AutoMA: Automated Generation of Multi-level Annotations for Time Series Visualization

Qi Jiang* Guodao Sun† Tong Li‡ Jingwei Tang§ Wang Xia¶ Yunchao Wang‖ Li Jiang** Ronghua Liang††

College of Computer Science and Technology, Zhejiang University of Technology

2025 IEEE 18th Pacific Visualization Conference (PacificVis) | 979-8-3315-0581-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/PacificVis64226.2025.00014

ABSTRACT

Time series data is ubiquitous in people's daily production and life, and visualizations augmented with annotations can significantly facilitate the understanding of such data and promote downstream tasks. Consequently, numerous annotation tools have been developed to detect and narrate useful patterns within time series visualizations. However, most existing tools can only identify basic factual insights (e.g., increasing or decreasing trends) that are already present in the charts. When users need deeper insights (e.g., predicting future trends) and richer contextual information (e.g., associative patterns between dimensions within and beyond the chart), these tools often fall short. To address this challenge, we present AutoMA, a system that automatically generates multi-level annotations for time series visualizations. We introduce an LLM-based pattern extraction method that supports the identification of seven distinct temporal patterns. Furthermore, we present a multi-level annotation design space that encompasses six specific annotation tasks, aimed at delivering richer contextual information. The generated annotations span a spectrum of information, ranging from directly observable temporal patterns to deeper insights obtained through further computation, and ultimately to advanced inter-dimensional association patterns. Finally, we demonstrate the effectiveness of our approach through experiments and user evaluations. The results indicate that AutoMA significantly enhances users' ability to comprehend and explore time series data.

Index Terms: Human-centered computing—Visualization—Visualization application domains—Visual analytics; Human-centered computing—Human computer interaction (HCI)—Interactive systems and tools;

1 INTRODUCTION

In real-world scenarios, the time series (TS) data that people explore and analyze are often multidimensional (i.e., multidimensional time series data, MTSD), but they typically focus on analyzing only a few dimensions at a time [58, 59]. During the process of users visually analyzing data in these specific dimensions, researchers have found that adding annotations to the output visualizations can significantly enhance users' understanding and insights into the data [11]. Consequently, numerous visualization annotation tools have been developed to describe and emphasize important patterns and features within visualizations [13, 23, 31, 32, 45, 66].

*e-mail: jiangqi@zjut.edu.cn,
†e-mail: guodao@zjut.edu.cn, Guodao Sun is the corresponding author
‡e-mail: litong@zjut.edu.cn
§e-mail: jwtang@zjut.edu.cn
¶e-mail: xiawang@zjut.edu.cn
‖e-mail: wyctears@gmail.com
**e-mail: jl@zjut.edu.cn
††e-mail: rhliang@zjut.edu.cn

However, most existing annotation tools concentrate on describing the obvious basic facts in charts (e.g., increasing or decreasing trends), without providing more in-depth and comprehensive contextual information that users need. Apart from the visual information directly presented in the visualization, the potential external contextual information is rich and diverse [44]. For instance, the rendered data in existing visualizations can be further processed to form computational insights (e.g., statistics and predictions); the dimensions presented (observed) in the chart can form association patterns with other unobserved attributes in the dataset; as well as higher-level background information derived from knowledge documents or human insights. Comprehensively utilizing this contextual information from different sources and forms can enhance the information content of annotations, which helps users form more structured knowledge [38, 51]. Nevertheless, to our knowledge, there is still a lack of research and practice in this area. Therefore, how to effectively integrate richer contextual information in annotations remains a challenge.

Furthermore, identifying visually salient patterns in charts is often necessary for visual annotations to help users quickly acquire important information from visualizations [64]. However, most existing annotation tools still employ traditional methods (e.g., based on statistical characteristics) for identifying these visual patterns [12, 23], which poses challenges in terms of scalability and generalizability. Although some recent works have attempted to enhance generalizability using machine learning techniques [21, 35, 63], they often face limitations related to the difficulty in acquiring suitable and available datasets. The recent rise of large language models (LLMs) has provided new perspectives for addressing numerous tasks, but to our knowledge, research on leveraging LLMs for time-series pattern recognition remains limited.

To address the above challenges, we present AutoMA, a system that automatically generates rich contextual annotations for single-attribute time series visualizations. The focus on time-series data is driven by their prevalence in both everyday life and scientific practice. Focusing on a single attribute allows us to manage problem complexity effectively while laying a methodological foundation for extending to other data types. AutoMA introduces a pattern extraction method based on LLMs that supports the identification of seven typical temporal patterns. Considering the lack of readily available training data for LLMs in the temporal pattern recognition task, we collaborate with domain experts to iteratively construct a synthetically generated temporal pattern dataset for fine-tuning the model. Furthermore, we propose a multi-level annotation design space encompassing six specific annotation tasks to provide users with richer contextual information. AutoMA also offers a flexible annotation interaction and editing environment, allowing users to interactively choose annotation content of interest as needed. Additionally, we implement an auto-scaling feature to adapt to changes in the data range. When the annotation content changes, the system automatically tracks the numerical range and readjusts the x-axis and y-axis ranges of the visualization to avoid excessive whitespace on the screen due to extreme data distributions, thereby enhancing the information communication ratio of the visualization view. We also provide a chart export function, enabling users to share the edited charts with collaborators. Finally, we validate the effectiveness of

979-8-3315-0581-3/25/$31.00 ©2025 IEEE
DOI: 10.1109/PacificVis64226.2025.00014

AutoMA through experiments and user studies.

In summary, our contributions are as follows:

- We propose a novel temporal pattern recognition method based on LLMs, which is achieved by fine-tuning an LLM on a large-scale, synthetically generated dataset with a wide distribution of temporal patterns.
- We extend the content dimensions of visualization annotations and propose a multi-level annotation design space to provide users with more context-rich annotations.
- We present AutoMA, a runnable system that supports multi-level annotations, and evaluate the effectiveness of the proposed methods through experiments and user studies, confirming the application potential of LLMs in the field of temporal pattern recognition.

2 Related Work

In this section, we review the literature closely related to our study, specifically focusing on three key areas: temporal pattern concepts, temporal pattern recognition, and visualization annotations.

2.1 Temporal pattern concepts

As a special type of data pattern, temporal patterns play a crucial role in TS analysis. According to the definition by Andrienko et al. [5], a temporal pattern involves elements of at least two sets - usually time units and corresponding measurements. The relationships within and across these sets determine the types of temporal patterns. In analytical practice, researchers often employ graphical representations and a pattern can be considered a specific graphical shape generated from this process [1]. Thus, temporal patterns can also be identified by recognizing these shapes. Existing research has investigated temporal patterns from various perspectives. For instance, Gota Shirato et al. [50] proposed that the basic units of temporal patterns can be categorized into "up", "down", and "constancy", and more complex patterns can be composed by combining these fundamental units. On the other hand, Alexander Bendeck et al. [9, 12] focused on quantifying time-series trends to help users understand the subtle differences between various trends. Furthermore, temporal patterns in specific domains (e.g., stock market [14], electroencephalography [15], and electrocardiography [40]) have been extensively studied. It is not difficult to find that people's desired temporal patterns vary greatly across different domains, making it impossible to develop a one-size-fits-all pattern extraction program that can satisfy all tasks. Given this, we currently focus on seven common temporal patterns to provide a more focused research scope. These patterns encompass the mainstream temporal patterns in current visualization research (i.e., increasing and decreasing trends) and expand to include some more complex temporal shapes (e.g., double peaks and double valleys). Overall, by analyzing these representative temporal patterns, we hope to explore a universal temporal pattern extraction framework, laying the foundation for subsequent expansion to more temporal patterns.

2.2 Temporal pattern recognition

Identifying patterns in TS is a widely investigated problem that primarily involves two tasks: time series segmentation (TSS) and pattern type recognition [30, 37]. The former involves partitioning a time series into meaningful segments, while the latter focuses on identifying the pattern types to which these segments belong. Researchers often approach the TSS problem as finding a series of change points (CPs) in TS to partition it into homogeneous segments [4]. These CP detection-based approaches commonly model the task as an optimization problem, designing cost functions to search the solution space for the optimal solution [16, 54]. Based on the segmentation search strategy employed, TSS methods can be further categorized into "sliding window", "top-down", "bottom-up" approaches [37].

Regarding the task of identifying temporal pattern types, the simplest and most intuitive approach is to discriminate based on statistical features [12, 23], such as determining whether a sequence exhibits an upward or downward trend by calculating the slope. Furthermore, Gota Shirato et al. [50] proposed determining the shape of a TS by calculating the largest triangle within it. However, these methods often have limited scalability and can only handle limited pattern recognition scenarios. Another commonly used class of methods is based on similarity matching, which determines the pattern category of a TS by measuring its similarity to predefined patterns using distance metrics such as Euclidean Distance (ED) or Dynamic Time Warping (DTW) [10]. This approach has been widely applied in the field of TS retrieval, where users can provide a rough shape of the target sequence through sketching or other means, and then utilize similarity metrics to quickly retrieve sequences that are similar to it [36]. Most of the methods introduced above rely on prior knowledge of patterns. In contrast, there are also works where time-series patterns are identified implicitly by means of clustering [16].

The aforementioned methods mostly depend on domain-specific model parameters or make specific assumptions about the value distribution of the TS. In recent years, data-driven machine-learning approaches have emerged as a promising direction for temporal pattern recognition. Over the past decade, the effectiveness of deep learning models(e.g., long short-term memory (LSTM) [22] and temporal convolutional network (TCN) [7]), has been demonstrated across various domains [21,35,63]. Similarly, the recently developed LLMs have also demonstrated impressive performance in various tasks. However, some research works [2, 42, 53] have revealed that the performance of LLMs on certain specific tasks (e.g., time series prediction) does not always exhibit superior results compared to deep-learning methods. Moreover, to our knowledge, there is a lack of studies investigating the practical application of LLMs to temporal pattern extraction tasks and validating their effectiveness in this context. To bridge this gap, in this work, we employ a novel LLM-based time-series pattern extraction method. Our approach involves fine-tuning LLMs using artificially synthesized data to enable the classification of seven common temporal patterns. Overall, on the one hand, this work expands new directions for temporal pattern recognition techniques; on the other hand, our experimental results preliminarily validate the potential and effectiveness of LLMs in temporal pattern discovery.

2.3 Visualization annotations

Visualization annotation is an effective technique to enhance users' understanding of visualized content [52]. It typically refers to adding additional textual or graphical elements to the visualization interface to help explain or highlight specific patterns. Based on the degree of human involvement in the annotation process, existing annotation techniques can be roughly divided into two categories: user-driven and fully automated. User-driven annotation allows users to determine the annotation content independently while incorporating automated techniques to reduce manual effort. Representative works include Click2Annotate [17], ChartAccent [45], Graphical Overlays [31], and Inksight [33]. The advantage of this approach is that users can flexibly control the content and form of annotations, but it also potentially relies on users' domain expertise. In contrast, fully automated annotation requires no human intervention and mainly uses algorithmic models to automatically identify key visual features in the visualization and convey this information through textual descriptions. Chart summarization is one such method, except that it often provides a more comprehensive description of the chart content, including both an overall introduction and a depiction of salient features. Representative works in this category include AutoCaption [34], LineCap [39], and so on. Additionally, there are some other fully automated annotation generation tools. For example, Contextifier [23] leverages news articles to automatically generate customized, annotated stock behavior visualizations. Lai et al. [32] employ deep learning to recognize visual elements in charts and match the element information with pre-provided contextual

description text, ultimately presenting annotations in a narrative manner.

Although the above works have different focuses on the generation mechanism of annotation content, it is not difficult to find that the content they generate mostly concentrates on the interpretation of the presented data, i.e., so-called *observation* annotations. Despite some works attempting to introduce *additional* background information or human insights into chart annotations, this information is primarily presented in textual form and still focuses on describing or explaining the visualization content itself [13, 23]. In fact, the insights implied by charts and visualizations are not limited to the data directly presented; the potential connections or patterns between the visualized data and other unexposed attributes in the source dataset can also serve as valuable annotations [44]. Unfortunately, few existing works have considered this point. To fill this gap, this paper proposes a novel three-level annotation design space that provides annotations at the following levels: explicitly presented data in the chart, computations or inferences based on displayed data, and source dataset of the chart (i.e., MTSD). Moreover, AutoMA incorporates an auto-scaling feature to ensure an appropriate amount of information in the visualization view and enables users to interactively select annotation contents on demand. Overall, by extending the focus of annotation to a deeper and broader data semantic space, our method not only promotes users' comprehensive understanding of the current visualization chart but also has the potential to inspire and guide users to conduct deeper exploration.

3 AUTOMA

In this section, we first introduce the data background - MTSD. Next, we discuss the key design goals that our intelligent multi-level annotation system for MTSD should achieve. Finally, we present the system pipeline, which describes the functional modules within our system.

3.1 Data background

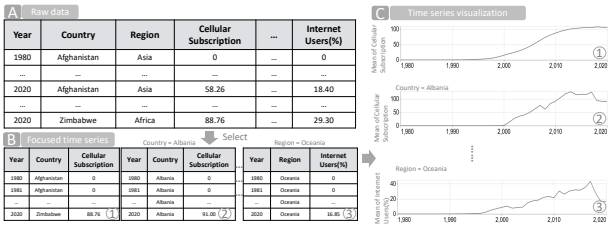

Figure 1: The illustration of multidimensional time series data and its visualization examples.

In practice, the time-series data analyzed are usually multidimensional, and such data refers to a collection of data points recorded across multiple dimensions over time, typically presented in a tabular format. As illustrated in the Fig. 1-A, each row in the table corresponds to a data record at a specific time point, while columns represent different attribute dimensions or variables. These dimensions can be temporal (e.g., "Year"), numerical (e.g., "Cellular Subscription", "Broadband Subscription"), or categorical (e.g., "Country", "Region"). In this study, we primarily focus on visualizing single-attribute time-series data. This decision is based on the following considerations: First, analyzing single attributes often serves as the foundation for building a comprehensive understanding of complex MTSD, as it reduces the cognitive load on users and allows them to perform in-depth and fine-grained exploration of each attribute [55, 61]. Second, focusing on single-attribute analysis helps manage the research complexity by simplifying the visualization design and implementation process. During the analysis, users' perspectives can be flexible and dynamic (Fig. 1-B). Sometimes, they may select an entire attribute column to gain a global overview of the temporal changes within that dimension (Fig. 1-①). In other instances, users may focus their attention on a subset of data by applying filters (e.g., "Country = Albania" and "Region = Oceania") to the original data (Fig. 1-② and ③). Building upon these visualizations, we further explore how to automatically generate multi-level contextual annotations to enhance users' understanding of MTSD. To facilitate the subsequent technical discussion, we use the data shown in Fig. 1 as an illustrative example and make the following definitions: the MTSD in Fig. 1-A serves as the source data/dataset for user-focused visualizations (Fig. 1-①, ②, and ③), providing additional contextual background information.

3.2 Design Goals

The aim of this research is to enhance the understanding of time series visualizations among general audiences by automatically generating contextual annotations, thereby promoting in-depth analysis of temporal data. Visualization annotation is a well-established research field, with numerous works conducting thorough explorations and practices [13, 23, 25, 32, 33, 45, 49]. Based on a survey of existing literature, we summarized a set of design requirements as presented in *Appendix*-Section 2. Considering the alignment with our research objectives and preliminary feasibility, we currently focus our research scope on the most essential and fundamental tasks, and customize them based on our specific scenarios. Although additional advanced tasks such as optimizing annotation layout and enhancing annotation narration contribute to the construction of superior annotations, these aspects are currently beyond the scope of the present research and will be explored in future endeavors. Specifically, we established the following design goals to better guide the development of our system:

DG1 Temporal patterns automatic extraction. The first step in generating visual annotations is typically identifying the objects to be annotated [13, 23, 33], which directly determines the content and form of the generated annotations. From the user's perspective, they often desire this process to be automated. To meet this requirement, our system should be capable of automatically detecting and extracting meaningful patterns (e.g., trends, peaks, and valleys) from the data as annotation targets.

DG2 Multi-level contextual annotations generation. Enriched context helps users to form a more comprehensive and structured knowledge of the visualization [38, 51]. To deepen users' understanding of the visualized information, our system should generate multi-level contextual annotations that go beyond the information directly described in the charts. This can be achieved by mining extra contextual patterns to construct more synthesized annotations.

DG3 Interactive user engagement. While automatic pattern extraction and annotation generation are fundamental functionalities, allowing users to adjust and edit annotations according to their own needs is equally critical [33]. Specifically, the system should enable users to select the data dimensions they wish to analyze and annotate, as well as enable them to prioritize the content of generated annotations based on their perceptions.

DG4 Convenient insight communication. In today's collaborative work environments, facilitating the sharing of insights and discoveries among team members and stakeholders is crucial [48,56]. Therefore, our system needs to provide not only convenient annotation editing features but also allow users to easily export annotated charts for sharing with others.

3.3 System pipeline

To achieve the design goals outlined in Sect. 3.2, the AutoMA system pipeline is constructed as illustrated in Fig. 2. AutoMA adopts a frontend-backend architecture, with blue arrows indicating user-driven interactions and orange arrows representing automated system processes. The core components of AutoMA are the temporal pattern detection and multi-level annotation generation modules. The temporal pattern detection module utilizes fine-tuned LLMs to identify salient visual patterns within the user-specified visualizations (**DG1**). Following this, the multi-level annotation generation module automatically produces six types of contextual annotations from various annotation data sources and levels (**DG2**). These annotations are then presented in the frontend annotation editing interface, enabling

users to refine the annotation content as needed (**DG3**). Additionally, the system offers an annotation export feature, allowing users to save and share annotated charts for various purposes, such as collaboration and communication (**DG4**).

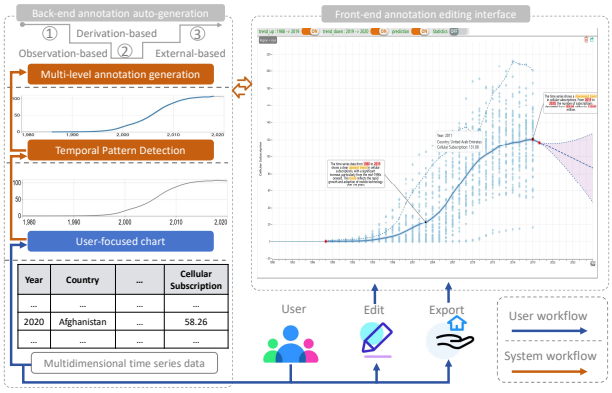

Figure 2: The pipeline of the AutoMA system, which contains two main components: temporal pattern extraction and multi-level annotation generation.

4 Temporal Pattern Extraction

This module aims to identify salient visual content in temporal visualizations, providing anchoring objects for subsequent annotation processes. To achieve this, we propose a novel approach based on LLMs. Given the vast pattern space in time-series data and the varying definitions of meaningful patterns across application domains, we focus on seven common temporal patterns [9, 12, 14]: *increasing*, *decreasing*, *single-peak*, *single-valley*, *double-peak*, *double-valley*, and *periodic* patterns (Fig. 3-B). Currently, LLMs are primarily applied in two ways: prompt-based and model fine-tuning. In this work, we opt for the latter, as fine-tuning enables better adaptation to specific tasks and yields more stable performance. However, achieving optimal fine-tuning results often requires curated datasets, which are scarce and costly to annotate in reality. Next, we elaborate on constructing a temporal pattern dataset and leveraging LLMs to recognize patterns in TS.

4.1 Dataset

Although there exist several publicly available time series datasets for temporal pattern analysis [14, 15, 40], most of them originate from specific industry domains, resulting in domain-specific temporal patterns. However, in this work, our focus is on identifying general time-series patterns that are more prevalent in everyday production and life. To our knowledge, there is a lack of such a well-customized dataset. In machine learning research, when the desired dataset is scarce and expensive to acquire, utilizing artificially synthesized datasets for model training is a viable solution [8, 18, 47, 65]. Synthetic datasets offer the advantages of lower costs and flexible control over data quality and quantity. Particularly, when the desired time-series patterns are clearly predefined, this prior knowledge imbues the synthetic data generation process with controllability and purposefulness. Moreover, compared to other data types, time-series data synthesis possesses unique advantages due to its explicit temporal characteristics, which make the results less ambiguous and uncertain than other data types (e.g., text). To further enhance the quality of the synthesized dataset, we collaborate with two data experts with over five years of experience in data analysis. Through face-to-face collaborative discussions, we facilitated the exchange of ideas and the collision of insights between the experts. This approach helps to balance individual cognitive differences and mitigate potential subjective biases that may arise from a single reviewer. The two experts serve as quality assurance inspectors for data generation, while we act as coordinators, developing programs to construct the dataset and incorporating expert feedback for iterative improvement.

The specific data synthesis steps are as follows: First, a basic timing sequence following a normal distribution is generated as background noise. Subsequently, one of the seven predefined temporal patterns is randomly chosen and superimposed onto the background noise sequence at a random position (orange rectangular box portion of A1 and A2 in Fig. 3). The probability of occurrence for each pattern is roughly balanced to ensure an approximately uniform distribution of patterns within the dataset. To enhance the model's robustness, we introduce various randomized parameters to simulate real-world scenarios, ensuring good generalization capabilities when applied to real data [26]. These randomized parameters include different noise levels, pattern time spans, value ranges, and interpolation functions (e.g., linear, power, exponential) for generating pattern sequences. It is important to note that the tuning of these parameters is an iterative process, e.g., excessively high noise levels can hinder pattern recognition, while insufficient noise may not achieve the desired robustness (see supplementary material for a list of these parameters currently used). In each iteration, we randomly generate one hundred test samples based on the current production plan, which are then reviewed by two experts under the moderation of the coordinator. The primary review criteria include assessing whether the generated results align with the prior knowledge of these time-series patterns and the reasonableness of the parameters. Based on their feedback, we continuously refine the dataset. This iterative process lasts for approximately ten rounds, with Fig. 3-A1, A2 illustrating an exemplary enhancement made during one such cycle. In early versions (Fig. 3-A1), after adding a pattern, the subsequent sequence abruptly reverts to the initially set noise level. Experts suggest that after adding a pattern, the mean value of the noise should smoothly transition to the value at the end of the pattern, making the results more realistic (Fig. 3-A2). Only when both experts are satisfied with the synthesized dataset is it used for subsequent LLMs fine-tuning. The synthetic dataset we currently use and its statistical description can be found in the supplementary materials.

4.2 LLM-based temporal feature extraction

Ultimately, the obtained dataset is formatted as follows: the input is a TS normalized to the range of 0-1, and the output is a tuple representing the pattern added to the sequence, where the first element records the pattern category and the second records the index information of the pattern. Based on this dataset, we fine-tune the LLM (Fig. 3-C). Currently, we do not provide additional prompt descriptions for the input sequences. This decision is based on our initial attempts, which revealed that adding simple background descriptions in the prompts did not noticeably enhance the performance of the fine-tuned model. Certainly, we also explored other improvement strategies. We find that including additional randomly sampled points (between the start and end indices) in the output pattern index information helps improve the model's accuracy. This improvement may be attributed to these random intermediate points, which make the model more attentive to the start and end positions of the patterns, thereby increasing its sensitivity to positional information.

5 Multilevel Annotation Generation

Based on the insights from Rahman et al.'s research [44], we propose a novel multi-level annotation design space (Fig. 4) according to the data sources utilized for the annotations: observation-based, derivation-based, and external-based. These three levels correspond to the data sources of annotations as data directly presented in the chart, derived calculations based on the visually presented data, and information beyond the chart itself, respectively. Although both derivation-based and external-based annotations involve computational processes, they differ in the scope of the computations involved. The former focuses on deriving new insights (e.g., statistics and prediction) by leveraging the visualized data within the chart, while the latter goes beyond the chart to incorporate external attributes for generating contextual patterns. It is also worth noting that our definition of external-based annotations differs from that of Rahman et al [44]. In our design space, external data sources specifically refer to information from the MTSD where the current chart

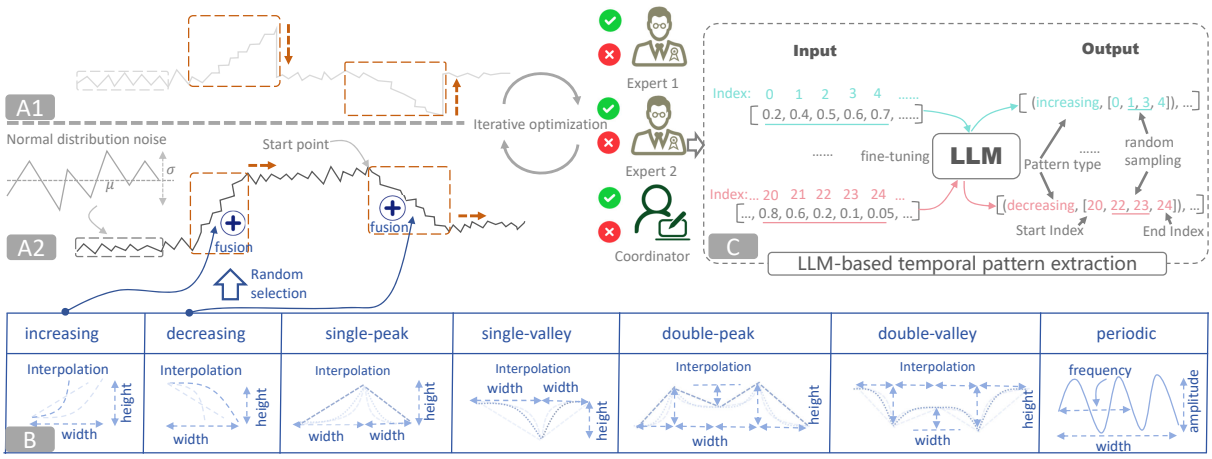

Figure 3: The temporal pattern extraction module. we first collaborate with domain experts to construct a dataset containing seven distinct temporal patterns. The curated dataset is then used to fine-tune LLMs.

data originates, rather than broader background articles or personal insights. This is because visualizations are generally created based on a source dataset (i.e., MTSD), so the source dataset is usually the most readily available external data source. Moreover, the source dataset may contain rich contextual patterns that are useful for an in-depth understanding of the chart. Finally, existing pattern mining theoretical methods support extracting reliable contextual patterns from the source data [3, 20, 27–29, 48, 56].

Annotation level	Data Sources	Annotation Tasks	Granularity	Calculation Method
Observation-based	Directly presented data	Text description	Local	/
Derivation-based	Derived calculations based on the presented data	Statistics	Global,Local	Aggregation
		Prediction	Global	Arima
External-based	Source dataset of the visualization	Details	Local	/
		Correlation	Local	Pearson Coefficient
		Lag	Local	DTW Distance

Figure 4: The annotation design space, which contains six annotation tasks at three levels.

More specifically, in our proposed three-level annotation design space (Fig. 4), each level can be further divided into several specific annotation tasks, totaling six annotation tasks. An annotation example can be represented in the form of <anchor object, annotation content>. The anchored object of the annotation generally only targets the visually salient intervals detected in the chart (marked as "Local" in "Granularity" column of Fig. 4), which helps to guide the user's attention toward key areas. Only derivation-based annotations may target the entire sequence data presented in the chart (e.g., aggregating results for the entire line chart, predicting future trends based on existing historical data in the line chart), marked as "Global" in Fig. 4. Note that a detected pattern (i.e., an anchoring object) may be associated with multiple annotations, and users can interactively inspect and add this contextual information. In addition, the generation process of certain annotations will use computational methods to generate indicator-like results (see "Calculation Method" column in Fig. 4). Next, we will describe these annotations in detail.

5.1 Observation-based

Observation-based annotations focus on the content directly presented in the visualization without additional computation or input. In this annotation level, only one task type is included: text annotation, which is the most frequently used and common task type in this scenario [23]. In actual implementation, we also use visual encodings to highlight the annotated elements. However, we do not classify this visual emphasis as a separate annotation task because it usually does not convey additional information. Its main purpose is visual guidance, serving as an assistive visual design technique. In this work, the annotation tasks discussed should provide a certain amount of information increment to help users acquire new knowledge.

Text annotations are used to provide direct textual descriptions of the patterns observed in the dataset. These annotations not only reveal the overall patterns visible at the visual level but also deliver details that visual inspections alone cannot achieve. Currently, we do not provide textual annotations for the entire sequence; instead, we focus on describing specific detected patterns. This approach is taken to avoid redundancy in text descriptions between the overall TS and individual patterns and to prevent clutter on the limited screen space.

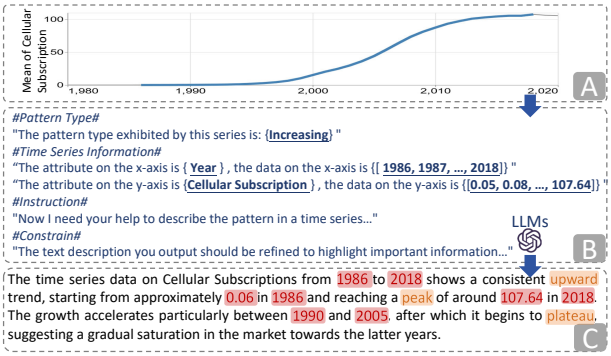

Figure 5: The demonstration of text annotation generation, which uses LLMs to generate descriptive information and highlights the key descriptive information through color.

We use an LLM-driven approach (GPT-4) to generate text annotations. This decision is primarily based on two considerations. First, LLMs possess advanced contextual understanding capabilities, enabling them to accurately produce text that aligns with the contextual patterns. Second, the text generated by LLMs exhibits human-like characteristics, providing a more natural and fluent reading experience compared to template-based methods. Specifically, we input detected pattern types, related data, and other contextual information into the LLM, crafting prompts to guide the model in generating both informative and concise descriptions (Fig. 5-B). To further enhance the readability and information communication efficiency of the text annotations, we implement a visual optimization measure by selectively highlighting key information within the generated text using distinct colors (Fig. 5-C). For instance, we use orange to mark pattern information (such as rapidly increasing or decreasing trends) and red to highlight numerical information (including time intervals and attribute values). To achieve this, we pre-construct a keyword

lexicon, then examine the generated text annotations, identify the key information within them, and apply the corresponding color highlighting.

5.2 Derivation-based

Derivation-based annotations are a commonly used method for deriving deeper insights from data visualizations beyond what is directly represented. Here, they are primarily divided into two categories: statistics annotations and prediction annotations. This classification arises because, in the context of time-series visualization, statistical analysis and forecasting are highly prevalent and significant computational tasks [44, 57]. The following are specific descriptions of them.

Statistics annotations generate statistical metrics that can quantitatively describe data characteristics (such as mean, median, etc.) by calculating presented data [44]. Such annotations serve to provide accurate statistical attributes that are difficult to perceive visually, thereby helping users gain a deeper understanding of the intrinsic characteristics of the data. The structure of statistical annotations also consists of two parts <statistical scope, statistical result>. The statistical scope defines the specific region of the data being analyzed, while the statistical result represents the computed metric or measure. These messages are highlighted in the visualization with auxiliary lines and text to help users identify (as shown in Fig. 6).

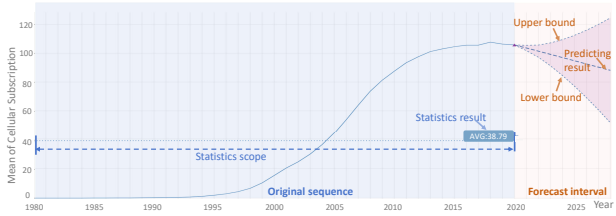

Figure 6: Derivation-based level annotations, which include statistics annotations and prediction annotations.

Prediction annotations aim to forecast and extrapolate future trends by leveraging historical data and statistical models [57]. However, real-world time-series data often exhibit complex and varying characteristics, and their future trajectories inherently involve a considerable degree of uncertainty. Therefore, the purpose of prediction annotations is not to provide precise future values but to describe the general trajectory and possible range of future changes. Although this trend forecast is not a deterministic conclusion, it can provide valuable references for decision-making.

The key to the forecasting lies in selecting an appropriate predictive model. Currently, we employ the Automatic Autoregressive Moving Average (AutoARMA) [24] model to dynamically estimate future data points. The rationale behind choosing AutoARMA is that it automatically identifies the optimal model orders p and q by computing the autocorrelation function and partial autocorrelation function. This enables the model to adaptively adjust its structure and parameters to accommodate TS with different characteristics. To enhance the reliability of the predictions, we also calculate prediction intervals with a 95% confidence level, visually presenting the range of potential future data fluctuations (Fig. 6). Moreover, the choice of prediction horizon is also a trade-off issue. By default, the prediction horizon is usually set to one-fifth of the current series length (can be adjusted as needed), which is a compromise between ensuring prediction timeliness and controlling the propagation of prediction errors.

5.3 External-based

External-based annotations expand the informational dimensions of visualizations by discovering and introducing contextual associations between the source data and the annotated charts. Drawing on existing research on data insights and data facts [3, 20, 48, 56], we select three annotation tasks closely related to temporal relationships: **details**, **correlation**, and **lag**. These three annotation tasks are chosen because of the valuable information they convey and their potential to enable the realization of other tasks. For instance, details annotations may potentially facilitate anomaly detection, while correlation and lag annotations can help uncover causal relationships. The following is a detailed introduction to these annotation tasks.

Details annotations provide more fine-grained information about the observed patterns in time-series data. A significant characteristic of time-series data is that it usually contains a large number of data points, with each time point potentially having numerous records. To better observe overall trends, users often perform aggregation operations on the observed dimensions (e.g., calculating statistical measures like mean, minimum, and maximum). However, such aggregations inevitably lead to the loss of detailed information. It is also worth noting that the same aggregated results may originate from various data distributions. As shown in Fig. 7-A, for the same "Cellular Subscription" average curve, the actual corresponding data may be widely distributed (Fig. 7-①), concentrated (Fig. 7-②), or even include some outliers (Fig. 7-③). Although these different distribution scenarios appear consistent in the aggregated results, they have different implications for analysis and decision support. Therefore, enabling users to observe data details is essential. With the help of details annotations, analysts can identify underlying distribution characteristics, uncover potential anomalies, and form a more comprehensive and accurate understanding of the observed patterns.

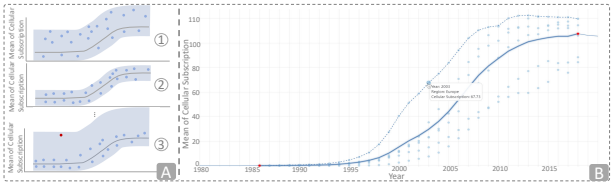

Figure 7: The showcase of details annotations, which provides a finer-grained view of the underlying data distribution.

When generating detail annotations, we select categorical fields from the data as labels, which determine the granularity of the data presentation. For instance, choosing "Country" as a label, compared to "Region", can reveal a more detailed data breakdown. Users can interactively adjust these detail labels to meet their analysis needs. For the visual presentation of detail data, we employ two distinct strategies based on the number of label categories. When the number of categories is below a threshold (here, we refer to the value of seven proposed by Miller et al. [41]), the system displays all the detail data points along with their corresponding category connecting lines. However, when the number of categories surpasses the threshold, we opt to display only the detail data points without drawing connecting lines. The rationale behind this decision is that plotting multi-label category curves in the presence of a large number of categories can easily lead to visual clutter, which may hinder users' comprehension of the data. In such cases, to enhance users' perception of data trends for specific categories, we designed an interactive detail inspection mechanism. When users examine the detailed information of a specific data point, the system automatically connects that point with other data points of the same category and highlights their trend changes over the selected time period. This targeted interaction approach helps users keep track of the change patterns in categories of interest while maintaining interface clarity and readability. Fig. 7-B is an example, when a user examines the "Cellular Subscription" for Europe in 2003, the system highlights and connects all data points for Europe within the annotated period, revealing trends in "Cellular Subscription" across the Europe region.

Correlation annotations reveal potential relationships between data by comparing the changing trends across different dimensions. These annotations help users quickly identify significant patterns of synchronous increases/decreases or inverse movements within specific time periods. Although correlation itself does not directly

equate to causation, it can serve as a valuable clue, guiding users to investigate the possible reasons behind particular changes.

In this work, the generation of correlation annotations considers two perspectives of association: attribute-oriented and subset-oriented. The attribute-oriented association focuses on other attribute fields parallel to the current analysis object. For instance, when analyzing "Cellular Subscription", related attributes such as "No. of Internet Users" and "Broadband Subscription" may be considered. However, integrating these heterogeneous attributes into a unified visualization view poses the challenge of handling different scales. Traditional solutions often involve setting multiple y-axes for attributes with varying scales, but this approach may further exacerbate visual complexity and hinder users' insight into data patterns. To optimize the visualization effect and enhance users' understanding of data correlations, we introduce a mapping strategy. Specifically, we use the position of the current analysis attribute in the chart (i.e., the orientation of the x-axis and y-axis) as a reference and remap the data points of all associated attributes to align with it. Through this mapping, data points of different attributes achieve visual alignment, facilitating intuitive comparison and correlation analysis (Fig. 8-C). Moreover, to help users accurately identify the scale differences among attributes, we explicitly label each associated attribute and attach specific numerical annotations to each data point.

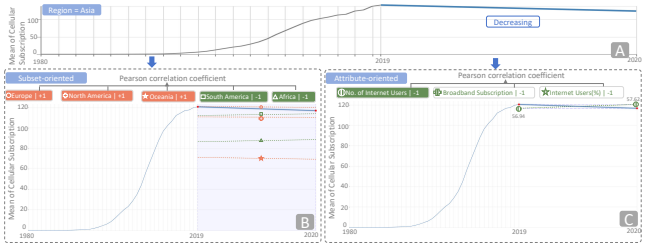

Figure 8: The demonstration of correlation annotations, which provides correlation information from attribute-oriented and subset-oriented levels.

On the other hand, the subset-oriented association emphasizes comparisons under different data slices or filtering conditions. For example, when analyzing "Cellular Subscription" trends in Asia, they can be compared with the changing patterns in other markets like Europe and Africa. These attributes share the same scale as the original concerned attribute and can thus be directly integrated into the corresponding annotation intervals of the chart (Fig. 8-B).

The process of generating correlation annotations consists of the following steps: First, for the identified temporal patterns, users can interactively select the association dimensions of interest. Next, the system automatically calculates the correlation coefficients between the selected association dimensions and the targeted attribute within that time window. Here, we adopt the commonly used Pearson correlation coefficient to quantify the correlation. Finally, the system presents the change curves of the associated attributes and uses colors to map the degree of correlation (green represents negative correlation, red represents positive correlation). In Fig. 8, users can choose all subset-oriented association attributes for a comprehensive comparison of different regions, or select only the key attribute "No. of Internet Users" from the attribute-oriented association candidates for targeted analysis.

Lag annotations aim to reveal the lead-lag relationships between different sequences, where changes in one sequence may lag behind or lead to another. Unlike correlation annotations that analyze the numerical correlations between variables within the same time interval, lag annotations focus on examining the dynamic dependencies across time. While these annotations do not directly imply causation, they can provide insights into potential causal relationships.

Similar to correlation annotations, lag annotations also consider both attribute-oriented and subset-oriented perspectives. Attribute-

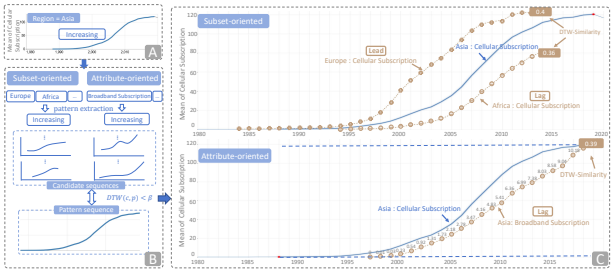

Figure 9: The illustration of lag annotations, which delineates the lead-lag relationships between attributes and subsets.

oriented lag analysis focuses on the dynamic associations between different attribute variables. As an example in Fig. 9-C, the growth of "Cellular Subscription" may precede that of "Broadband Subscription", which aligns with the historical process of technological development. However, this does not imply that the former directly caused the increase in the latter. On the other hand, subset-oriented lag analysis emphasizes comparing the temporal differences of the same attribute across different subsets. For instance, "Cellular Subscription" in developed regions (e.g., Europe) may lead those in less developed regions (e.g., Africa), reflecting disparities in the speed of technology adoption (Fig. 9-C). Similar to correlation annotations, lag annotations also encounter the challenge of inconsistent attribute scales. Here, we use the value range of the annotated pattern sequence on the attribute axis (i.e., y-axis) of the chart as a reference. The data of the associated attributes is then remapped to positions aligned with this reference range while maintaining their original positions on the time axis (i.e., the x-axis). Through this mapping, users can more intuitively observe and analyze the lead-lag relationships between attributes.

The primary goal of lag annotation is to identify segments from associated attributes that exhibit similar data patterns to the focused pattern but with temporal differences. As shown in Fig. 9-B, the process begins by extracting temporal patterns from the associated attributes and selecting sequences of the same type as the analyzed pattern for the candidate set. This preprocessing step can reduce the scale of subsequent matching computations, enhancing overall efficiency. DTW is then employed to quantify the similarity between these candidate sequences and the annotated pattern sequence, after normalizing both sequences to a 0-1 scale. DTW is chosen because it calculates similarity by finding the optimal nonlinear mapping between two TS, making it capable of handling pattern-matching scenarios with temporal distortion or displacement that are difficult for ED and other metrics. After obtaining the DTW distances, we select sequences with distances below a preset threshold as lag annotation objects (currently, we set the distance threshold to 1 but allow for dynamic adjustment). Finally, the selected lag sequences are integrated into the original chart, and the corresponding attribute tags and similarity values are labeled.

6 AutoMA System

To facilitate the demonstration and evaluation of our proposed multi-level contextual annotation approach, we developed AutoMA (Fig. 10), a visual analytics prototype system that integrates multi-level annotation into the analytical workflow of time-series data. The system's interface is designed to be concise and intuitive, consisting of three main components: data summary view, chart editing view, and chart annotation view. The data summary view (Fig. 10-B) employs a default four-layer horizon graph, providing an overall overview of the MTSD data. Horizon Graph is a compact time-series visualization technique that can display multiple temporal trends within a limited space. The chart editing view (Fig. 10-C) includes a parameter panel and a chart display area. Users can select the dimensions of interest from the MTSD data for in-depth analysis through the parameter panel and adjust the data analysis range using

filters. The corresponding results are presented in real-time in the chart display area. In the chart annotation view (Fig. 10-D), users can interactively add and view automatically generated multi-level annotations. We adopt an in-place annotation mode, directly integrating all annotation information with chart elements rather than creating a new view for each annotation. This design avoids frequent switching of user focus in multi-view analysis, reduces distraction, and enhances user concentration.

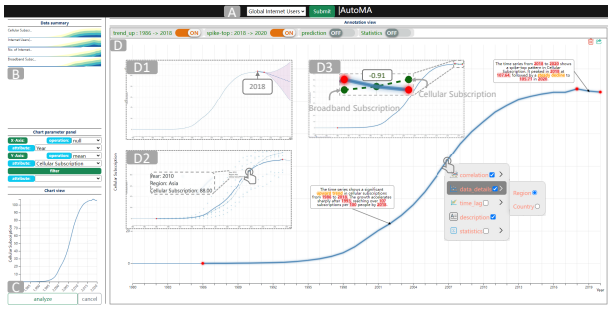

Figure 10: The user interface of AutoMA, which consists of three main components: data summary view (B), chart editing view (C), and chart annotation view (D).

Although multi-level annotations provide richer contextual information for charts, excessive annotations may lead to visual clutter. To address this, we allow users to interactively select the annotation content of interest. When the annotation content remains excessive, users can split it into multiple sub-annotation charts and export them (Fig. 10-D1, D2, and D3), ensuring the clarity and readability of each annotation chart. Additionally, variations in numerical ranges among annotation data can present display challenges. Specifically, adding annotations with extremely small or large values while maintaining the original scale may cause some data points to render improperly due to their values falling outside the scale range. Conversely, removing such annotations might result in substantial blank spaces, as the majority of the data points were previously compressed into a narrow interval and have not been restored to their original positions. To resolve this problem, we implement an automatic scaling feature to adapt to changes in the data range. When the annotation content changes, the system automatically tracks the value range and dynamically adjusts the data mapping scale, allowing the annotated data to be remapped to an appropriate range on the x-axis and y-axis. This approach avoids excessive blank space on the screen, enhancing the information delivery efficiency of the visualization view.

7 Evaluation

To comprehensively evaluate the effectiveness of AutoMA, we established several evaluation dimensions based on previous guidelines for visualization annotation assessment [19] and aligned with the research focus of this study. First, through comparative experiments, we tested the accuracy of the proposed LLM-based temporal pattern extraction method, as its performance is one of the key factors influencing the quality of generated annotations. Second, we carried out a controlled user study that involved quantitative task experiments to evaluate the impact of multi-level contextual annotations on users' data understanding and analysis processes, and user ratings to further supplement and refine the overall evaluation of AutoMA. In addition, we also present a usage scenario to demonstrate the effectiveness of AutoMA in *Appendix*-Section 1.

7.1 Temporal pattern extraction experiments

Our objective in this set of experiments was to determine whether LLMs are superior, equivalent, or inferior to existing machine learning models in the task of temporal pattern extraction. To this end, we selected a set of machine learning models as baselines to compare their accuracy on this task.

Experimental Setup. In this study, we formulate the temporal pattern recognition problem as a temporal classification task, aiming to predict the pattern type of each element in a TS. In AutoMA, we adopt the fine-tuning approach using the Qwen-7B [6] model open-sourced by Alibaba Group. We compare the proposed method with three basic machine learning models (LSTM, TCN, and hybrid TCN-LSTM) and three state-of-the-art time series models (CrossFormer [62], PatchTST [43], and TimesNet [60]) to validate its effectiveness. For the detailed settings of these models, please refer to *Appendix*-Section 3. The model performance is evaluated based on the accuracy metric, which measures the proportion of correctly classified elements in the time series. The experiments are first conducted on datasets of varying scales to analyze the impact of data size on model accuracy. Subsequently, the generalization ability of the models is evaluated on a test set that simulates real-world scenarios. All datasets for the above two tests are also introduced in *Appendix*-Section 3.

Model \ Dataset	5K	10K	25K	50K
LSTM	74.04%	79.70%	85.14%	86.57%
TCN	83.07%	85.17%	86.89%	87.58%
TCN-LSTM	80.69%	83.82%	88.64%	90.67%
CrossFormer [62]	78.28%	82.67%	88.03%	91.71%
PatchTST [43]	88.25%	91.24%	93.34%	94.27%
TimesNet [60]	88.29%	91.52%	94.82%	95.89%
Fine-tuned LLM	89.32%	94.25%	95.47%	96.15%

Table 1: Accuracy of different models across datasets of various sizes.

Results. The experimental results from Table 1 demonstrate that the LLM model outperforms traditional machine learning models in terms of accuracy across datasets of varying sizes. Even when compared to several state-of-the-art deep learning models, the LLM still exhibits advantageous performance. Furthermore, the LLM-based approach consistently delivers favorable results relative to baseline models under different real-world simulated conditions (more details of the results can be found in *Appendix*-Section 4). Further examination of the error cases in the model outputs indicates that the primary source of errors in the LLM method lies in the failure to locate the two ends of patterns, and this type of error is often accompanied by offset phenomena. In summary, the current experimental results provide preliminary validation of the effectiveness and robustness of LLMs in temporal pattern recognition tasks.

7.2 User study

This user study aims to investigate the impact of different chart annotation methods on users' understanding of chart content and their engagement with the analytical system, and to enhance a comprehensive evaluation of the AutoMA system through multi-indicator ratings. To this end, we first conducted a within-subjects study comparing the AutoMA tool against a baseline system containing only descriptive insights (similar to systems using auto-caption [34, 39]). The baseline system leveraged LLMs (GPT-4) to generate complete textual descriptions for the entire chart (the baseline system interface can be found in the supplementary material). We considered the following two hypotheses:

- **H1**: Users will record more findings when using AutoMA compared to the baseline system with only descriptive annotations.
- **H2**: Users will prefer the annotations provided by AutoMA over the baseline with only descriptive annotations.

Participants. We recruited a total of 12 participants (3 females and 9 males, age: $\mu = 23.42$ years, $\sigma = 1.56$ years) for this evaluation. They were all graduate students, with four newcomers and others from higher academic years. They were familiar with and had used common data analysis tools (e.g., ggplot and Matplotlib). They came from various research fields (including data visualization, image understanding, and computer network analysis). For clarity, we numbered these participants from P1 to P12.

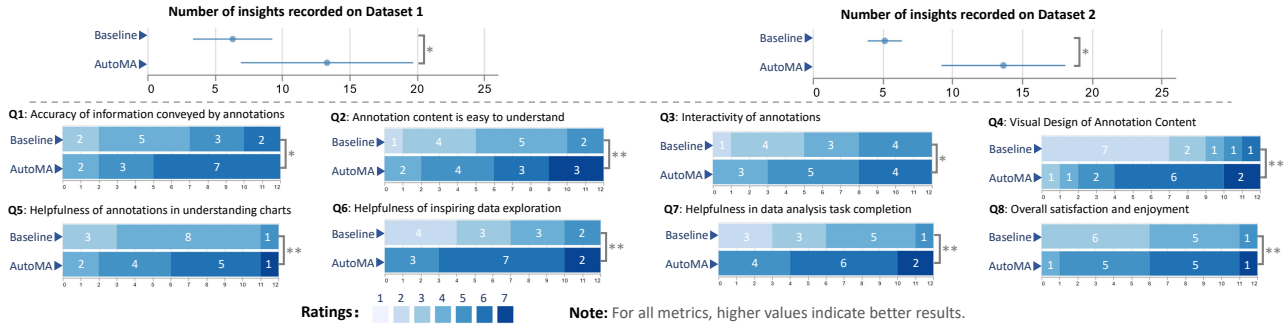

Figure 11: User study evaluation results. The top part presents the number of insights recorded by participants using different annotation systems on various datasets, with error bars representing 95% CI; the bottom part shows participants' ratings of the two systems, * indicates $p < .05$ and ** denotes $p < .001$.

Datasets. We selected two datasets in total: Dataset 1 is the global internet users dataset described in Sect. 3, and Dataset 2 is another dataset about global population migration. The global internet users dataset contains 7 attributes and 7,825 records, while the global population migration dataset has 7 attributes and 10,325 records. These two datasets are comparable in scale, which helps avoid cognitive bias that might arise from participants switching between data scenarios with drastically different dataset sizes. Meanwhile, they reflect different themes, facilitating the evaluation of AutoMA's performance across various contexts.

Design & Procedure. The entire user study lasted an average of 45 minutes, all conducted offline face-to-face. We first spent about 15 minutes training the participants to familiarize them with the AutoMA system's features and inform them of the study design. All participants were evenly divided into two groups, A and B. Group A members used AutoMA to analyze Dataset 1 and then used the baseline to analyze Dataset 2. Group B did the opposite of Group A, using the baseline to analyze Dataset 1 first and then AutoMA for Dataset 2. This way, each user ended up analyzing data using both comparison systems. We asked users to record their interesting findings during exploration as an evaluation metric. These findings/insights refer to meaningful pattern information (e.g., trends, anomalies) that users observe from the system while analyzing time-series data, as well as their further thoughts, interpretations, and inferences based on these observations. Although the quantity of recorded discoveries is commonly employed as a metric for evaluating general-purpose visual analysis tools, it also serves as a critical indicator for assessing the effectiveness of annotations and has been utilized in previous annotation evaluation studies [33, 46]. We allowed participants to freely copy, paste, or edit the textual descriptions provided by the system and encouraged them to explore the data thoroughly. After completing the above tasks, participants were asked to evaluate the two systems with different annotations using a 7-point Likert scale. We also recorded the feedback participants raised throughout the study.

Results. To evaluate **H1**, we compared the number of findings/insights documented by participants when exploring the two different datasets using AutoMA and the baseline system (Fig. 11-top). Following data collection, we conducted a manual inspection to identify and eliminate low-quality insights. These low-quality insights consisted of overly vague or generic statements that lacked specificity and depth. The distinguishing characteristics of these insights were their excessive brevity and failure to provide substantive information or make meaningful contributions to the analysis. Then, the Mann-Whitney U test was used to analyze the difference in the number of insights on each dataset between the two systems. The statistical analysis results showed that the number of insights recorded by participants was significantly higher when using AutoMA than the baseline system (Dataset 1: U = 2.5, $p < .05$; Dataset 2: U = 0.5, $p < .01$). This result is not surprising, as AutoMA provides more perspectives on data understanding through automatic multi-level annotations, thus promoting insight generation. This is also reflected in user feedback, with P4 stating that *"(translated) the hierarchical structure of the annotations in the multi-level Annotations not only facilitates the understanding of the current chart information but also provides connections to other charts and attributes"*. Moreover, we observed that the number of insights on Dataset 1 (AutoMA: $\mu = 13.33$, $\sigma = 4.13$; Baseline: $\mu = 6.33$, $\sigma = 1.97$) was slightly higher than on Dataset 2 (AutoMA: $\mu = 13.67$, $\sigma = 4.32$; Baseline: $\mu = 5.17$, $\sigma = 1.60$). This trend may stem from inherent differences in the information content of the datasets, with Dataset 1 potentially containing richer pattern information, providing more opportunities for insight discovery. Combining the above findings, hypothesis **H1** is supported.

To evaluate **H2**, we collected participants' ratings on eight indicators (Fig. 11-bottom), including annotation implementation and its helpfulness for chart understanding (**Q1**-**Q5**), facilitation of exploration and task completion (**Q6**-**Q7**), and overall satisfaction (**Q8**). The Mann-Whitney U test was used to analyze the difference in scores between the two systems on each indicator. The statistical analysis results showed that AutoMA performed significantly better than the baseline system on all indicators ($p < .05$). Specifically, participants found the annotation content generated by AutoMA to be richer (U = 111.5, $p = .05$), easier to understand (U = 131.0, $p < .001$), more interactive (U = 117.5, $p = .05$), visually more expressive (U = 131.5, $p < .001$), and more helpful for overall chart understanding (U = 132.0, $p < .001$). P1 remarked, *"(translated) the colorful visual annotation style (AutoMA) is more effective in capturing my attention and leaving a deeper impression on my memory"*. Furthermore, participants also felt that AutoMA better facilitated data exploration (U = 141.0, $p < .001$) and task completion (U = 142.0, $p < .001$), and they were more satisfied with the overall experience of conducting analysis using AutoMA (U = 138.0, $p < .001$). For instance, P7 commented, *"(translated) I appreciate the in-situ annotation approach in AutoMA, as it prevents me from having to switch between different views (baseline)"*. Integrating the above findings, our evaluation results support hypothesis **H2**, indicating that compared to the baseline system, AutoMA can significantly enhance users' subjective experience with the visual analytics system.

8 Discussion

In this section, we review the design and implementation of AutoMA, discuss the current limitations, and propose directions for improvements.

Enhanced visualization pattern extraction. While our method has been validated on synthetic data, applying it to real-world datasets may introduce additional complexities. These complexities primarily arise from unforeseen patterns and noise in real data, which are difficult to fully simulate in synthetic data. One direction for improvement is to enhance the diversity of synthetic data. This can be achieved by introducing more sophisticated generative models and incorporating domain knowledge to simulate more diverse and complex real-world data distributions. On the other hand, the constantly changing nature of real-world environments necessitates models with continuous learning and adaptation capabilities.

Techniques such as incremental learning can be utilized to enable models to update promptly as new data becomes available, thereby allowing them to adapt to emerging patterns and changes effectively. Furthermore, our current focus is on annotating single-attribute time-series data visualizations. However, our proposed design space for multilevel annotations has the potential to be extended to a wider range of multidimensional data, chart types, and visualization tasks. Exploring these extensions could lead to a more comprehensive and versatile annotation system for data visualization. Moreover, although machine learning-based approaches for time-series pattern detection offer generalizability, the model outputs often face challenges in terms of uncertainty and interpretability. Our user study also revealed that incorrect pattern information can easily confuse users. To mitigate this issue, P3 and P5 suggested that we could consider providing explanations and correction mechanisms for erroneous facts to help readers better understand the chart content.

Richer and better-presented annotations. Regarding the context of annotations, our current three-level annotation design space does not integrate additional external information (e.g., human knowledge and textual resources). This makes AutoMA only describe visual content within charts and contextual relationships between attributes, but lacks explanations of some phenomena or emphasis on specific patterns. In the future, integrating these external sources of information to construct a four-level annotation space could be considered. Additionally, while AutoMA offers richer and more diverse annotation types, enhancing the contextuality of information, it also introduces challenges in information presentation. As the number of data dimensions grows, the scale of generated annotations also increases. Presenting excessive information within a limited screen space can hinder effective display. To address this, one potential optimization strategy suggested by P9 is to set priority scores for annotations and dynamically optimize the layout based on the importance and relevance of annotations, thereby maximizing information readability and overall chart aesthetics. Lastly, in real-world scenarios, users may have pre-created visualizations. Future work could extend AutoMA to directly process these existing visualization images and employ augmented reality techniques to add virtual annotation layers, providing a seamless and immersive annotation experience.

9 Conclusion

In this work, we introduced AutoMA, a system that leverages LLMs to enhance the generation of rich, contextual annotations for multidimensional time series visualizations. By fine-tuning an LLM on a synthetically generated dataset, AutoMA effectively identifies diverse temporal patterns, extending beyond basic visual trends. Our proposed multi-level annotation design space integrates diverse sources of contextual information, enabling the system to support six distinct annotation tasks and provide users with a more comprehensive understanding of the data. The system's flexibility in annotation interaction and its auto-scaling feature further enhance the user experience. Our evaluation, through experiments and user studies, has demonstrated the effectiveness of AutoMA in improving users' ability to interpret and utilize multidimensional time series visualizations. The positive feedback also underscores the potential of LLMs to transform temporal pattern recognition and visualization annotation. Finally, we also discuss the current limitations of our work and explore future directions for enhancement.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (62422607, 62372411, 62036009, 62432014), and Zhejiang Provincial Natural Science Foundation of China (LR23F020003, LTGG23F020005).

References

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*, volume 4. 2011.

[2] S. Alnegheimish, L. Nguyen, L. Berti-Equille, and K. Veeramachaneni. Can large language models be anomaly detectors for time series? In *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, 2024.

[3] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization*, pages 111–117, 2005.

[4] S. Aminikhanghahi and D. J. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2):339–367, 2017.

[5] N. Andrienko, G. Andrienko, S. Miksch, H. Schumann, and S. Wrobel. A theoretical model for pattern discovery in visual analytics. *Visual Informatics*, 5(1):23–42, 2021.

[6] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al. Qwen technical report. *ArXiv*, abs/2309.16609, 2023.

[7] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *ArXiv*, abs/1803.01271, 2018.

[8] F. Bajić, M. Habijan, and K. Nenadić. A synthetic dataset of different chart types for advancements in chart identification and visualization. *Data in Brief*, 53:110233, 2024.

[9] A. Bendeck, D. Bromley, and V. Setlur. Slopeseeker: A search tool for exploring a dataset of quantifiable trends. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, pages 817–836, 2024.

[10] D. BERNDT. Using dynamic time warping to find patterns in time series. In *Proceedings of AAAI Workshop on Knowledge Discovery in Databases, 1994*, pages 359–370, 1994.

[11] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What makes a visualization memorable? *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2306–2315, 2013.

[12] D. Bromley and V. Setlur. What is the difference between a mountain and a molehill? quantifying semantic labeling of visual features in line charts. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pages 161–165. IEEE, 2023.

[13] C. Bryan, K.-L. Ma, and J. Woodring. Temporal summary images: An approach to narrative visualization via interactive annotation generation and placement. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):511–520, 2016.

[14] T. N. Bulkowski. *Encyclopedia of chart patterns*. John Wiley & Sons, 2021.

[15] A. Chaddad, Y. Wu, R. Kateb, and A. Bouridane. Electroencephalography signal processing: A comprehensive review and analysis of methods and techniques. *Sensors*, 23(14):6434, 2023.

[16] C.-H. Chen, V. S. Tseng, H.-H. Yu, and T.-P. Hong. Time series pattern discovery by a pip-based evolutionary approach. *Soft Computing*, 17:1699–1710, 2013.

[17] Y. Chen, S. Barlowe, and J. Yang. Click2annotate: Automated insight externalization with rich semantics. In *2010 IEEE Symposium on Visual Analytics Science and Technology*, pages 155–162, 2010.

[18] V. Dibia and Ç. Demiralp. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE Computer Graphics and Applications*, 39(5):33–46, 2019.

[19] M. Dilshadur Rahman, B. Doppalapudi, G. Jilani Quadri, and P. Rosen. A survey on annotations in information visualization: Empirical insights, applications, and challenges. *arXiv*, 2024.

[20] R. Ding, S. Han, Y. Xu, H. Zhang, and D. Zhang. Quickinsights: Quick and automatic discovery of insights from multi-dimensional data. In *Proceedings of the 2019 International Conference on Management of Data*, pages 317–332, 2019.

[21] M. A. Farahani, M. McCormick, R. Gianinny, F. Hudacheck, R. Harik, Z. Liu, and T. Wuest. Time-series pattern recognition in smart manufacturing systems: A literature review and ontology. *Journal of Manufacturing Systems*, 69:208–241, 2023.

[22] S. Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, pages 37–45, 1997.

[23] J. Hullman, N. Diakopoulos, and E. Adar. Contextifier: Automatic generation of annotated stock visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2707–2716, 2013.

[24] R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3):1–22,

2008.
[25] T. Ibanez, V. Setlur, and M. Agrawala. ALMANAC: An api for recommending text annotations for time-series charts using news headlines. *ArXiv*, 2023.
[26] G. Iglesias, E. Talavera, Á. González-Prieto, A. Mozo, and S. Gómez-Canaval. Data augmentation techniques in time series domain: a survey and taxonomy. *Neural Computing and Applications*, 35(14):10123–10145, 2023.
[27] Q. Jiang, G. Sun, Y. Dong, and R. Liang. DT2VIS: A focus+ context answer generation system to facilitate visual exploration of tabular data. *IEEE Computer Graphics and Applications*, 41(5):45–56, 2021.
[28] Q. Jiang, G. Sun, Y. Dong, L. Pan, B. Chang, L. Jiang, H. Liang, and R. Liang. Factexplorer: Fact embedding-based exploratory data analysis for tabular data. *International Journal of Human–Computer Interaction (Early Access)*, pages 1–19, 2025.
[29] Q. Jiang, G. Sun, T. Li, J. Tang, W. Xia, S. Zhu, and R. Liang. Qutaber: task-based exploratory data analysis with enriched context awareness. *Journal of Visualization*, 27(3):503–520, 2024.
[30] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. In *Data mining in time series databases*, pages 1–21. 2004.
[31] N. Kong and M. Agrawala. Graphical overlays: Using layered elements to aid chart reading. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2631–2638, 2012.
[32] C. Lai, Z. Lin, R. Jiang, Y. Han, C. Liu, and X. Yuan. Automatic annotation synchronizing with textual description for visualization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.
[33] Y. Lin, H. Li, L. Yang, A. Wu, and H. Qu. Inksight: Leveraging sketch interaction for documenting chart findings in computational notebooks. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):944–954, 2024.
[34] C. Liu, L. Xie, Y. Han, D. Wei, and X. Yuan. Autocaption: An approach to generate natural language description from visualization automatically. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pages 191–195, 2020.
[35] P. Liu, X. Sun, Y. Han, Z. He, W. Zhang, and C. Wu. Arrhythmia classification of LSTM autoencoder based on time series anomaly detection. *Biomedical Signal Processing and Control*, 71:103228, 2022.
[36] S. Liu, Y. Tian, Z. Deng, W. Cui, H. Zhang, D. Weng, and Y. Wu. Relation-driven query of multiple time series. *IEEE Transactions on Visualization and Computer Graphics (Early Access)*, 2024.
[37] M. Lovrić, M. Milanović, and M. Stamenković. Algoritmic methods for segmentation of time series: An overview. *Journal of Contemporary Economic and Business Issues*, 1(1):31–53, 2014.
[38] P. Ma, R. Ding, S. Han, and D. Zhang. Metainsight: Automatic discovery of structured knowledge for exploratory data analysis. In *Proceedings of the 2021 International Conference on Management of Data*, pages 1262–1274, 2021.
[39] A. Mahinpei, Z. Kostic, and C. Tanner. Linecap: Line charts for data visualization captioning models. In *2022 IEEE Visualization and Visual Analytics (VIS)*, pages 35–39, 2022.
[40] E. Merdjanovska and A. Rashkovska. Comprehensive survey of computational ecg analysis: Databases, methods and applications. *Expert Systems with Applications*, 203:117206, 2022.
[41] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.
[42] J. A. Miller, M. Aldosari, F. Saeed, N. H. Barna, S. Rana, I. B. Arpinar, and N. Liu. A survey of deep learning and foundation models for time series forecasting. *ArXiv*, 2024.
[43] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.
[44] M. D. Rahman, G. J. Quadri, B. Doppalapudi, D. A. Szafir, and P. Rosen. A qualitative analysis of common practices in annotations: A taxonomy and design space. *IEEE Transactions on Visualization and Computer Graphics (Early Access)*, pages 1–11, 2024.
[45] D. Ren, M. Brehmer, B. Lee, T. Höllerer, and E. K. Choe. Chartaccent: Annotation for data-driven storytelling. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 230–239, 2017.
[46] M. M. U. Rony, F. Du, R. Rossi, J. Hoffswell, N. Chhaya, I. Burhanuddin, and E. Koh. Augmenting visualizations with predictive and investigative insights to facilitate decision making. In *Companion Proceedings of the ACM Web Conference 2023*, page 77–81, 2023.
[47] K. Seweryn, L. Katarzyna, A. Wróblewska, and S. Sysko-Romańczuk. What will you tell me about the chart?–automated description of charts. In *28th International Conference on Neural Information Processing*, 2021.
[48] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao. Calliope: Automatic visual data story generation from a spreadsheet. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):453–463, 2020.
[49] M. Shin, J. Kim, Y. Han, L. Xie, M. Whitelaw, B. C. Kwon, S. Ko, and N. Elmqvist. Roslingifier: Semi-automated storytelling for animated scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 29(6):2980–2995, 2022.
[50] G. Shirato, N. Andrienko, and G. Andrienko. Identifying, exploring, and interpreting time series shapes in multivariate time intervals. *Visual Informatics*, 7(1):77–91, 2023.
[51] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):672–681, 2018.
[52] C. Stokes, C. X. Bearfield, and M. A. Hearst. The role of text in visualizations: How annotations shape perceptions of bias and influence predictions. *IEEE Transactions on Visualization and Computer Graphics*, 30(10):6787–6800, 2023.
[53] M. Tan, M. A. Merrill, V. Gupta, T. Althoff, and T. Hartvigsen. Are language models actually useful for time series forecasting? *CoRR*, abs/2406.16964, 2024.
[54] C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.
[55] A. Ulmer, M. Angelini, J.-D. Fekete, J. Kohlhammer, and T. May. A survey on progressive visualization. *IEEE Transactions on Visualization and Computer Graphics*, 30(9):6447–6467, 2024.
[56] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. Datashot: Automatic generation of fact sheets from tabular data. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):895–905, 2019.
[57] A. S. Weigend. *Time series prediction: forecasting the future and understanding the past*. Routledge, 2018.
[58] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016.
[59] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 2648–2659, 2017.
[60] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.
[61] E. Zgraggen, A. Galakatos, A. Crotty, J.-D. Fekete, and T. Kraska. How progressive visualizations affect exploratory analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(8):1977–1987, 2017.
[62] Y. Zhang and J. Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.
[63] Y. Zhao, J. Ren, B. Zhang, J. Wu, and Y. Lyu. An explainable attention-based TCN heartbeats classification model for arrhythmia detection. *Biomedical Signal Processing and Control*, 80:104337, 2023.
[64] Y. Zhou, X. Meng, Y. Wu, T. Tang, Y. Wang, and Y. Wu. An intelligent approach to automatically discovering visual insights. *Journal of Visualization*, 26(3):705–722, 2023.
[65] J. Zhu, J. Ran, R. K.-W. Lee, Z. Li, and K. Choo. Autochart: A dataset for chart-to-text generation task. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 1636–1644, 2021.
[66] S. Zhu, G. Sun, Q. Jiang, M. Zha, and R. Liang. A survey on automatic infographics and visualization recommendations. *Visual Informatics*, 4(3):24–40, 2020.