Large Language Models for Time Series: A Survey

Xiyuan Zhang, **Ranak Roy Chowdhury**, **Rajesh K. Gupta** and **Jingbo Shang**
University of California, San Diego
{xiyuanzh, rrchowdh, rgupta, jshang}@ucsd.edu

Abstract

Large Language Models (LLMs) have seen significant use in domains such as natural language processing and computer vision. Going beyond text, image and graphics, LLMs present a significant potential for analysis of time series data, benefiting domains such as climate, IoT, healthcare, traffic, audio and finance. This survey paper provides an in-depth exploration and a detailed taxonomy of the various methodologies employed to harness the power of LLMs for time series analysis. We address the inherent challenge of bridging the gap between LLMs' original text data training and the numerical nature of time series data, and explore strategies for transferring and distilling knowledge from LLMs to numerical time series analysis. We detail various methodologies, including (1) direct prompting of LLMs, (2) time series quantization, (3) aligning techniques, (4) utilization of the vision modality as a bridging mechanism, and (5) the combination of LLMs with tools. Additionally, this survey offers a comprehensive overview of the existing multimodal time series and text datasets in diverse domains, and discusses the challenges and future opportunities of this emerging field.

1 Introduction

Time series analysis plays a critical role in a variety of fields, including climate modeling, traffic management, healthcare monitoring and finance analytics. Time series analysis comprises a wide range of tasks such as classification [Liu *et al.*, 2023b], forecasting [Gruver *et al.*, 2023], anomaly detection, and imputation. Traditionally, these tasks have been tackled using classical signal processing techniques such as time-frequency analysis and decomposition-based approaches. More recently, deep learning approaches like Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs) [Zhang *et al.*, 2023a], and Transformers [Jin *et al.*, 2023a] have revolutionized this field and proved effective in extracting meaningful patterns from time series data, making them the primary approaches of time series analysis in various application domains.

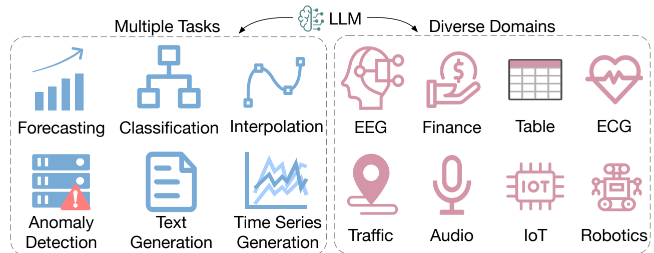


Figure 1: Large language models have recently been applied for various time series tasks in diverse application domains.

In recent years, Large Language Models (LLMs) have gained substantial attention particularly in the fields of Natural Language Processing (NLP) and Computer Vision (CV). Prominent models such as GPT-4 have transformed the landscape of text processing by offering unprecedented accuracy in tasks such as text generation, translation, sentiment analysis, question answering and summarization. In the CV domain, Large Multimodal Models (LMMs) have also facilitated advancements in image recognition, object detection, and generative tasks, leading to more intelligent and capable visual systems [Girdhar *et al.*, 2023]. Inspired by these successes, researchers are now exploring the potential of LLMs in the realm of time series analysis, expecting further breakthroughs, as shown in Figure 1. While several surveys offer a broad perspective on large models for time series in general [Jin *et al.*, 2023b; Ma *et al.*, 2023], these do not specifically focus on LLMs or the key challenge of bridging modality gap, which stems from LLMs being originally trained on discrete textual data, in contrast to the continuous numerical nature of time series.

Our survey uniquely contributes to the existing literature by emphasizing how to bridge such modality gap and transfer knowledge from LLMs for time series analysis. Our survey also covers more diverse application domains, ranging from climate, Internet of Things (IoT), to healthcare, traffic management, and finance. Moreover, certain intrinsic properties of time series, like continuity, auto-regressiveness, and dependency on the sampling rate, are also shared by audio, speech, and music data. Therefore, we also present representative LLM-based works from these domains to explore how we can use LLMs for other types of time series. We present

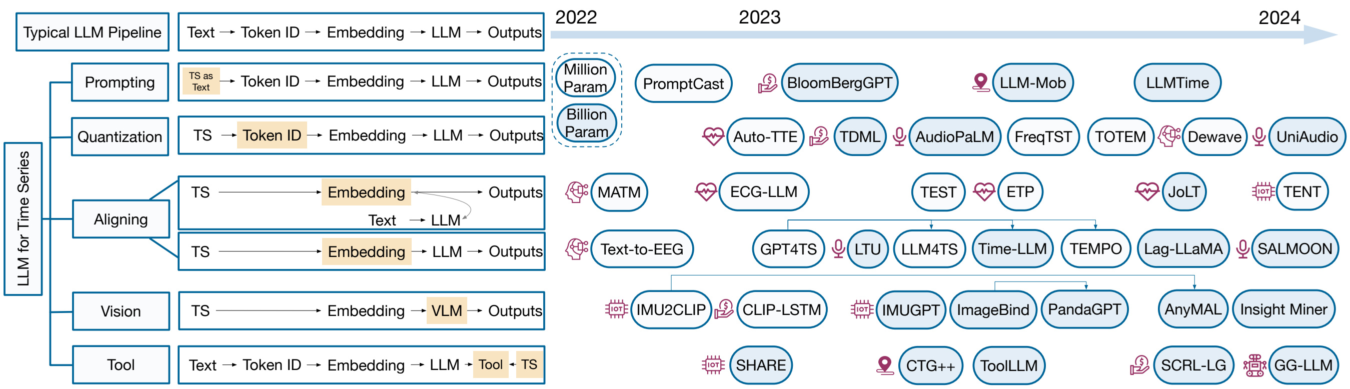


Figure 2: Left: Taxonomy of LLMs for time series analysis (prompting, quantization, aligning which is further categorized into two groups as detailed in Figure 4, vision as bridge, tool integration). For each category, key distinctions are drawn in comparison to the standard LLM pipeline shown at the top of the figure. Right: We present representative works for each category, sorted by their publication dates. The use of arrows indicates that later works build upon earlier studies. Dark(light)-colored boxes represent billion(million)-parameter models. Icons to the left of the text boxes represent the application domains of domain-specific models, with icons' meanings illustrated in Figure 1.

a comprehensive taxonomy by categorizing these methodologies into five distinct groups, as shown in Figure 2. If we outline typical LLM-driven NLP pipelines in five stages - input text, tokenization, embedding, LLM, output - then each category of our taxonomy targets one specific stage in this pipeline. Specifically, (i) *Prompting* (input stage) treats time series data as raw text and directly prompts LLMs with time series; (ii) *Time Series Quantization* (tokenization stage) discretizes time series as special tokens for LLMs to process; (iii) *Aligning* (embedding stage) designs time series encoder to align time series embeddings with language space; (iv) *Vision as Bridge* (LLM stage) connects time series with Vision-Language Models (VLM) by employing visual representations as a bridge; (v) *Tool Integration* (output stage) adopts LLMs to output tools to benefit time series analysis. Beyond this taxonomy, our survey also compiles an extensive list of existing multimodal datasets that incorporate both time series and text. We conclude our paper by discussing future research directions in this emerging and promising field.

We maintain an up-to-date Github repository[1] which includes all the papers and datasets discussed in the survey.

2 Background and Problem Formulation

Large language models are characterized by their vast number of parameters and extensive training data. They excel in understanding, generating, and interpreting human language, and recently represent a significant advancement in artificial intelligence. The inception of LLMs can be traced back to models like GPT-2, BERT, BART, and T5, which laid the foundational architecture. Over time, the evolution of these models has been marked by increasing complexity and capabilities, such as LLAMA-2, PaLM, and GPT-4. More recently, researchers have developed multimodal large language models to integrate and interpret multiple forms of data, such as text, images, and time series, to achieve a more comprehensive understanding of information.

[1]https://github.com/xiyuanzh/awesome-llm-time-series

This survey focuses on how LLMs could benefit time series analysis. We first define the mathematical formulation for the input and output, which may contain time series or (and) text depending on the downstream tasks, as well as the models.

Input. Denoted as $\mathbf{x}$, composed of time series $\mathbf{x}_s \in \mathbb{R}^{T\times c}$ and optional text data $\mathbf{x}_t$ represented as strings, where T, c represent the sequence length and the number of features.

Output. Denoted as $\mathbf{y}$ and may represent time series, text or numbers depending on the specific downstream task. For time series generation or forecasting task, $\mathbf{y}$ represents generated time series $\mathbf{y}_s$ or predicted k-step future time series $\mathbf{y}_s^{T+1:T+k}$. For text generation task, such as report generation, $\mathbf{y}$ represents text data $\mathbf{y}_t$. For time series classification or regression task, $\mathbf{y}$ represents numbers indicating the predicted classes or numerical values.

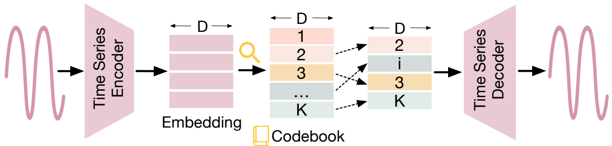
Model. We use f_θ parameterized by θ, g_ϕ parameterized by ϕ, and h_ψ parameterized by ψ to represent language, time series and vision models, where f_θ is typically initialized from pre-trained large language models. We optimize parameters θ, ϕ and ψ through loss function $\mathcal{L}$.

3 Taxonomy

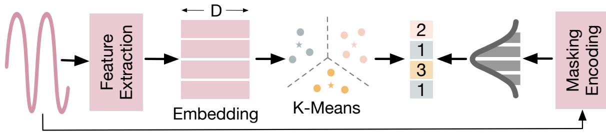
In this section, we detail our taxonomy of applying LLMs for time series analysis, categorized by five groups. We summarize the representative works, mathematical formulation, advantages and limitations of each category in Table 1.

3.1 Prompting

Number-Agnostic Tokenization. The method treats numerical time series as raw textual data and directly prompts existing LLMs. For example, PromptCast [Xue and Salim, 2022] proposes prompt-based time series forecasting by converting numerical time series into text prompts and forecasting time series in a sentence-to-sentence manner. The input prompts are composed of context and questions following pre-defined templates, e.g., "From $\{t_1\}$ to $\{t_{\text{obs}}\}$, the average temperature of region $\{U_m\}$ was $\{x_t^m\}$ degree on each



(a) VQ-VAE based quantization method.



(b) K-Means based quantization method.

Figure 3: Two types of index-based quantization methods.

day. What is the temperature going to be on $\{t_{\text{obs}}\}$?" Similar prompting methods have been applied to forecast Place-of-Interest (POI) customer flows (AuxMobLCast) and user's next location (LLM-Mob). Recent works also prompt PaLM-24B for health-related tasks such as activity recognition and daily stress estimate [Liu *et al.*, 2023b]. For example, they prompt the model to "classify the following accelerometer data in meters per second squared as either walking or running: 0.052, 0.052, 0.052, 0.051, 0.052, 0.055, 0.051, 0.056, 0.06, 0.064". Other examples include extracting historical price features such as open, close, high, and low prices to prompt ChatGPT in a zero-shot fashion [Xie *et al.*, 2023a].

Number-Specific Tokenization. More recently, LLM-Time [Gruver *et al.*, 2023] pointed out that Byte Pair Encoding (BPE) tokenization has the limitation of breaking a single number into tokens that don't align with the digits, leading to inconsistent tokenization across different floating point numbers and complicating arithmetic operations. Therefore, following LLMs such as LLaMA and PaLM, they propose to insert spaces between digits to ensure distinct tokenization of each digit and use a comma (",") to separate each time step in a time series. They also scale time series to optimize token usage and keep fixed precision (e.g., two digits of precision) to efficiently manage context length. For example, they convert "0.123, 1.23, 12.3, 123.0" to "1 2 , 1 2 3 , 1 2 3 0 , 1 2 3 0 0". Meanwhile, BloomberGPT [Wu *et al.*, 2023] trains on financial data with text and numerical data and places each digit in its own chunk to better handle numbers. Using similar space-prefixed tokenization, recent works also show that large language models are general pattern machines capable of sequence transformation, completion and improvement.

3.2 Quantization

Quantization based method converts numerical data into discrete representations as input to LLMs. This approach can be further divided into two main categories based on the discretization technique employed.

Discrete Indices from VQ-VAE. The first type of quantization method transforms continuous time series into discrete indices as tokens. Among them one of the most popular methods is training a Vector Quantized-Variational AutoEncoder (VQ-VAE), which learns a codebook $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^{K}$ of K D-dimensional codewords $\mathbf{c}_i \in \mathbb{R}^D$ to capture the latent representations, as illustrated in Figure 3a. The method identifies the nearest neighbor k_i of each step i of the encoded time series representation $g_\phi(\mathbf{x}_s) \in \mathbb{R}^{\frac{T}{S}\times D}$ in the codebook (S denotes the cumulative stride of VQ-VAE encoder), and uses the corresponding indices $\mathbf{k}$ as the quantized input to language models:

$$\mathbf{q}_i = \mathbf{c}_{k_i}, k_i = \arg\min_j \|g_\phi(\mathbf{x}_s)_i - \mathbf{c}_j\|_2, \mathbf{k} = [k_i]_{i=1}^{\frac{T}{S}}. \quad (1)$$

Based on VQ-VAE, Auto-TTE [Chung *et al.*, 2023] quantizes ECGs into discrete formats and generates 12-lead ECG signals conditioned on text reports. DeWave [Duan *et al.*, 2023] adapts VQ-VAE to derive discrete codex encoding and aligns it with pre-trained BART for open-vocabulary EEG-to-text translation tasks. TOTEM [Talukder and Gkioxari, 2023] also quantizes time series through VQ-VAE as input to Transformers for multiple downstream applications such as forecasting, classification, and translation. In the audio domain, UniAudio [Yang *et al.*, 2023] tokenizes different types of target audio using Residual Vector Quantization (RVQ) (a hierarchy of multiple vector quantizers) and supports 11 audio generation tasks. VioLA unifies various crossmodal tasks involving speech and text by converting speech utterances to discrete tokens through RVQ. AudioGen learns discrete audio representations using vector quantization layers and generates audio samples conditioned on text inputs.

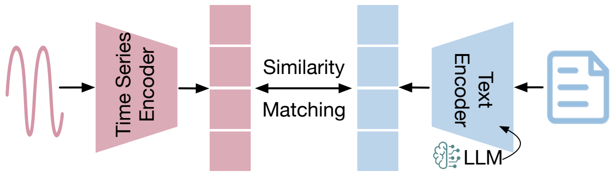
Discrete Indices from K-Means. Apart from employing VQ-VAE, researchers have also explored K-Means clustering for index-based tokenization, which uses the centroid indices as discretized tokens, as shown in Figure 3b. Such methods are mostly applied in the audio domain. For example, SpeechGPT shows capability to perceive and generate multi-modal contents using K-Means based discrete unit extractor. AudioLM discretizes codes produced by a neural audio codec using K-means clustering to achieve high-quality synthesis. It also combines discretized activations of language models pre-trained on audio using RVQ to capture long-term structure. Following the same quantization procedure, AudioPaLM [Rubenstein *et al.*, 2023] aligns PaLM-2 and AudioLM with a joint vocabulary that can represent speech and text with discrete tokens.

Discrete Indices from Other Techniques. Apart from the VQ-VAE and K-Means based time-domain quantization, FreqTST [Li *et al.*, 2023] utilizes frequency spectrum as a common dictionary to discretize time series into frequency units with weights for downstream forecasting task.

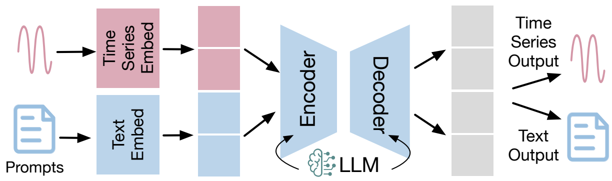
Text Categories. The second type of quantization converts numerical data into pre-defined text categories, which is primarily adopted in financial domain. For example, TDML [Yu *et al.*, 2023] categorizes the weekly price fluctuations into 12 bins represented as "Di" or "Ui", where "D" indicates a decrease in price and "U" means an increase, and $i = 1, 2, 3, 4, 5, 5+$ represents the level of price change.

3.3 Aligning

The third type of works trains a separate encoder for time series, and aligns the encoded time series to the semantic space



(a) Aligning by similarity matching (Type one).



(b) Aligning with large language models as backbones (Type two), where the output could be time series (e.g., forecasting) or text (e.g., EEG-to-text) depending on the downstream tasks.

Figure 4: Two types of aligning based methods.

of language models. These works can be further categorized into two groups based on their specific aligning strategies, as illustrated in Figure 4.

Similarity Matching through Contrastive Loss. The first type of method aligns the time series embeddings with text embeddings through similarity matching, such as minimizing the contrastive loss:

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp(\text{sim}(g_\phi(\mathbf{x}_{si}), f_\theta(\mathbf{x}_{ti})))^{\frac{1}{\gamma}}}{\sum_{k=1}^{B}\exp(\text{sim}(g_\phi(\mathbf{x}_{si}), f_\theta(\mathbf{x}_{tk})))^{\frac{1}{\gamma}}}, \tag{2}$$

where B, γ represent batch size and temperature parameter that controls distribution concentrations, and sim represents similarity score, typically computed as inner product:

$$\text{sim}(g_\phi(\mathbf{x}_{si}), f_\theta(\mathbf{x}_{ti})) = \langle g_\phi(\mathbf{x}_{si}), f_\theta(\mathbf{x}_{ti})\rangle. \tag{3}$$

For instance, ETP [Liu *et al.*, 2023a] integrates contrastive learning based pre-training to align electrocardiography (ECG) signals with textual reports. Contrastive framework is also used to align 17 clinical measurements collected in Intensive Care Unit (ICU) to their corresponding clinical notes [King *et al.*, 2023]. TEST [Sun *et al.*, 2023] uses contrastive learning to generate instance-wise, feature-wise, and text-prototype-aligned time series embeddings to align with text embeddings. TENT [Zhou *et al.*, 2023b] aligns text embeddings with IoT sensor signals through a unified semantic space using contrastive learning. JoLT [Cai *et al.*, 2023] utilizes Querying Transformer (Q-Former) optimized with contrastive loss to align time series and text representations.

Similarity Matching through Other Losses. Apart from contrastive loss, other loss functions are also employed to optimize similarity matching between time series embeddings and text embeddings. ECG-LLM [Qiu *et al.*, 2023] aligns the distribution between ECG and language embedding from ECG statements with an Optimal Transport based loss function to train an ECG report generation model. MTAM [Han *et al.*, 2022] uses various aligning techniques, such as Canonical Correlation Analysis and Wasserstein Distance, as loss functions to align electroencephalography (EEG) features with their corresponding language descriptions.

LLMs as Backbones. The second type of aligning method directly uses large language models as backbones following time series embedding layers. EEG-to-Text [Wang and Ji, 2022] feeds EEG embeddings to pre-trained BART for open vocabulary EEG-To-Text decoding and EEG-based sentiment classification. GPT4TS [Zhou *et al.*, 2023a] uses patching embeddings as input to frozen pre-trained GPT-2 where the positional embedding layers and self-attention blocks are retained during time series fine-tuning. The method provides a unified framework for seven time series tasks, including few-shot or zero-shot learning. Following GPT4TS, researchers further incorporated seasonal-trend decomposition (TEMPO [Cao *et al.*, 2023]), two-stage fine-tuning (LLM4TS [Chang *et al.*, 2023]), domain descriptions (UniTime), graph attention mechanism (GATGPT), and spatial-temporal embedding module (ST-LLM). Time-LLM [Jin *et al.*, 2023a] reprograms time series data into text prototypes as input to LLaMA-7B. It also provides natural language prompts such as domain expert knowledge and task instructions to augment input context. Lag-Llama builds univariate probabilistic time series forecasting model based on LLaMA architecture. In the audio, speech and music domains, researchers have also designed dedicated encoders to embed speech (WavPrompt, Speech LLaMA), music (MU-LLaMA), and general audio inputs (LTU [Gong *et al.*, 2023], SALMONN [Tang *et al.*, 2023]), and feed the embeddings to large language models.

3.4 Vision as Bridge

Time series data can be effectively interpreted or associated with visual representations, which align closer with textual data and have demonstrated successful integrations with large language models. Therefore, researchers have also leveraged vision modality as a bridge to connect time series with LLMs.

Paired Data. ImageBind [Girdhar *et al.*, 2023] uses image-paired data to bind six modalities (images, text, audio, depth, thermal, and Inertial Measurement Unit (IMU) time series) and learn a joint embedding space, enabling new emergent alignments and capabilities. PandaGPT [Su *et al.*, 2023] further combines the multimodal encoders from ImageBind and large language models to enable visual and auditory instruction-following capabilities. IMU2CLIP [Moon *et al.*, 2022] aligns IMU time series with video and text, by projecting them into the joint representation space of Contrastive Language-Image Pre-training (CLIP). AnyMAL [Moon *et al.*, 2023] builds upon IMU2CLIP by training a lightweight adapter to project the IMU embeddings into the text token embedding space of LLaMA-2-70B. It is also capable of transforming data from other modalities, such as images, videos, audio, into the same text embedding space.

Physics Relationships. IMUGPT [Leng *et al.*, 2023] generates IMU data from ChatGPT-augmented text descriptions. It first generates 3D human motion from text using pre-trained motion synthesis model, and derives IMU data from 3D motion based on physics relationships of motion kinetics.

Method	Subcategory	Representative Works	Equations	Advantages	Limitations
Prompting	Number-Agnostic	PromptCast [Xue and Salim, 2022]	$\mathbf{y} = f_\theta(\mathbf{x}_s, \mathbf{x}_t)$	easy to implement;	lose semantics;
	Number-Specific	LLMTime [Gruver *et al.*, 2023]		zero-shot capability	not efficient
Quantization	VQ-VAE	DeWave [Duan *et al.*, 2023]	$k_i = \arg\min_j \lVert g_\phi(\mathbf{x}_s)_i - \mathbf{c}_j \rVert_2$	flexibility of	may require
	K-Means	AudioPaLM [Rubenstein *et al.*, 2023]	$\mathbf{k} = [k_i]_{i=1}^{\frac{T}{S}}, \mathbf{y} = f_\theta(\mathbf{k}, \mathbf{x}_t)$	index and time	two-stage
	Text Categories	TDML [Yu *et al.*, 2023]	$\mathbf{y} = f_\theta(q(\mathbf{x}_s), \mathbf{x}_t)$	series conversion	training
Aligning	Similarity Match	ETP [Liu *et al.*, 2023a]	$\mathbf{y} = g_\phi(\mathbf{x}_s)$	align semantics of	complicated
		MATM [Han *et al.*, 2022]	$\mathcal{L} = \mathrm{sim}(g_\phi(\mathbf{x}_s), f_\theta(\mathbf{x}_t))$	different modalities;	design and
	LLM Backbone	GPT4TS [Zhou *et al.*, 2023a]	$\mathbf{y} = f_\theta(g_\phi(\mathbf{x}_s), \mathbf{x}_t)$	end-to-end training	fine-tuning
Vision as Bridge	Paired Data	ImageBind [Girdhar *et al.*, 2023]	$\mathcal{L} = \mathrm{sim}(g_\phi(\mathbf{x}_s), h_\psi(\mathbf{x}_v))$	additional visual	not hold
	TS Plots	[Wimmer and Rekabsaz, 2023]	$\mathbf{y} = h_\psi(\mathbf{x}_s)$	knowledge	for all data
Tool	Code	CTG++ [Zhong *et al.*, 2023]	$z = f_\theta(\mathbf{x}_t)$	empower LLM	optimization
	API	ToolLLM [Qin *et al.*, 2023]	$\mathbf{y} = z(\mathbf{x}_s)$	with more abilities	not end-to-end

Table 1: Summary of five major categories of applying LLMs for time series analysis, including their respective subcategories, representative works, mathematical formulations, advantages and limitations. q and $\mathbf{x}_v$ represent text-based quantization process and image data.

Time Series Plots as Images. CLIP-LSTM [Wimmer and Rekabsaz, 2023] transforms stock market data into sequences of texts and images of price charts, and leverages pre-trained CLIP vision-language model to generate features for downstream forecasting. Insight Miner [Zhang *et al.*, 2023b] converts time series windows into images using lineplot, and feeds images into vision language model LLaVA to generate time series trend descriptions.

3.5 Tool

This type of method does not directly use large language models to process time series. Instead, it applies large language models to generate indirect tools $z(\cdot)$, such as code and API calls, to benefit time series related tasks.

Code. CTG++ [Zhong *et al.*, 2023] applies GPT-4 to generate differentiable loss functions in a code format from text descriptions to guide the diffusion model to generate traffic trajectories. With this two-step translation, the large language model and diffusion model efficiently bridge the gap between user intent and traffic simulation.

API Call. ToolLLM [Qin *et al.*, 2023] introduces a general tool-use framework composed of data construction, model training, and evaluation. This framework includes API calls for time series tasks such as weather and stock forecasting.

Text Domain Knowledge. SHARE [Zhang *et al.*, 2023a] exploits the shared structures in human activity label names and proposes a sequence-to-sequence structure to generate label names as token sequences to preserve the shared label structures. It applies GPT-4 to augment semantics of label names. GG-LLM [Graule and Isler, 2023] leverages LLaMA-2 to encode world knowledge of common human behavioral patterns to predict human actions without further training. SCRL-LG [Ding *et al.*, 2023] leverages LLaMA-7B as stock feature selectors to extract meaningful representations from news headlines, which are subsequently employed in reinforcement learning for precise feature alignments.

4 Comparison within the Taxonomy

We compare the five categories of our taxonomy and provide general guidelines for which category to choose based on considerations of data, model, efficiency and optimization.

Data. When no training data is available and the objective is to apply LLM for time series in an zero-shot fashion, it is preferable to use prompting-based methods. This is because direct prompting enables the utilization of pre-trained language models' inherent capabilities without fine-tuning. However, representing numbers as strings can diminish the semantic value intrinsically tied to numerical data. Therefore, with adequate training data, quantization or aligning-based methods become more advantageous. As shown in Figure 2, these two categories are the most extensively studied ones in existing literature. Furthermore, if time series data can be interpreted or associated with visual representations, these representations can be incorporated to utilize the intrinsic knowledge embedded in the vision modality or pre-trained vision-language models.

Model. Prompting and tool integration methods tend to apply billion-parameter models as they often apply off-the-self LLMs without architectural modifications. By contrast, aligning and quantization methods vary from million to billion-parameter models, depending on the specific application requirements and available computational resources.

Efficiency. Prompting-based methods are not efficient for numerical data with high precision, as well as multivariate time series as it requires transforming each dimension into separate univariate time series, resulting in extremely long input. They are also less efficient for long-term predictions due to the computational demands of generating long sequences. These methods are more effective when dealing with simple numerical data that is richly interwoven with textual information, such as opening and closing stock prices in financial news articles. By contrast, quantization and aligning meth-

Domain	Dataset	Size	Major Modalities	Task
IoT	Ego4D[2] [Grauman *et al.*, 2022]	$3,670$h data, 3.85M narrations	text, IMU, video, audio, 3D	classification, forecasting
	DeepSQA[3] [Xing *et al.*, 2021]	25h data, 91K questions	text, imu	classification, QA
Finance	PIXIU[4] [Xie *et al.*, 2023b]	136K instruction data	text, tables	5 NLP tasks, forecasting
	MoAT[5] [Lee *et al.*, 2023]	6 datasets, 2K timesteps	text, time series	forecasting
Healthcare	Zuco 2.0[6] [Hollenstein *et al.*, 2019]	739 sentences	text, eye-tracking, EEG	classification, text generation
	PTB-XL[7] [Wagner *et al.*, 2020]	60h data, 71 unique statements	text, ECG	classification
	ECG-QA[8] [Oh *et al.*, 2023]	70 question templates	text, ECG	classification, QA
Audio	OpenAQA-5M[9] [Gong *et al.*, 2023]	5.6M (audio, QA) tuples	text, audio	tagging, classification
Music	MusicCaps[10] [Agostinelli *et al.*, 2023]	5.5K music clips	text, music	captioning, generation
Speech	CommonVoice[11] [Ardila *et al.*, 2019]	$7,335$h in 60 languages	text, speech	ASR, translation

Table 2: Summary of representative time series and text multimodal datasets.

ods are more efficient to handle long sequences, as time series are typically down-sampled or segmented into patches before feeding into large language models.

Optimization. Depending on the specific discretization technique, quantization based method may require a two-stage training process (such as first training the VQ-VAE model), which may result in sub-optimal performance compared with that achieved through end-to-end training in aligning methods. Using large language models as indirect tools empowers LLMs with more capabilities to manage numerical data, but also raises the level of complexity to optimize both LLMs and other components in an end-to-end fashion. Therefore, existing works of tool integration typically employ off-the-shelf LLMs without further fine-tuning.

5 Multimodal Datasets

Applying LLMs for time series benefits from the availability of multimodal time series and text data. In this section, we introduce representative multimodal datasets organized by their respective domains (Table 2). Due to space limit, additional datasets are listed in our Github repository[12].

Internet of Things (IoT). Human activity recognition is an important task in IoT domain, which identifies human activities given time series collected with IoT devices (such as IMU sensors). The corresponding text data are the labels or text descriptions of these activities. Ego4D [Grauman *et al.*, 2022] presents 3,670 hours of daily-life activity data with multiple modalities, including IMU time series, and dense temporally-aligned textual descriptions of the activities. Ego-Exo4D further offers three kinds of paired natural language datasets including expert commentary, narrate-and-act descriptions provided by the participants themselves, and atomic action descriptions similar as Ego4D. DeepSQA [Xing *et al.*, 2021] presents a generalized Sensory Question Answering (SQA) framework to facilitate querying raw sensory data related to human activities using natural language.

Finance. PIXIU [Xie *et al.*, 2023b] presents multi-task and multi-modal instruction tuning data in the financial domain with 136K data samples. It contains both financial natural language understanding and prediction tasks, and covers 9 datasets of multiple modalities such as text and time series. MoAT [Lee *et al.*, 2023] constructs multimodal datasets with textual information paired with time series for each timestep, such as news articles extracted with relevant keywords, mostly covering finance related domains such as fuel, metal, stock and bitcoin.

Healthcare. Zuco datasets [Hollenstein *et al.*, 2019] contain simultaneous eye-tracking and EEG during natural reading and during annotation. PTB-XL [Wagner *et al.*, 2020] offers comprehensive metadata regarding ECG annotated by expert cardiologists, covering information such as ECG reports, diagnostic statements, diagnosis likelihoods, and signal-specific properties. Based on PTB-XL, ECG-QA [Oh *et al.*, 2023] introduces the first Question Answering dataset for ECG analysis, containing 70 question templates that cover a wide range of clinically relevant ECG topics.

Audio/Music/Speech. AudioSet is a collection of 2 million 10-second audio clips excised from YouTube videos and labeled with the sounds that the clip contains from a set of 527 labels. OpenAQA-5M [Gong *et al.*, 2023] dataset consists of 1.9 million closed-ended and 3.7 million open-ended, diverse (audio, question, answer) tuples. MusicCaps [Agostinelli *et al.*, 2023] is a high-quality music caption dataset, including 5.5K music clips. MTG-Jamendo is a dataset with 55,000 audio songs in various languages. Libri-Light is an English dataset encompassing 60,000 hours of speech data. Common-

[2]https://ego4d-data.org/
[3]https://github.com/nesl/DeepSQA
[4]https://github.com/chancefocus/PIXIU
[5]https://openreview.net/pdf?id=uRXxnoqDHH
[6]https://osf.io/2urht/
[7]https://physionet.org/content/ptb-xl/1.0.3/
[8]https://github.com/Jwoo5/ecg-qa
[9]https://github.com/YuanGongND/ltu
[10]https://www.kaggle.com/datasets/googleai/musiccaps
[11]https://commonvoice.mozilla.org/en/datasets
[12]https://github.com/xiyuanzh/awesome-llm-time-series

Voice [Ardila *et al.*, 2019] is a multilingual speech dataset consisting of 7,335 validated hours in 60 languages.

These datasets offer valuable benchmarks for multimodal time series and text analysis. These contain both time series focused tasks, including classification, which is evaluated using accuracy and macro-F1 scores, and forecasting, which utilizes metrics such as MSE, MAE, RMSE, and MAPE, as well as NLP focused tasks such as captioning, question answering, and translation, assessed through BLEU, ROUGE, METEOR, and EM scores, among others.

6 Challenges and Future Directions

6.1 Theoretical Understanding

Existing works empirically show the benefits of applying LLMs for time series analysis. For example, recent works have empirically shown that large language models learn linear representations of space and time across multiple scales that are robust to prompting variations. Despite these empirical findings, there remains a gap in theoretical understanding of how models, primarily trained on textual data, can effectively interpret numerical time series. As a preliminary theoretical analysis, it is proved that Transformer models can universally approximate arbitrary continuous sequence-to-sequence functions on a compact domain [Yun *et al.*, 2019]. Additionally, GPT4TS [Zhou *et al.*, 2023a] theoretically shows that such generic capability of large language models can be related to Principal Component Analysis (PCA), as minimizing the gradient with respect to the self-attention layer shares similarities with PCA. Further investigations on the generalizability of large language models on numerical data is essential to establish solid understanding of the synergy between LLMs and time series analysis.

6.2 Multimodal and Multitask Analysis

Existing papers that apply LLMs for time series analysis mostly focus on single modality and single task at a time, such as forecasting, classification, text generation, and do not support simultaneous multimodal and multitask analysis. In computer vision and audio domains, models such as Unified-IO and UniAudio [Yang *et al.*, 2023] have unified multiple input modalities into a sequence of discrete vocabulary tokens to support multiple tasks within a single transformer-based architecture. More research into leveraging LLMs for multimodal and multitask analysis would lead to more powerful time series foundation models.

6.3 Efficient Algorithms

Time series, especially those that are multivariate or possess long history information may increase the computational complexity for existing large language models. Patching (treating each segmented patch as a token) has been a widely adopted strategy to improve performance as well as reduce complexity, but large patches may obscure the semantic information of time series and negatively impact the performance. Therefore, developing more efficient algorithms is especially crucial for facilitating large-scale time series analysis with LLMs and enhancing interactions with end users.

6.4 Combining Domain Knowledge

Combining existing statistical domain knowledge with LLMs may further boost the model's capability for time series analysis. For example, TEMPO [Cao *et al.*, 2023] applies time series seasonal-trend decomposition and treats decomposed components as different semantic inductive biases as input to the pre-trained transformer. FreqTST [Li *et al.*, 2023] leverages insights from the frequency domain by tokenizing single time series into frequency units with weights for downstream forecasting. Further incorporating domain knowledge, such as wavelet decomposition, auto-correlation analysis, and empirical mode decomposition may augment LLMs' capabilities in analyzing time series data.

6.5 Customization and Privacy

Existing works on large language models and time series analysis typically train a global model for all end users. Training customized models for different users based on the global model may bring further benefits and flexibility. Another important consideration is privacy, especially as many time series data are collected in private settings for clinical purposes or smart home applications. Federated learning offers a solution by enabling the training of machine learning models across multiple decentralized devices holding local data samples. Advancing research into model customization and user privacy preservation like federated learning would broaden the utility of LLM-empowered time series analysis.

7 Conclusion

We present the first survey that systematically analyzes the categorization of transferring knowledge from large language models for numerical time series analysis: direct prompting, time series quantization, aligning, the use of the vision modality to connect text and time series, and the integration of large language models with other analytical tools. For each category, we introduce their mathematical formulation, representative works, and compare their advantages and limitations. We also introduce representative multimodal text and time series datasets in various domains such as healthcare, IoT, finance, and audio. Concluding the paper, we outline the challenges and emerging directions for potential future research of LLM-empowered time series analysis.

Acknowledgements

Our work is supported in part by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. Our work is also sponsored by NSF CAREER Award 2239440, NSF Proto-OKN Award 2333790, NIH Bridge2AI Center Program under award 1U54HG012510-01, Cisco-UCSD Sponsored Research Project, as well as generous gifts from Google, Adobe, and Teradata. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes not withstanding any copyright annotation hereon.

References

[Agostinelli *et al.*, 2023] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv:2301.11325*, 2023.

[Ardila *et al.*, 2019] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv:1912.06670*, 2019.

[Cai *et al.*, 2023] Yifu Cai, Mononito Goswami, Arjun Choudhry, Arvind Srinivasan, and Artur Dubrawski. Jolt: Jointly learned representations of language and time-series. In *NeurIPS Deep Generative Models for Health Workshop*, 2023.

[Cao *et al.*, 2023] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv:2310.04948*, 2023.

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

[Chung *et al.*, 2023] Hyunseung Chung, Jiho Kim, Joon-myoung Kwon, Ki-Hyun Jeon, Min Sung Lee, and Edward Choi. Text-to-ecg: 12-lead electrocardiogram synthesis conditioned on clinical text reports. In *ICASSP*, pages 1–5. IEEE, 2023.

[Ding *et al.*, 2023] Yujie Ding, Shuai Jia, Tianyi Ma, Bingcheng Mao, Xiuze Zhou, Liuliu Li, and Dongming Han. Integrating stock features and global information via large language models for enhanced stock return prediction. *arXiv:2310.05627*, 2023.

[Duan *et al.*, 2023] Yiqun Duan, Charles Zhou, Zhen Wang, Yu-Kai Wang, and Chin-teng Lin. Dewave: Discrete encoding of eeg waves for eeg to text translation. In *NeurIPS*, 2023.

[Girdhar *et al.*, 2023] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, pages 15180–15190, 2023.

[Gong *et al.*, 2023] Yuan Gong, Hongyin Luo, Alexander H Liu, Leonid Karlinsky, and James Glass. Listen, think, and understand. *arXiv:2305.10790*, 2023.

[Graule and Isler, 2023] Moritz A Graule and Volkan Isler. Gg-llm: Geometrically grounding large language models for zero-shot human activity forecasting in human-aware task planning. *arXiv:2310.20034*, 2023.

[Grauman *et al.*, 2022] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, pages 18995–19012, 2022.

[Gruver *et al.*, 2023] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero-shot time series forecasters. *arXiv:2310.07820*, 2023.

[Han *et al.*, 2022] William Han, Jielin Qiu, Jiacheng Zhu, Mengdi Xu, Douglas Weber, Bo Li, and Ding Zhao. An empirical exploration of cross-domain alignment between language and electroencephalogram. *arXiv:2208.06348*, 2022.

[Hollenstein *et al.*, 2019] Nora Hollenstein, Marius Troendle, Ce Zhang, and Nicolas Langer. Zuco 2.0: A dataset of physiological recordings during natural reading and annotation. *arXiv:1912.00903*, 2019.

[Jin *et al.*, 2023a] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv:2310.01728*, 2023.

[Jin *et al.*, 2023b] Ming Jin, Qingsong Wen, Yuxuan Liang, Chaoli Zhang, Siqiao Xue, Xue Wang, James Zhang, Yi Wang, Haifeng Chen, Xiaoli Li, et al. Large models for time series and spatio-temporal data: A survey and outlook. *arXiv:2310.10196*, 2023.

[King *et al.*, 2023] Ryan King, Tianbao Yang, and Bobak J Mortazavi. Multimodal pretraining of medical time series and notes. In *ML4H*, pages 244–255. PMLR, 2023.

[Lee *et al.*, 2023] Geon Lee, Wenchao Yu, Wei Cheng, and Haifeng Chen. Moat: Multi-modal augmented time series forecasting. 2023.

[Leng *et al.*, 2023] Zikang Leng, Hyeokhyen Kwon, and Thomas Plötz. Generating virtual on-body accelerometer data from virtual textual descriptions for human activity recognition. *arXiv:2305.03187*, 2023.

[Li *et al.*, 2023] Junkai Li, Weizhi Ma, and Yang Liu. Modeling time series as text sequence a frequency-vectorization transformer for time series forecasting. 2023.

[Liu *et al.*, 2023a] Che Liu, Zhongwei Wan, Sibo Cheng, Mi Zhang, and Rossella Arcucci. Etp: Learning transferable ecg representations via ecg-text pre-training. *arXiv:2309.07145*, 2023.

[Liu *et al.*, 2023b] Xin Liu, Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine, Jiening Zhan, Ming-Zher Poh, Shun Liao, Paolo Di Achille, and Shwetak Patel. Large language models are few-shot health learners. *arXiv:2305.15525*, 2023.

[Ma *et al.*, 2023] Qianli Ma, Zhen Liu, Zhenjing Zheng, Ziyang Huang, Siying Zhu, Zhongzhong Yu, and James T Kwok. A survey on time-series pre-trained models. *arXiv:2305.10716*, 2023.

[Moon *et al.*, 2022] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Alireza Dirafzoon, Aparajita Saraf, Amy Bearman, and Babak Damavandi. Imu2clip: Multimodal contrastive learning for imu motion sensors from egocentric videos and text. *arXiv:2210.14395*, 2022.

[Moon *et al.*, 2023] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Tushar Nagarajan, Matt Smith, Shashank

Jain, Chun-Fu Yeh, Prakash Murugesan, Peyman Heidari, Yue Liu, et al. Anymal: An efficient and scalable any-modality augmented language model. *arXiv:2309.16058*, 2023.

[Oh *et al.*, 2023] Jungwoo Oh, Seongsu Bae, Gyubok Lee, Joon-myoung Kwon, and Edward Choi. Ecg-qa: A comprehensive question answering dataset combined with electrocardiogram. *arXiv:2306.15681*, 2023.

[Qin *et al.*, 2023] Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv:2307.16789*, 2023.

[Qiu *et al.*, 2023] Jielin Qiu, William Han, Jiacheng Zhu, Mengdi Xu, Michael Rosenberg, Emerson Liu, Douglas Weber, and Ding Zhao. Transfer knowledge from natural language to electrocardiography: Can we detect cardiovascular disease through language models? *arXiv:2301.09017*, 2023.

[Rubenstein *et al.*, 2023] Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. Audiopalm: A large language model that can speak and listen. *arXiv:2306.12925*, 2023.

[Su *et al.*, 2023] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to instruction-follow them all. *arXiv:2305.16355*, 2023.

[Sun *et al.*, 2023] Chenxi Sun, Yaliang Li, Hongyan Li, and Shenda Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. *arXiv:2308.08241*, 2023.

[Talukder and Gkioxari, 2023] Sabera J Talukder and Georgia Gkioxari. Time series modeling at scale: A universal representation across tasks and domains. 2023.

[Tang *et al.*, 2023] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models. *arXiv:2310.13289*, 2023.

[Wagner *et al.*, 2020] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):154, 2020.

[Wang and Ji, 2022] Zhenhailong Wang and Heng Ji. Open vocabulary electroencephalography-to-text decoding and zero-shot sentiment classification. In *AAAI*, 2022.

[Wimmer and Rekabsaz, 2023] Christopher Wimmer and Navid Rekabsaz. Leveraging vision-language models for granular market change prediction. *arXiv:2301.10166*, 2023.

[Wu *et al.*, 2023] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv:2303.17564*, 2023.

[Xie *et al.*, 2023a] Qianqian Xie, Weiguang Han, Yanzhao Lai, Min Peng, and Jimin Huang. The wall street neophyte: A zero-shot analysis of chatgpt over multimodal stock movement prediction challenges. *arXiv:2304.05351*, 2023.

[Xie *et al.*, 2023b] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. Pixiu: A large language model, instruction data and evaluation benchmark for finance. *arXiv:2306.05443*, 2023.

[Xing *et al.*, 2021] Tianwei Xing, Luis Garcia, Federico Cerutti, Lance Kaplan, Alun Preece, and Mani Srivastava. Deepsqa: Understanding sensor data via question answering. In *IoTDI*, 2021.

[Xue and Salim, 2022] Hao Xue and Flora D Salim. Promptcast: A new prompt-based learning paradigm for time series forecasting. 2022.

[Yang *et al.*, 2023] Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. Uniaudio: An audio foundation model toward universal audio generation. *arXiv:2310.00704*, 2023.

[Yu *et al.*, 2023] Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. Temporal data meets llm–explainable financial time series forecasting. *arXiv:2306.11025*, 2023.

[Yun *et al.*, 2019] Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? *arXiv:1912.10077*, 2019.

[Zhang *et al.*, 2023a] Xiyuan Zhang, Ranak Roy Chowdhury, Jiayun Zhang, Dezhi Hong, Rajesh K Gupta, and Jingbo Shang. Unleashing the power of shared label structures for human activity recognition. In *CIKM*, pages 3340–3350, 2023.

[Zhang *et al.*, 2023b] 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 AI4Science*, 2023.

[Zhong *et al.*, 2023] Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. Language-guided traffic simulation via scene-level diffusion. *arXiv:2306.06344*, 2023.

[Zhou *et al.*, 2023a] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained lm. *arXiv:2302.11939*, 2023.

[Zhou *et al.*, 2023b] Yunjiao Zhou, Jianfei Yang, Han Zou, and Lihua Xie. Tent: Connect language models with iot sensors for zero-shot activity recognition. *arXiv:2311.08245*, 2023.