string>}`
`<string>` must be an answer string containing only A, B, C, or D.
Ensure your output parses as JSON with exactly one top-level object containing the answer field.

Answer

`"answer": "C"`

Figure 5.9. Example of a *plot matching* question.

5.3.4 Time Series Comparison Tasks

These tasks involve reasoning over pairs of time series, provided in numeric format, and answering comparative questions. We include four subtypes: amplitude comparison, peak time comparison, mean comparison, and variance comparison. We describe them in depth in the following.

Amplitude Comparison

Skill tested: `Which series exhibits a larger overall amplitude range?`
This question type evaluates a model's ability to identify and compare the overall range of values in two given time series, defined as the difference between the maximum and the minimum values. An example is probided below in 5.10.

Question

Given two time series A and B, detect which one has a higher amplitude defined as maximum - minimum.

A: [1.15, 0.92, 0.85, 0.75, 0.57, 0.62, 0.6, 0.5, 0.68, 0.72, 0.8, 0.67, 0.8, 0.55, 0.55, 0.7, 0.88]
B: [87.0, 83.0, 77.0, 74.0, 84.0]

You must respond only with valid JSON, and no extra text or markdown.

The JSON schema is:
`{"answer": <string>}`
`<string>` must be an answer string containing only A, B.
Ensure your output parses as JSON with exactly one top-level object containing the answer field.

Answer

`"answer": "B"`

Figure 5.10. Example of a *time series amplitude comparison* question.

Peak Time Comparison

Skill tested: `Which series reaches its maximum value first?`
This task tests a model's ability to detect and locate the maximum values in two given time series, and then make a comparison of their indices. See an example in 5.11.

Question

Given two time series A and B, detect which one reaches its maximum earlier.

A: [66.76, 83.06, 85.77, 90.77, 98.81, 90.62, 80.05, 91.36, 89.59, 76.4, 80.1, 85.6, 84.41]
B: [949.0, 689.0, 561.0, 552.0, 563.0]

You must respond only with valid JSON, and no extra text or markdown.

The JSON schema is:
`{"answer": <string>}`
`<string>` must be an answer string containing only A, B.
Ensure your output parses as JSON with exactly one top-level object containing the answer field.

Answer

`"answer": "B"`

Figure 5.11. Example of a *time series peak comparison* question.

Mean Comparison

Skill tested: `Which series has a higher mean value?`
This task is particularly challenging as it requires the model to perform fine-grained mathematical reasoning by inferring the mean of two time series and making a meaningful comparison. See 5.12.

Question
Given the following two time series A and B, please identify which one has higher overall values.
A: [65.0, 65.0, 64.0, 37.0, 55.0, 51.0]
B: [6.29, 6.29, 6.29, 7.0, 7.0, 7.0, 7.0, 6.71, 6.71, 6.71, 6.71, 6.717, 7.57, 7.57, 7.14, 7.14, 7.14, 7.14, 7.43]

You must respond only with valid JSON, and no extra text or markdown.

The JSON schema is:
`{"answer": <string>}`
`<string>` must be an answer string containing only A, B.
Ensure your output parses as JSON with exactly one top-level object containing the answer field.

Answer
`"answer": "A"`

Figure 5.12. Example of a *time series mean comparison* question.

Variance Comparison

Skill tested: `Which series has greater variability?`
Likewise, this task also demands an understanding of temporal variability and the ability to reason about fluctuations within each series, making it an even more challenging task compared to mean comparison. An example question is showen in 5.13.

Question
Given the following two time series A and B, please identify which one has higher volatility.
A: [0.14, 0.14, 0.14, 0.29, 0.29, 0.29, 0.29, 0.29, 0.29, 0.29, 0.57, 0.57, 0.57, 0.57, 0.57, 0.57]
B: [0.21, 0.33, 0.41, 0.39, 0.44, 0.35, 0.35, 0.43, 0.51, 0.65, 0.69, 0.74]

You must respond only with valid JSON, and no extra text or markdown.

The JSON schema is:
`{"answer": <string>}`
`<string>` must be an answer string containing only A, B.
Ensure your output parses as JSON with exactly one top-level object containing the answer field.

Answer
`"answer": "A"`

Figure 5.13. Example of a *time series variance comparison* question.

5.3.5 Post-Processing

An initial pool of $4k$ questions per type is created, and then we filtered them to remove easier instances; specifically, those questions answered correctly by the `Qwen 2.5 Omni` model are removed. From the remaining pool of approximately $7k$ questions, a random subset is selected for evaluation, yielding a final set of 460 challenging questions. To avoid biased comparisons, `Qwen`

`2.5 Omni` is excluded from the pool of baselines. During our preliminary inspection, we found several Time Series Matching questions to be ambiguous, having multiple plausible answers; these were manually reviewed and revised to ensure a single correct answer.

Chapter 6

Evaluation Protocol

To comprehensively evaluate model-generated captions against the ground truth, we employ a diverse set of metrics that targets linguistic quality, statistical inference, and numerical fidelity. Each generated caption is evaluated on the metrics we describe next.

6.1 Time Series Captioning

6.1.1 Standard Linguistic Metrics

We evaluate caption similarity using standard reference-based metrics DeBERTa Score [Zhang* et al., 2020] measures token-level semantic similarity using contextual embeddings. BLEU [Papineni et al., 2002] captures exact n-gram overlap (averaged over 1- to 4-grams), while ROUGE-L [Chin-Yew, 2004] focuses on the longest common subsequence to reflect fluency. METEOR [Banerjee and Lavie, 2005] accounts for synonymy, stemming, and paraphrase matching. Lastly, SimCSE [Gao et al., 2021] computes cosine similarity between sentence embeddings produced by a contrastively pretrained RoBERTa [Liu et al., 2019] model to assess deeper semantic alignment beyond lexical overlap.

DeBERTa Score The DeBERTa Score is a contextual similarity metric based on cosine similarity between contextual embeddings of tokens in the candidate (c) and reference (r) captions. Given token embeddings from the DeBERTa encoder, the metric computes token-level precision, recall, and F1:

$$\mathrm{F1}_{\mathrm{DeBERTa}} = \frac{2 \cdot P \cdot R}{P + R}, \quad P = \frac{1}{|c|} \sum_{i \in c} \max_{j \in r} \cos(\mathbf{e}_i, \mathbf{e}_j), \quad R = \frac{1}{|r|} \sum_{j \in r} \max_{i \in c} \cos(\mathbf{e}_j, \mathbf{e}_i) \tag{6.1}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are the contextual embeddings of candidate and reference tokens, respectively.

BLEU BLEU evaluates n-gram overlap between a candidate caption and reference using precision with a brevity penalty to discourage short outputs:

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n \right), \quad \mathrm{BP} = \begin{cases} 1, & \text{if } c > r \\ e^{1-r/c}, & \text{if } c \leq r \end{cases} \tag{6.2}$$

where p_n is the modified precision for n-grams, w_n are weights (usually uniform), c is candidate length, and r is reference length.

ROUGE-L ROUGE-L measures fluency via the length of the longest common subsequence (LCS) between candidate and reference:

$$\text{ROUGE-L}_{\text{F1}} = \frac{(1+\beta^2)\cdot \text{LCS}}{r+c}, \quad \text{LCS} = \text{LongestCommonSubsequence}(r, c) \tag{6.3}$$

where β balances recall and precision (often $\beta = 1$), and r and c are the reference and candidate lengths.

METEOR METEOR aligns unigrams using exact matches, stems, synonyms, and paraphrases. It then computes an F-score and applies a fragmentation penalty:

$$\text{METEOR} = F_{\text{mean}} \cdot (1 - \text{Pen}), \quad F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9P}, \quad \text{Pen} = 0.5 \left(\frac{\text{chunks}}{\text{matches}}\right)^3 \tag{6.4}$$

where P and R are unigram precision and recall, and chunks refers to non-contiguous matched segments.

SimCSE SimCSE computes semantic similarity at the sentence level using cosine similarity between sentence embeddings:

$$\text{SimCSE}(c, r) = \cos(\mathbf{h}_c, \mathbf{h}_r) = \frac{\mathbf{h}_c \cdot \mathbf{h}_r}{||\mathbf{h}_c|| ||\mathbf{h}_r||} \tag{6.5}$$

where $\mathbf{h}_c$ and $\mathbf{h}_r$ are sentence embeddings of the candidate and reference, produced by a contrastively pretrained RoBERTa encoder.

6.1.2 LLM-based Oracle Score

To incorporate qualitative judgment beyond lexical or numeric matching, we include an Oracle Score computed via an LLM (`Gemini 2.0 Flash`). The model is few-shot prompted with examples illustrating how to evaluate captions based on *numeric accuracy* (correctness of reported values and statistics), *coherence* (fluency, grammar, and logical consistency), and *semantic similarity* (alignment with the meaning of the ground truth caption).

Given a generated caption and its ground truth counterpart, the oracle produces a holistic quality score in the range $[0, 100]$, which we normalize to $[0, 1]$. This score reflects a human-like evaluation of caption quality. The prompting strategy used to elicit this score is detailed below.

```
You are an expert evaluator of artificially generated captions against ground truth
    captions.

Your task is to provide scores (0-100) for the generated caption’s quality in
    comparison to the ground truth.

Scoring Criteria:
  1. Semantic Similarity: How closely does the generated caption convey the same
      meaning as the ground truth?
  2. Information Overlap: How much of the factual information present in the ground
      truth is also accurately represented in the generated caption?
  3. Numeric Correctness: Are all numbers in the generated caption exactly the same as
       those in the ground truth? A single numerical mismatch results in a score of 0.
  4. Overall Quality: A holistic score reflecting the overall accuracy and usefulness
      of the generated caption. Assign higher weight to numeric correctness and
      semantic similarity.


Examples:
    Generated: "The chart shows a slight increase in sales."

    Ground Truth: "Sales increased by 5%."

    Scores:
      - Semantic Similarity: 80
      - Information Overlap: 50
      - Numeric Correctness: 0
      - Overall: 40

    Generated: "The average daily temperature of Seattle was 11 degrees Celsius in the
         beginning of March 2023, and it increased to 14 by the end of the month."

    Ground Truth: "The average daily temperature of Seattle was 10 degrees Celsius in
        the beginning of March 2023, and it increased to 14 by the end of the month."

    Scores:
      - Semantic Similarity: 100
      - Information Overlap: 100
      - Numeric Correctness: 50
      - Overall: 70

    Generated: "There are two peaks in this time series."

    Ground Truth: "The time series shows two distinct peaks."

    Scores:
      - Semantic Similarity: 95
      - Information Overlap: 100
      - Numeric Correctness: 100
      - Overall: 98

    Generated: <generated caption>

    Ground Truth: "<ground-truth caption>

      Provide your scores in the following STRICT format:
      - Semantic Similarity: [score]
      - Information Overlap: [score]
      - Numeric Correctness: [score]
      - Overall: [score]

  Do NOT include any additional text or explanations.
```

We leverage `Gemini 2.0 Flash` both as an evaluation baseline and as a scoring oracle. This dual use is justified by prior work, finding no significant bias toward self-generated text when models evaluate text [Huang et al., 2024], and demonstrating that language models can serve as effective zero-shot evaluators of caption quality [Hsu et al., 2023, Maeda et al., 2024].

6.1.3 Numeric Fidelity Metrics

Since TSC involves reporting exact or approximate numerical values, we introduce two tailored metrics to quantify numerical accuracy, both bounded within $[0, 1]$:

1. **Statistical Inference Accuracy**: While models are not explicitly instructed to compute descriptive statistics, they occasionally infer and verbalize metrics such as the mean, standard deviation, minimum, and maximum based on the raw time series and metadata. To evaluate this behavior, we report the percentage of captions in which these statistics are mentioned and fall within a 5% relative error, using offline-computed ground truth values. Importantly, captions are not penalized for omitting statistics—only inaccurately reported values are considered errors. This metric primarily measures hallucination, favoring omission over incorrect numerical claims.

2. **Numeric Score**: For each ground truth caption, we extract all numerical values (excluding time-related ones like year or month) and search for the closest numerical value in the generated caption. A match is recorded if the closest value is within a 5% relative tolerance. We compute *Recall* (fraction of ground truth numbers matched), *Accuracy* (mean of $1 - \min\{\text{relative\_error}, \text{tolerance}\}$) over all matched numbers), and a *Final Score* as a weighted combination: $\lambda_A \cdot \text{Accuracy} + \lambda_R \cdot \text{Recall}$, with $\lambda_A = 0.3$ and $\lambda_R = 0.7$ to emphasize recall. While the previous metric targets numerical hallucinations, this one focuses on penalizing captions that fail to include adequate numerical detail.

6.2 Q&A Tasks

For Q&A tasks, we adopt **accuracy** as the evaluation metric, as each question is designed to have a single correct answer.

Chapter 7

Experiments

We evaluated a variety of vision-language models (VLMs)—including both open-source and proprietary systems—on CaTS-Bench across two tasks: free-form time series captioning and multiple-choice question answering. A subset of the open-source models was also finetuned on our training split, with finetuning and hardware configurations detailed next. To ensure a fair comparison, all models were prompted using a consistent, template-based format, without any task- or model-specific prompt tuning. The complete list of models, along with a description of the human baseline constructed from volunteer student responses, is provided in the following sections.

7.1 Hardware Resources

All experiments were conducted on a high-performance computing node featuring two *AMD EPYC 7453* processors, providing a total of 56 logical CPUs, and 125 GB of RAM (with over 117 GB available during runtime). For GPU acceleration, the system includes eight *NVIDIA A100* GPUs—six PCIe 80 GB models and two PCIe 40 GB models—alongside an ASPEED graphics controller used for display purposes. This configuration offers ample computational and memory resources suitable for mid- to large-scale deep learning training and inference. The models we finetune range in size from 2 billion to 11 billion parameters, with finetuning times spanning from a few hours to a day.

7.2 Finetuning Setup

For finetuning, we adopt a unified training strategy guided by best practices in instruction tuning for multimodal inputs. All models are trained using the AdamW optimizer with a cosine learning rate scheduler and a base learning rate of 2×10^{-5}. We apply gradient accumulation to simulate a larger batch size. Mixed precision training and gradient checkpointing are enabled for memory efficiency. Low Rank Adaptation (LoRA) is used to adapt large models by tuning a small subset of parameters, while keeping the rest of the model frozen or partially frozen. To ensure deterministic and focused generation, we use a temperature of 0.3 during inference across all evaluated models. Each model is fine-tuned using a structured JSONL dataset comprising time series plot images and corresponding image-grounded chat-style conversations. We preprocess data with each model's native processor and apply minimal resizing to maintain fidelity in the visual input. Special care is taken to exclude padding and <image> tokens from loss computation by assigning them an ignore index. See Table 7.1 for an overview of hyperparameters and choices.

Table 7.1. Finetuning configurations

Hyperparameter	**Value**
Batch size	4
Gradient Accumulation	12
Epochs	3
Learning Rate	2×10^{-5}
Scheduler	Cosine
Optimizer	AdamW
Precision	`bf16`
LoRA rank	8 or 16
Dropout	0.05
Image resolution	224–560

7.3 Baselines

7.3.1 Models

We evaluate `Gemini 2.0 Flash` and `Gemini 2.5 Pro Preview` [Team et al., 2023], `Claude 3 Haiku` and `Claude 3.7 Sonnet` [Anthropic, 2024], `GPT-4o` [Achiam et al., 2023], `InternVL 2.5 (8b & 38b)` [Chen et al., 2024b], `LLaVA v1.6 Mistral 7b` (default) and `34b` [Liu et al., 2023], `Phi-4 Multimodal Instruct 5.6b` [Abdin et al., 2024], `Idefics 2 (8b)` [Laurençon et al., 2024], `SmolVLM (2b)` [Marafioti et al., 2025], `QwenVL (7b)` [Bai et al., 2023], `Llama 3.2 Vision (11b)` [Grattafiori et al., 2024], and `Gemma 3 (12b & 27b)` [Team et al., 2025] for both TSC and Q&A tasks.

Program-Aided Language Model Since TSC requires accurate numerical calculations in addition to text generation, it is a suitable context for program-aided language (PAL) models Gao et al. [2023]. We evaluate the performance of `QwenVL 32b` in a PAL context: for each time series, the model is instructed to write a Python program that produces the entire time series caption, both text and numbers, as its output. The model's response is then evaluated as code in a Python environment, with the program's return value taken as the final caption. The vast majority (approx. 90%) of model-generated Python programs are successful on the first attempt; in the cases where the generated program errors, we increase the permitted output token count and randomly re-generate the model's response until a valid program is returned. Below we show the prompt given to the PAL baseline.

```
<caption_prompt>

### Instructions for the assistant
1. You are an expert coding assistant; think through the task **step-by-step**.
2. Write **Python 3.12** code (inside one ‘‘‘python‘‘‘ block) that computes the final
    answer.
   * Use only the Python Standard Library (e.g. you may use the ‘math‘, ‘statistics‘
      libraries).
   * Wrap everything in a ‘solve()‘ function that will be invoked to produce the final
       caption.
   * The code **must produce the caption string itself**. Any numerical values can be
      computed
     in Python and formatted into the caption string. Make sure to use any values you
        compute
     in the resulting caption string.
3. The ‘solve()‘ function you write will be invoked to produce the final caption.


### Output format (exactly; no extra text, explanations, or formatting)
‘‘‘python
# code that defines solve() and any desired strings
solve()
‘‘‘
```

The full TSC prompt from 5.2.1 replaces the <caption_prompt> placeholder.

7.3.2 Human

To establish a human performance baseline, we invited university students to voluntarily complete all four Q&A tasks. These tasks span a range of reasoning types, including fine-grained statistical comparisons, semantic interpretation, and multimodal alignment. Participants were recruited through academic networks and completed the tasks without the aid of external tools, ensuring a fair comparison with models operating under similar conditions. Participation was entirely voluntary, with no compensation, and individuals could withdraw at any time. Below we present the instructions given to the volunteers for their participation.

```
Participant Information and Consent Form for Time Series QA Questionnaire

Thank you for considering participation in our study!

This questionnaire is part of a research project evaluating human performance
    on time series understanding tasks. Your responses will help us establish a
     baseline for comparing human performance to that of current language
    models. You will be given a Google Form consisting of 10 to 14 multiple-
    choice questions of the same type, and you should not use any external
    tools.

Please read the following information carefully before continuing:

Voluntary Participation: Your participation is entirely voluntary. You may
    choose not to participate or to withdraw at any time without any
    consequences.

Duration: The questionnaire is brief and is estimated to take between 3 and 6
    minutes to complete.

Anonymity & Data Use: No personal information will be collected or stored.
    Your answers will remain anonymous and will be used solely for research
    purposes, such as evaluating and reporting model performance in academic
    publications.
```

```
No Compensation: There is no monetary or material compensation for
   participating in this study.

Confidentiality: All collected data will be handled securely. Only aggregated
   and anonymized results will be published.

By proceeding, you confirm that you understand the above terms and agree to
   participate in this research study.

Thank you for your collaboration and contribution to our research!

Date: __________________ Signature: __________________
```

7.4 Main Results

7.5 Time Series Captioning Results

To ensure a fair comparison across the 11 source datasets, we report macro-averaged scores for each metric. This approach mitigates sample size imbalances, since some domains contain significantly more data—and prevents any single domain from disproportionately influencing the results. Results are shown in Table 7.2.

Our experiments reveal that finetuning yields substantial performance gains across the majority of evaluated metrics. For proprietary models, `GPT-4o` and *Gemini* models generally outperform `Claude` models, although `Claude 3.7 Sonnet` excelled in statistical inference. Finetuned models demonstrate strong overall performance in both textual and statistical inference assessments, but the performance is still moderate in the numeric score. Notably, finetuned `Idefics 2` dominates, achieving the highest scores in critical metrics like BERT F1 (0.759), SimCSE (0.908), ROUGE-L (0.452), mean inference (0.885), and numeric score (0.733). Similarly, finetuned `LLaVA v1.6 Mistral` significantly outperforms most proprietary and pretrained models in several text-based evaluations. In contrast, pretrained models generally lag behind, exhibiting more dispersed performance without clear overall leaders. This underscores the effectiveness of finetuning to improve both the linguistic quality and the numerical precision of generated captions, suggesting a crucial step for developing robust captioning models.

7.5.1 Q&A Tasks

Figure 7.1 offers a visual summary of model performance on our Q&A tasks, while Table 7.3 provides the full results and discussion. Model performance is highly variable, with even proprietary models occasionally failing to exceed random chance on specific tasks. No model consistently outperforms others across all categories.

Models perform better on time series comparison tasks with binary choices, where the reduced option space likely simplifies reasoning. We also observe a striking asymmetry between time series matching and caption matching: identifying the correct time series for a caption is significantly harder than the other way around. The most challenging task overall is plot matching, which requires true cross-modal grounding. This difficulty underscores a core weakness in current VLMs: the limited capacity to associate numerical patterns with corresponding visual features. Proprietary models such as GPT-4o and `Gemini 2.0 Flash` outperform others across several

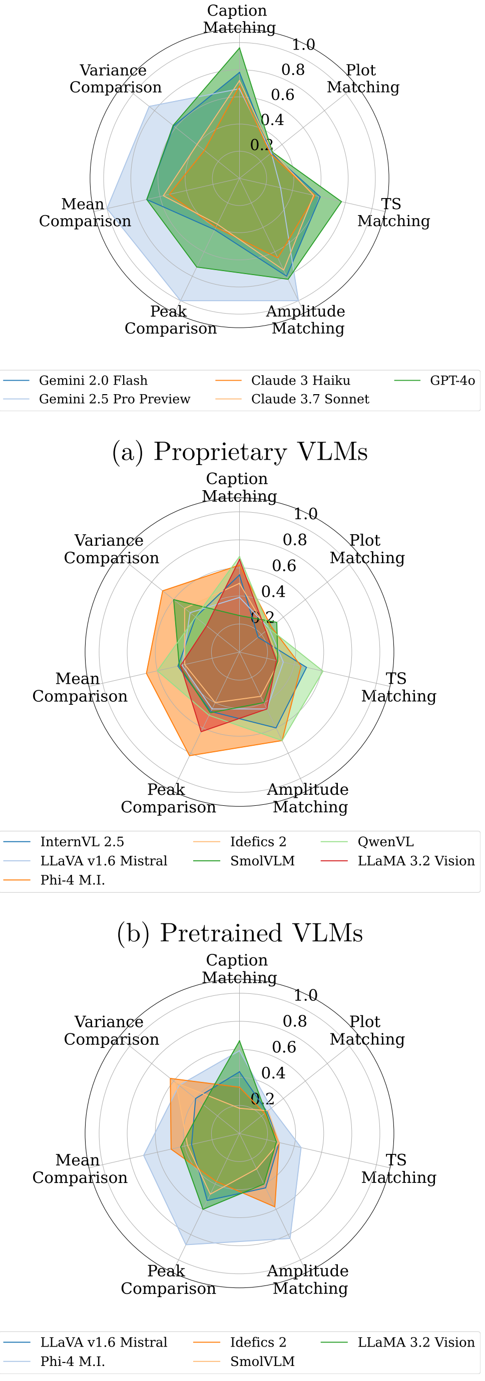
Table 7.2. Evaluation of generated captions. Numeric: numeric score. Mean/STD/Max/Min refer to statistical inference accuracy. **Bolded** and underlined scores denote first and second places.

Model	DeBERTa F1	SimCSE	BLEU	ROUGE-L	METEOR	Oracle	Mean	STD	Max	Min	Numeric
Proprietary											
Gemini 2.0 Flash	0.688	0.858	0.137	0.318	0.279	**0.724**	0.651	0.916	0.985	0.917	0.677
Gemini 2.5 Pro Preview	0.668	0.845	0.088	0.267	0.284	0.689	0.494	0.667	**0.994**	**0.971**	0.714
Claude 3 Haiku	0.682	0.856	0.112	0.291	0.300	0.671	0.693	0.735	0.977	0.898	0.623
Claude 3.7 Sonnet	0.661	0.845	0.083	0.257	0.272	0.663	0.707	**0.925**	0.985	0.923	0.613
GPT-4o	0.681	0.865	0.112	0.284	0.296	0.705	0.700	0.778	0.990	0.921	0.644
Pretrained											
InternVL 2.5 (8b)	0.659	0.794	0.081	0.247	0.260	0.601	0.610	0.920	0.949	0.794	0.589
InternVL 2.5 (38b)	0.688	0.868	0.129	0.305	0.331	0.683	0.784	0.640	0.966	0.887	0.685
LLaVA v1.6 Mistral	0.650	0.820	0.086	0.259	0.287	0.551	0.644	0.611	0.864	0.743	0.517
LLaVA v1.6 34b	0.655	0.825	0.094	0.265	0.285	0.526	0.445	0.550	0.843	0.698	0.560
Phi-4 M.I.	0.624	0.797	0.074	0.274	0.239	0.543	0.457	0.443	0.942	0.859	0.583
Idefics 2	0.604	0.698	0.040	0.226	0.162	0.507	0.616	0.368	0.903	0.806	0.455
SmolVLM	0.594	0.693	0.044	0.224	0.178	0.474	0.747	0.446	0.864	0.705	0.474
QwenVL	0.643	0.890	0.082	0.249	0.261	0.494	0.565	0.257	0.822	0.657	0.504
QwenVL PAL	0.685	0.843	0.108	0.292	0.282	0.674	**0.903**	0.549	0.980	0.942	0.613
Llama 3.2 Vision	0.671	0.850	0.118	0.290	0.315	0.598	0.594	0.666	0.952	0.877	0.685
Gemma 3 12b	0.676	0.867	0.097	0.279	0.317	0.654	0.653	0.578	0.957	0879.	0.673
Gemma 3 27b	0.667	0.863	0.085	0.263	0.309	0.661	0.694	0.900	0.968	0.864	0.668
Finetuned											
InternVL 2.5 (8b)	0.655	0.809	0.088	0.259	0.282	0.568	0.597	0.464	0.904	0.779	0.594
LLaVA v1.6 Mistral	0.758	0.907	0.285	0.445	**0.441**	0.524	0.828	0.294	0.976	0.926	0.732
Phi-4 M.I.	0.662	0.821	0.010	0.285	0.279	0.605	0.645	0.641	0.965	0.877	0.607
Idefics 2	**0.759**	**0.908**	**0.290**	**0.452**	0.437	0.537	0.885	0.379	0.985	0.927	**0.733**
SmolVLM	0.613	0.781	0.091	0.269	0.265	0.536	0.590	0.297	0.898	0.777	0.643
QwenVL	0.643	0.790	0.082	0.249	0.260	0.494	0.565	0.257	0.822	0.657	0.504
Llama 3.2 Vision	0.667	0.844	0.111	0.283	0.310	0.592	0.502	0.619	0.955	0.867	0.668

metrics, with `GPT-4o` achieving the highest scores in caption, plot, and TS matching. Among pretrained open-source models, `Phi-4 M.I.` shows strong performance, particularly in time series and statistical reasoning (e.g., amplitude and mean comparison).

An analysis of the highlighted statistics reveals a striking contrast between the finetuned and pretrained models. The finetuned model frequently produces highly confident yet incorrect predictions, whereas the pretrained model demonstrates more caution, acknowledging that the mean is lower than expected without attempting to estimate a specific value. This comparison indicates that finetuning on the CaTS-Bench training set does not consistently enhance the model's capacity for accurate statistical inference and may, in some cases, promote overconfidence. Notably, certain proprietary models are now reaching, and at times even surpassing, human performance on specific subsets of tasks. While this signals exciting progress in the field, it also highlights the nuances of human cognitive performance, particularly under conditions where distraction might occur. It is vital to note, however, that no singular model has consistently achieved near-human proficiency across the entirety of the benchmark's demands. The plot retrieval task, in particular, stands out as a significant hurdle, robustly affirming the unparalleled human capacity for holistic visual-numerical integration, a critical frontier for time series understanding.

Finetuning on the captioning task yields mixed results: while some models (e.g., `Phi-4 M.I.`, `Idefics 2`) show notable gains in specific sub-tasks, others exhibit consistent performance drops. Notably, finetuning often fails to improve Q&A accuracy and may even degrade it, likely due



(a) Proprietary VLMs

(b) Pretrained VLMs

(c) Finetuned VLMs

Figure 7.1. Model accuracy across Q&A sub-tasks. Proprietary models perform best, pretrained models lag behind, and finetuned models struggle across all tasks.

to task misalignment and catastrophic forgetting, as caption generation and multiple-choice reasoning require related but distinct skills. These outcomes highlight the risk of overfitting and reduced generalization, particularly when training data lacks linguistic diversity.

Table 7.3. Model accuracy for time-series Q&A tasks. **Bolded** and underlined scores respectively denote first and second places (excluding human performance). Caption/Plot/TS refer to caption, plot, and time series matching. Amplitude/Peak Earlier/Mean/Variance refer to amplitude, peak, mean, and variance comparison. Green and Red indicate improvement and degradation after finetuning, respectively.

Model	Caption	Plot	TS	Amplitude	Peak Earlier	Mean	Variance
Proprietary							
Gemini 2.0 Flash	0.78	0.30	0.61	0.8	0.42	0.7	0.62
Gemini 2.5 Pro Preview	0.66	0.30	0.31	**1.0**	**1.0**	**1.0**	**0.85**
Claude 3 Haiku	0.68	0.29	0.57	0.65	0.40	0.53	0.33
Claude 3.7 Sonnet	0.72	0.31	0.56	0.75	0.375	0.575	0.4
GPT-4o	**0.96**	0.31	**0.77**	0.825	0.725	0.7	0.625
Pretrained							
InternVL 2.5	0.55	0.17	0.49	0.60	0.47	0.45	0.40
LLaVA v1.6 Mistral	0.39	0.27	0.32	0.45	0.45	0.42	0.45
Phi-4 M.I.	0.62	0.29	0.45	0.7	0.82	0.68	0.7
Idefics 2	0.49	0.25	0.29	0.35	0.4	0.4	0.5
SmolVLM	0.26	**0.34**	0.28	0.4	0.48	0.44	0.6
QwenVL	0.68	0.27	0.61	0.7	0.5	0.6	0.4
Llama 3.2 Vision	0.66	0.24	0.27	0.45	0.63	0.43	0.3
Finetuned							
LLaVA v1.6 Mistral	0.44	0.25	0.29	0.43	0.53	0.35	0.4
Phi-4 M.I.	0.59	0.29	0.45	0.83	0.88	0.7	0.55
Idefics 2	0.33	0.23	0.29	0.58	0.38	0.5	0.63
SmolVLM	0.18	0.26	0.29	0.28	0.48	0.38	0.58
QwenVL	0.55	0.25	0.43	0.7	0.4	0.58	0.58
Llama 3.2 Vision	0.66	0.24	0.27	0.4	0.6	0.43	0.33
Human	0.81	0.95	0.83	0.925	0.85	0.95	0.90

Chapter 8

Ablations

8.1 Role of the Visual Modality

We conduct two additional experiments to investigate how VLMs handle the visual modality for TSC.

Visual Modality Ablation

We perform a modality removal experiment by stripping away the time series plot and providing only the associated textual metadata and the numerical values of the time series. This quantifies the contribution of the visual channel and enables a better understanding of the model's captioning performance. We evaluate a selected subset of pretrained baselines to assess their intrinsic reliance on vision. We first provide a heatmap of performance differences with and without the visual input, and the full results are then reported in the subsequent Table 8.1.

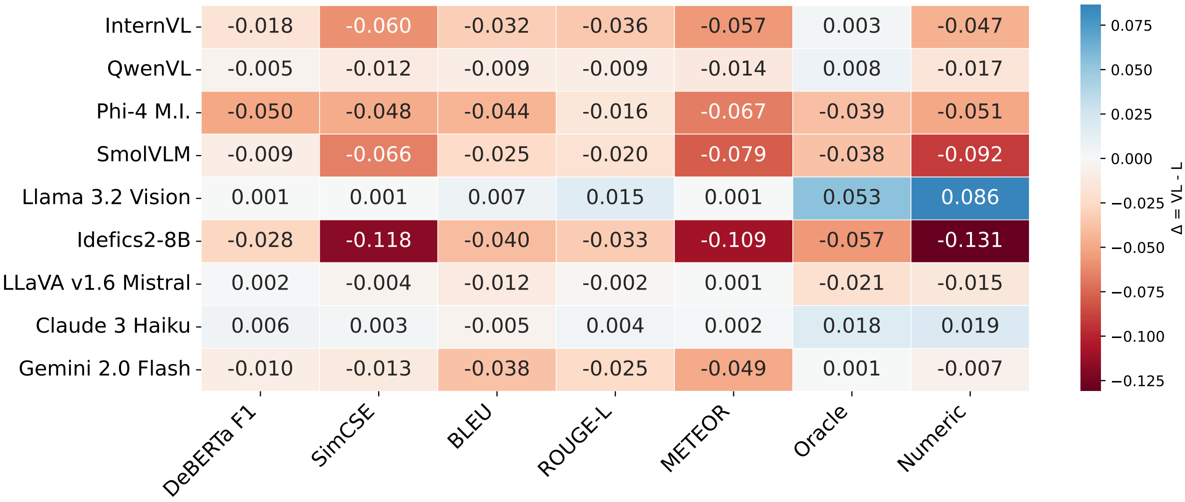


Figure 8.1. Heatmap for performance differences between VL (vision-language input) and L (text-only input) inputs across various metrics and models. Each cell shows the score $\Delta = \text{VL} - \text{L}$ with positive (blue) indicating improvement from visual input and negative (red) indicating degradation.

Table 8.1. Evaluation of generated captions under modality ablation. Each metric is split into two columns: **VL** (vision-language input) and **L** (text-only input).

Model	**DeBERTa F1**		**SimCSE**		**BLEU**		**ROUGE-L**		**METEOR**		**Oracle**		**Numeric**	
	VL	L	VL	L	VL	L	VL	L	VL	L	VL	L	VL	L
InternVL	0.659	0.677	0.794	0.854	0.081	0.113	0.247	0.283	0.260	0.317	0.601	0.598	0.589	0.636
QwenVL	0.643	0.648	0.790	0.802	0.081	0.090	0.249	0.258	0.260	0.274	0.493	0.485	0.503	0.520
Phi-4 M.I.	0.624	0.674	0.797	0.845	0.074	0.118	0.274	0.290	0.239	0.306	0.543	0.582	0.583	0.634
SmolVLM	0.594	0.603	0.692	0.758	0.043	0.068	0.224	0.244	0.178	0.257	0.473	0.511	0.473	0.565
Llama 3.2 Vision	0.670	0.669	0.850	0.849	0.117	0.110	0.290	0.275	0.314	0.313	0.597	0.544	0.684	0.598
Idefics2-8B	0.604	0.632	0.698	0.816	0.040	0.080	0.225	0.258	0.161	0.270	0.507	0.564	0.454	0.585
LLaVA v1.6 Mistral	0.650	0.648	0.820	0.824	0.086	0.098	0.259	0.261	0.287	0.286	0.551	0.572	0.517	0.532
Claude 3 Haiku	0.682	0.676	0.856	0.853	0.112	0.117	0.291	0.287	0.300	0.298	0.671	0.653	0.628	0.609
Gemini 2.0 Flash	0.688	0.698	0.858	0.871	0.137	0.175	0.318	0.343	0.279	0.328	0.724	0.723	0.677	0.684

Our experiments suggest that the additional contribution of the visual modality to caption quality is modest for most models. As shown in Figure 8.1, most models show only marginal performance drops—or even slight gains—when the time series plot is removed, suggesting a strong dependence on textual priors over visual understanding. In particular, models such as `Idefics2`, `Phi-4 M.I.`, and `QwenVL` perform better in text-only settings in semantic and lexical metrics, indicating that generation is largely driven by language pretraining or instruction tuning rather than true visual interpretation. Proprietary models such as `Gemini 2.0 Flash` and `Claude 3 Haiku` maintain strong performance with visual input, but the performance gap (Δ) remains modest, underscoring the underuse of plot-based information. Interestingly, numeric and oracle scores tend to decline when visual input is removed, hinting at weak but present reliance on plot structure for numeric reasoning. These results point to a subtle yet important misalignment: models are exposed to visual data but often fail to meaningfully attend to or reason with it.

Visual Attention Analysis

We display the attention maps to localize the focus of VLMs on the plot while generating captions. This qualitative analysis clarifies the reliance on visual cues versus textual priors.

Interpreting visual grounding in large multimodal models is non-trivial. as not all of them expose interpretable cross-modal attention mechanisms. We attempt this using the LLaVA model, which provides access to decoder-level cross-attention weights over vision tokens. We adapt the approach in Zhang for the `LLaVA 1.6` model.

We visualize per-token visual grounding via the following steps. For each generated token, we extract the decoder cross-attention matrix $\mathbf{A}_{\text{llm}} \in \mathbb{R}^{T \times V}$, where T is the number of generated tokens and V is the number of vision tokens.

Next, we zero out the attention to the beginning-of-sequence token and normalize each row:

$$\tilde{\mathbf{A}}_{\text{llm}}[t, v] = \begin{cases} 0, & \text{if } v = \texttt{<bos>} \\ \frac{\mathbf{A}_{\text{llm}}[t,v]}{\sum_{v'} \mathbf{A}_{\text{llm}}[t,v']}, & \text{otherwise} \end{cases} \tag{8.1}$$

From the CLIP style vision encoder, we extract attention matrices $\mathbf{A}_{\text{vit}}^{(l)} \in \mathbb{R}^{V \times V}$ from multiple

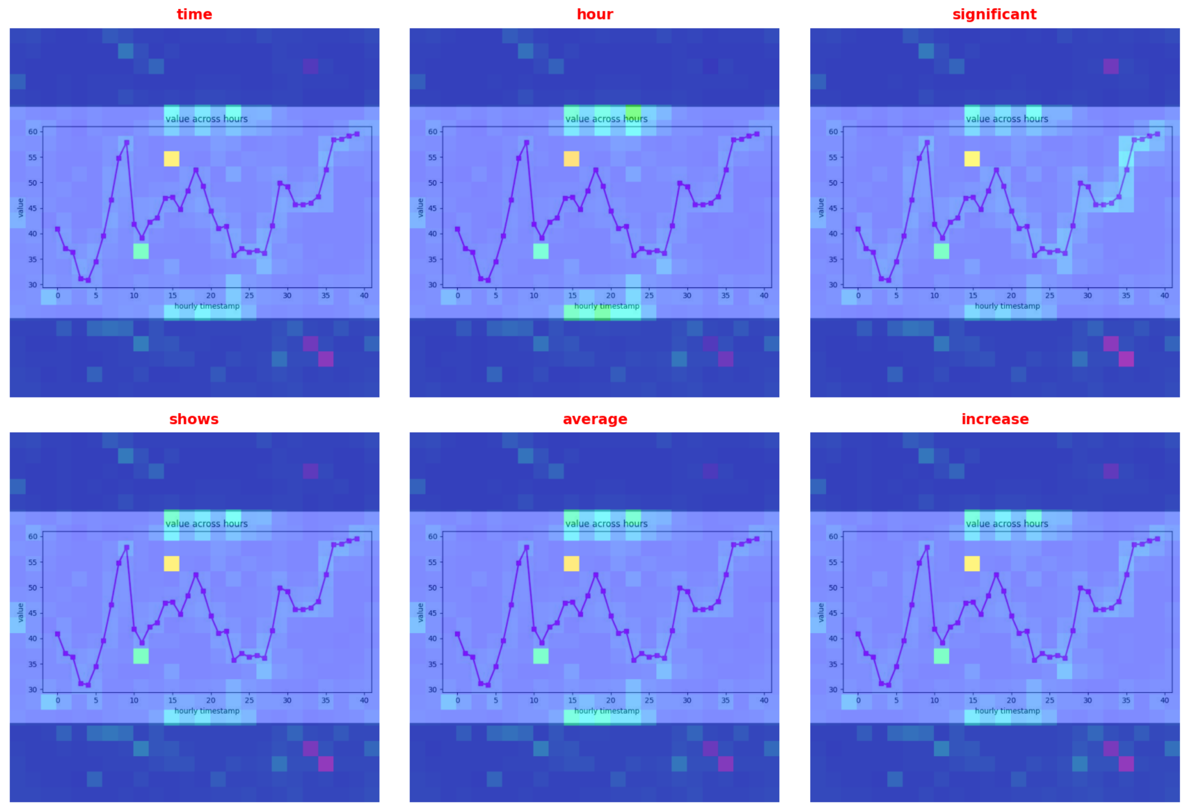


Figure 8.2. Word-level attention maps for the top six tokens from `LLaVA v1.6 Mistral` on a time series plot.

layers and average them:

$$\bar{\mathbf{A}}_{\text{vit}} = \frac{1}{L}\sum_{l=1}^{L}\mathbf{A}_{\text{vit}}^{(l)} \tag{8.2}$$

For each token t we compute its attention-weighted vision token distribution and project it back to the 2D image grid:

$$\hat{\mathbf{H}}_t = \texttt{Upsample}\left(\text{reshape}(\mathbf{H}_t, \text{grid})\right) \tag{8.3}$$

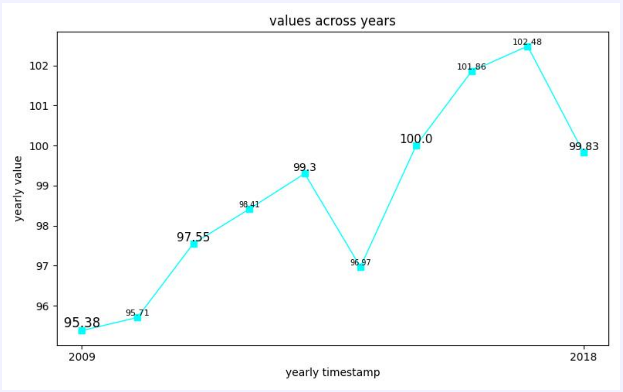
The projected map $\hat{\mathbf{H}}_t$ is rendered as a heatmap and overlaid on the original image. This allows inspection of which visual regions contribute to each generated token.

Figure 8.2 shows `LLaVA v1.6 Mistral` exhibiting minimal visual grounding in time series captioning. Attention is largely uniform across vision patches, indicating learned parameters mostly disregard visual cues. While some tokens show localized attention to the trend on the line plot, these are rare and inconsistent. These results provide weak evidence of localized visual grounding in TSC.

8.2 Additional Attention Analysis

In this section, we perform additional ablations to visualize how the model reasons and attends over the visual plot for the task of time series captioning.

We compare the generated captions with and without explicitly providing the numeric time series data as input to the model in the prompt. As seen in Figure 8.3, when a numeric time series is included, the model is able to attend to values rendered on the plot and reference them in the caption. However, it still produces several factual and interpretative errors. Notably, it describes the trend as "increased steadily," despite the clear dip in 2014 and a decline post-2017. It also incorrectly identifies 2018 as the year of the peak value 102.48, while the actual peak occurs in 2017. Similarly, the slight dip is misattributed to 2016 instead of the correct year 2014. In contrast, when the numeric series is removed from the input, the generated caption becomes significantly more erroneous. The model fabricates plausible-sounding but incorrect values, for example, claiming the index reached 90.5 in 2009 and spiked to 105 in 2014, neither of which is present in the actual plot. This suggests that the absence of explicit numeric context forces the model to hallucinate plausible trajectories based solely on the shape of the line plot. While both versions demonstrate limitations in temporal precision, the numeric-aware caption is more grounded and less prone to hallucinating specific values.



Generated Caption (With Numeric Time Series)

The time series plot shows the yearly aggregated input index (2015=100) in El Salvador from 2009 to 2018. The index comprises the components land, labor, capital, and materials. The plot shows a general upward trend, with a slight dip in 2016. The index has increased steadily over the years, reaching a peak of 102.48 in 2018. This indicates that the country's economy has been growing, and the input index has been increasing, which could be a sign of economic development and growth.

Generated Caption (Without Numeric Time Series)

The time series plot shows the yearly aggregated input index (2015=100) in El Salvador from 2009 to 2018. The index comprises the components land, labor, capital, and materials. The plot displays a general upward trend, with a noticeable increase in the index from 2009 to 2018. In 2009, the index was at 90.5, and it gradually increased to reach 100 in 2010. From 2010 to 2018, the index continued to rise, with a notable spike in 2014, when it reached 105. The index then settled back to around 100 in the following years.

Figure 8.3. Comparison of generated captions with and without numeric time series input. The numeric-aware model still produces factual errors but performs better than the numeric-agnostic version, which fabricates values entirely.

We also observe, as visualized in Figure 8.4 and Figure 8.5, that the model exhibits diffuse and non-discriminative attention across most output tokens in both cases. However, for a subset of tokens that correspond directly to textual or numeric content visually rendered in the plot, the attention becomes notably more focused and spatially localized. Many tokens receive sharp attention centered along the x-axis or near visible tick marks, indicating that the model is leveraging superficial visual-textual alignment for anchoring references. This behavior suggests that the model's visual grounding is heavily biased toward regions with explicitly rendered text rather than structurally meaningful visual patterns. While this can help reinforce alignment in some contexts, it also contributes to errors: hallucinated values receive attention despite not being present in the plot, and abstract inferences are made without strong visual evidence. This reveals a key insight: the model's visual attention is driven more by what looks prominent on the plot (like tick marks or text) than by what is semantically important (like the actual data trend). This becomes especially problematic when numeric inputs are removed, as the model relies on visual cues that may not reflect the true meaning of the data.

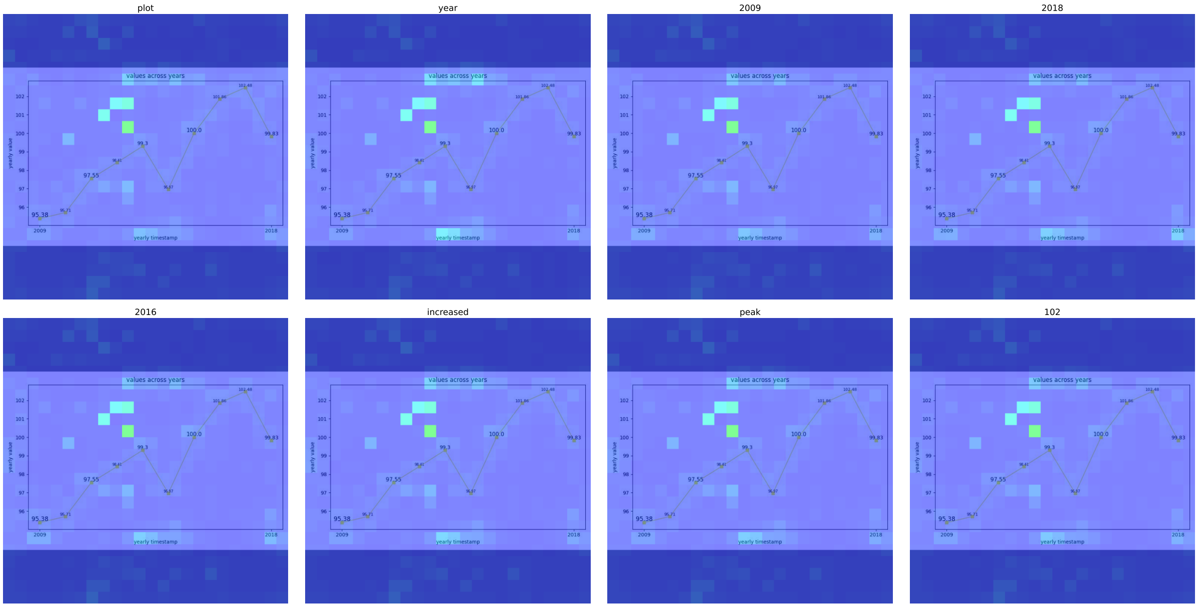


Figure 8.4. Word-level attention maps from `LLaVA v1.6 Mistral` on a time series plot with numeric time series present in the prompt.

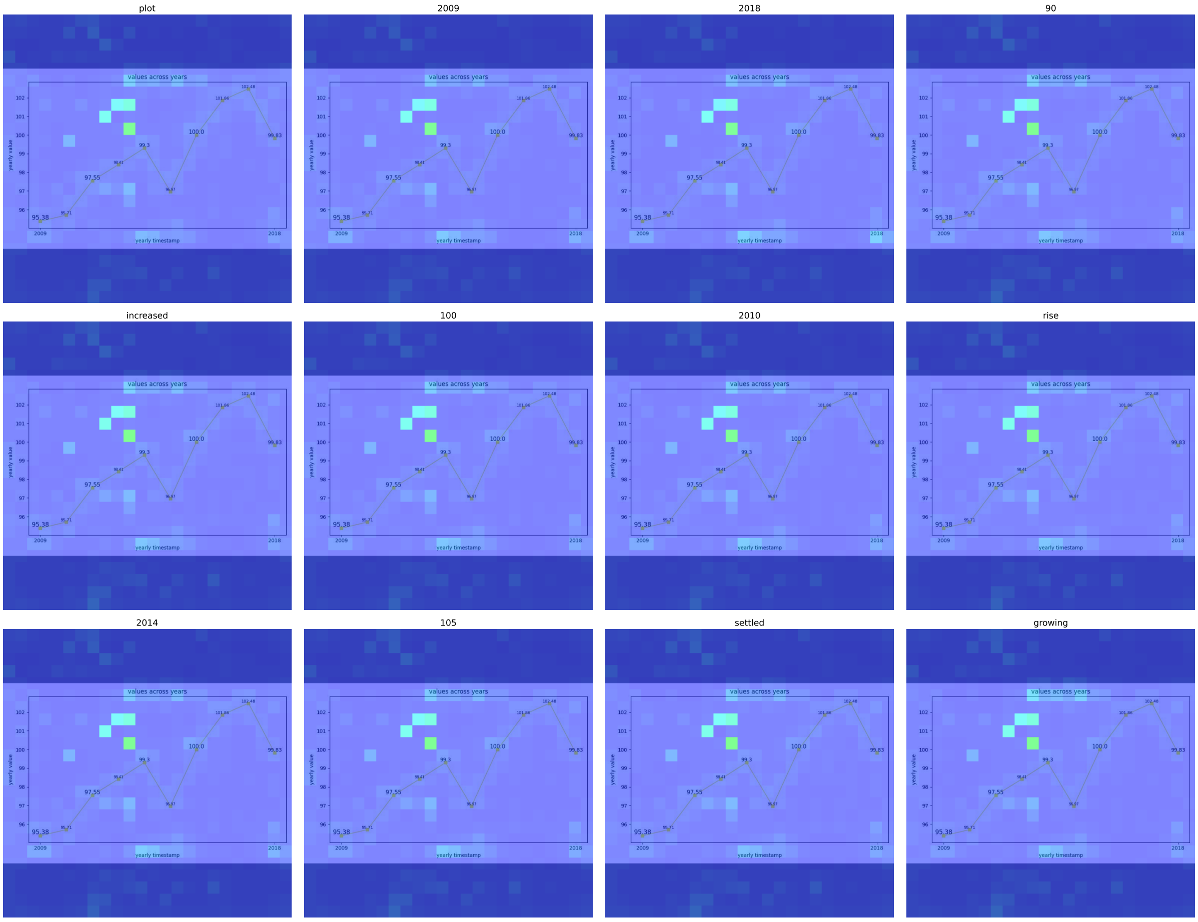


Figure 8.5. Word-level attention maps from `LLaVA v1.6 Mistral` on a time series plot without numeric time series present in the prompt.

8.3 Statistical Inference Failures & Success

Previously, we mentioned that finetuned models often become overconfident when inferring statistical properties such as means and standard deviations, despite lacking the capability to compute them accurately. In this section, we present two concrete cases that illustrate this overconfidence in practice, and one case where the finetuned model actually successfully inferred the statistics.

8.3.1 Case 1: Failure

The following error case shows the finetuned `LLaVA v1.6 Mistral` erroneously attempting to infer the time series mean, while its pretrained version does not. We compare the captions generated by both versions of the model against the numerically grounded reference caption.

Ground-Truth Caption

The Total Factor Productivity (TFP) index for Malawi from 2008 to 2017 reveals notable fluctuations, starting at 85.99 in 2008 and peaking at 110.47 in 2017, with 2015 set at 100.0. There's an initial rise from 2008 to 2009, followed by some volatility before a significant jump to 107.82 in 2013, indicating a period of strong productivity growth, and the mean of **98.93** for this period is significantly higher than the historical mean of **61.88**, suggesting a period of above-average productivity compared to Malawi's longer-term performance. Without global or regional TFP data for comparison, it's impossible to assess whether these trends are higher, lower, or follow expected patterns.

Caption from Pretrained Model

The time series for Malawi's Total Factor Productivity index shows a generally upward trend from 2008 to 2017, with some fluctuations. The index starts at 85.99 in 2008 and reaches a peak of 110.47 in 2014. It then experiences a decline, dropping below 100 in 2015 and 2016. In 2017, the index shows a slight increase, reaching 100.06. Compared to global or regional norms, Malawi's productivity index appears to be lower than expected, suggesting a need for economic development and productivity enhancement strategies.

Caption from Finetuned Model

From 2008 to 2017, Malawi's Total Factor Productivity index (2015=100) exhibited considerable volatility, starting at 85.99 in 2008 and ending at 110.47 in 2017. The index experienced a notable increase from 2008 to 2010, rising from 85.99 to 95.76, followed by a dip to 91.75 in 2011, before generally trending upwards with some fluctuations. Compared to the historical mean of **94.84**, the mean of this specific time series is moderately higher at **99.88**, suggesting a period of relatively stronger productivity growth than the broader historical context.

Figure 8.6. Case 1: finetuned model attempts to infer the mean, but it fails. Compare the **bolden** values.

We observe that, while the pretrained model refrains from inferring the mean of the time series, displaying a form of "self-awareness" about its limitations, the finetuned model tends to make confident guesses. This behavior underscores a key limitation of the cross-entropy loss objective. Since ground-truth captions often include the time series mean, the model can reduce its loss by always attempting to predict them, as numbers are treated like any other token. When the model generates captions with sentence structures resembling those in the ground truth, even erroneous guesses of these statistics incur less loss than omitting them entirely.

8.3.2 Case 2: Failure

The following error case presents one scenario where the finetuned `LLaVA v1.6 Mistral` also attempts to infer the standard deviation of the time series. Again, we compare the captions generated by the pretrained and finetuned versions of the model against the ground-truth caption.

Ground-Truth Caption

From 2016 to 2019, the number of severe motorcycle injuries in San Luis Obispo County showed a decreasing trend, starting at 32 in 2016 and declining to 23 by 2018. This contrasts with the average California county, which saw an increasing trend over the same period. The mean number of injuries in San Luis Obispo (**27.67**) was slightly higher than the average California county, but the standard deviation (**3.68**) indicates more variability in San Luis Obispo's numbers. The decline in San Luis Obispo is a moderate deviation from the statewide trend.

Caption from Pretrained Model

The time series shows a downward trend in the number of severe injuries on the road in Santa Barbara County from 2016 to 2019, with a significant drop from 194.0 in 2016 to 146.0 in 2019. This represents a 27.8% decrease in the number of severe injuries over the three-year period. The trend is consistent and shows a clear pattern of decline. Compared to global or regional norms, the number of severe injuries in Santa Barbara County appears to be lower than expected, indicating a positive trend in road safety in the area.

Caption from Finetuned Model

From 2016 to 2019, the number of severe road injuries in Santa Barbara County showed a decreasing trend, starting at 194 in 2016 and declining to 146 by 2019. This represents a 22% decrease over the four-year period. Compared to the average California county, which experiences a significantly higher mean of **300** severe injuries, Santa Barbara County's numbers are notably lower. The standard deviation of **19.79** in Santa Barbara indicates less variability in the number of severe injuries compared to the average California county, whose standard deviation is **48.7**.

Figure 8.7. Case 2: finetuned model attempts to infer the mean and standard deviation, but it fails. Compare the **bolden** values.

By examining the highlighted statistics, it is evident that the finetuned model's guesses are significantly inaccurate, yet presented with high confidence. In contrast, the pretrained model exercises caution, stating that the mean is lower than expected without attempting to provide a specific value. This comparison suggests that finetuning on the CaTS-Bench training data does not consistently improve the model's ability to perform accurate statistical inference—and may even encourage overconfident predictions.

8.3.3 Case 3: Success

The following is a success case where the finetuned `Idefics 2` is able to infer the time series mean accurately with a negligible error. We compare the captions generated by the pretrained and finetuned versions of the model against the ground-truth.

Ground-Truth Caption

The death rate in Papua New Guinea from 2017 to 2021 shows a relatively stable trend, fluctuating around a mean of **6.52** deaths per 1000 people, with a slight dip to 6.4 in 2020 and a peak of 6.68 in 2021. Compared to the global average, which steadily increased from 7.53 in 2017 to 8.57 in 2021, Papua New Guinea's death rate is significantly lower. While the global death rate saw a notable increase, particularly between 2020 and 2021, Papua New Guinea's rate remained relatively consistent, indicating a different trajectory than the global trend.

Caption from Pretrained Model

The yearly death rate in Papua New Guinea has been increasing since 2017. In 2017, the death rate was 6.51, and it has been increasing since then. In 2021, the death rate was 6.68, which is the highest value in the time series.

Caption from Finetuned Model

From 2017 to 2021, the death rate in Papua New Guinea, a lower-middle-income country, fluctuated slightly, ranging from a low of 6.40 in 2020 to a high of 6.68 in 2021. The average death rate during this period was **6.49**, with a standard deviation of **0.1**, indicating relatively stable numbers. Compared to the global average death rate, which decreased from 7.57 to 7.16 over the same period, Papua New Guinea's death rate was significantly lower and showed no clear downward trend.

Figure 8.8. Case 3: finetuned model successfully infers the mean and standard deviation within negligible error. Compare the **bolden** values.

Interestingly, the issue of statistical overconfidence appears to be model-specific, as different models exhibit varying behaviors after fine-tuning. In this case, the finetuned `Idefics 2` was able to infer both the mean and the standard deviation with reasonable accuracy, when even the ground-truth caption did not explicitly include the standard deviation. This signals that some models benefit more from finetuning on our training data.

Chapter 9

Conclusions

9.1 Conclusion

We introduce CaTS-Bench, a large-scale, multimodal benchmark for time series captioning and reasoning, integrating real-world time series data, rich metadata, plot images, numerically grounded captions from an oracle, and Q&A tasks to enable robust evaluation of VLMs on time series captioning and understanding. Our evaluations reveal that proprietary models generally outperform open-source models on time series captioning. Finetuned open-source models, however, can often match or even surpass proprietary models on linguistic metrics when trained specifically for caption generation, while also occasionally excelling in numerical metrics. Moreover, we identified a misalignment between captioning and Q&A tasks, where improvements in one skill do not transfer to the other, and captioning-focused finetuning can actually harm Q&A performance. Most models also show minimal reliance on visual inputs, performing similarly when visual input is removed, indicating heavy dependence on textual priors. On the landscape of time series analysis, CaTS-Bench lays the groundwork for future research on enhancing numerical fidelity, cross-modal alignment, and finetuning strategies, ultimately aiming to advance models that can generate accurate, grounded, and insightful narratives from complex time series data. Our dataset is publicly accessible on Hugging Face at https://huggingface.co/datasets/neurips2025submission/CaTS-Bench.

Bibliography

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR. Expert Certification.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. Model card, Anthropic, March 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

Saad Aziz. Population collapse. https://www.kaggle.com/datasets/saadaziz1985/population-collapse, 1985. Accessed: 2025-05-01.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

Rui Cao and Qiao Wang. An evaluation of standard statistical models and llms on time series forecasting. *arXiv preprint arXiv:2408.04867*, 2024.

Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469*, 2023.

Georgios Chatzigeorgakidis, Konstantinos Lentzos, and Dimitrios Skoutas. Multicast: Zero-shot multivariate time series forecasting using llms. In *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, pages 119–127. IEEE, 2024.

Daqing Chen. Online Retail. UCI Machine Learning Repository, 2015. https://doi.org/10.24432/C5BW33.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. Visionts: Visual masked autoencoders are free-lunch zero-shot time series forecasters. *arXiv preprint arXiv:2408.17253*, 2024a.

Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024b.

Lin Chin-Yew. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out, 2004*, 2004.

Kota Dohi, Aoi Ito, Harsh Purohit, Tomoya Nishida, Takashi Endo, and Yohei Kawaguchi. Domain-independent automatic generation of descriptive texts for time-series data. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

Jiaxiang Dong, Haixu Wu, Yuxuan Wang, Li Zhang, Jianmin Wang, and Mingsheng Long. Metadata matters for time series: Informative forecasting with transformers. *arXiv preprint arXiv:2410.03806*, 2024.

European Centre for Disease Prevention and Control. Download today's data on the geographic distribution of covid-19 cases worldwide. https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-distribution-covid-19-cases-worldwide, 2024. Accessed: 2025-04-03.

Elizabeth Fons, Rachneet Kaur, Zhen Zeng, Soham Palande, Tucker Balch, Svitlana Vyetrenko, and Manuela Veloso. Tadacap: Time-series adaptive domain-aware captioning. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 54–62, 2024.

Food and Agriculture Organization of the United Nations. Faostat - food balance sheets. http://www.fao.org/faostat/en/#data/FBS, 2024. Accessed: 2025-04-03.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models, 2023. URL https://arxiv.org/abs/2211.10435.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yasser Hassan. Walmart dataset. https://www.kaggle.com/datasets/yasserh/walmart-dataset, 2020. Accessed: 2025-04-03.

Ting-Yao Hsu, Chieh-Yang Huang, Ryan A. Rossi, Sungchul Kim, C. Lee Giles, and Ting-Hao Kenneth Huang. GPT-4 as an effective zero-shot evaluator for scientific figure captions. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=gVTtkPJbRq.

Kung-Hsiang Huang, Mingyang Zhou, Hou Pong Chan, Yi Fung, Zhenhailong Wang, Lingyu Zhang, Shih-Fu Chang, and Heng Ji. Do lvlms understand charts? analyzing and correcting factual errors in chart captioning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 730–749, 2024.

Abhishek S. Jha. Time series air quality data of india (2010–2023). https://www.kaggle.com/datasets/abhisheksjha/time-series-air-quality-data-of-india-2010-2023, 2023. Accessed: 2025-05-01.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. Truth-conditional captioning of time series data. In *EMNLP*, 2021.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Kai Kim, Howard Tsai, Rajat Sen, Abhimanyu Das, Zihao Zhou, Abhishek Tanpure, Mathew Luo, and Rose Yu. Multi-modal forecaster: Jointly predicting time series and textual data. *arXiv preprint arXiv:2411.06735*, 2024.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *Advances in Neural Information Processing Systems*, 37:87874–87907, 2024.

Chen Liu, Shibo He, Qihang Zhou, Shizhong Li, and Wenchao Meng. Large language model guided knowledge distillation for time series anomaly detection. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 2162–2170, 2024a.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered time series forecasting via cross-modality alignment. *CoRR*, 2024b.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

Haoxin Liu, Harshavardhan Kamarthi, Zhiyuan Zhao, Shangqing Xu, Shiyu Wang, Qingsong Wen, Tom Hartvigsen, Fei Wang, and B Aditya Prakash. How can time series analysis benefit from multiple modalities? a survey and outlook. *arXiv preprint arXiv:2503.11835*, 2025.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. *arXiv preprint arXiv:2403.07300*, 2024c.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Koki Maeda, Shuhei Kurita, Taiki Miyanishi, and Naoaki Okazaki. Vision language model-based caption evaluation method leveraging visual context extraction. *arXiv preprint arXiv:2402.17969*, 2024.

Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025.

Mike Merrill, Mingtian Tan, Vinayak Gupta, Thomas Hartvigsen, and Tim Althoff. Language models still struggle to zero-shot reason about time series. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3512–3533, 2024.

City of Los Angeles. Crime data from 2020 to present. https://catalog.data.gov/dataset/crime-data-from-2020-to-present, n.d. Accessed: 2025-05-01.

California Department of Public Health. Road traffic injuries narrative. https://catalog.data.gov/dataset/road-traffic-injuries-0935b/resource/72f5ab0d-9887-48d0-828d-67ab21661ca2, n.d. Accessed: 2025-05-01.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. s2 ip-llm: Semantic space informed prompt learning with llm for time series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

Hannah Ritchie. Many countries have decoupled economic growth from co2 emissions, even if we take offshored production into account. *Our World in Data*, 2021. https://ourworldindata.org/co2-gdp-decoupling.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. In *The Twelfth International Conference on Learning Representations*.

Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=DV15UbHCY1.

Benny J Tang, Angie Boggust, and Arvind Satyanarayan. Vistext: A benchmark for semantically rich chart captioning. *arXiv preprint arXiv:2307.05356*, 2023.

Hua Tang, Chong Zhang, Mingyu Jin, Qinkai Yu, Zhenting Wang, Xiaobo Jin, Yongfeng Zhang, and Mengnan Du. Time series forecasting with llms: Understanding and enhancing model capabilities. *ACM SIGKDD Explorations Newsletter*, 26(2):109–118, 2025.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Mohamed Trabelsi, Aidan Boyd, Jin Cao, and Huseyin Uzunalioglu. Time series language model for descriptive caption generation. *ArXiv*, abs/2501.01832, 2025. URL https://api.semanticscholar.org/CorpusID:275324039.

Bureau of Transportation Statistics U.S. Department of Transportation. Border crossing entry data. https://catalog.data.gov/dataset/border-crossing-entry-data-683ae, n.d. Accessed: 2025-05-01.

USDA Economic Research Service. International agricultural productivity. https://www.ers.usda.gov/data-products/international-agricultural-productivity, 2024. Accessed: 2025-04-03.

Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. *Advances in Neural Information Processing Systems*, 37:58118–58153, 2024.

Andrew Robert Williams, Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Jithendaraa Subramanian, Roland Riachi, James Requeima, Alexandre Lacoste, Irina Rish, Nicolas Chapados, and Alexandre Drouin. Context is key: A benchmark for forecasting with essential textual information, 2024. URL https://arxiv.org/abs/2410.18959.

Hao Xue and Flora D Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6851–6864, 2023.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 11121–11128, 2023.

Jingyang Zhang. Visualizing the attention of vision-language models. https://github.com/zjysteven/VLM-Visualizer. Accessed: May 2025.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Xiyuan Zhang, Ranak Roy Chowdhury, Rajesh K Gupta, and Jingbo Shang. Large language models for time series: a survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 8335–8343, 2024.

Yunkai Zhang, Yawen Zhang, Ming Zheng, Kezhen Chen, Chongyang Gao, Ruian Ge, Siyuan Teng, Amine Jelloul, Jinmeng Rao, Xiaoyuan Guo, et al. Insight miner: A time series analysis dataset for cross-domain alignment with natural language. In *NeurIPS 2023 AI for Science Workshop*, 2023.

Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

Zihao Zhou and Rose Yu. Can LLMs understand time series anomalies? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LGafQ1g2D2.