EVITIME: EVIDENCE-GUIDED VISION-LANGUAGE LEARNING FOR MULTIMODAL TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal time series forecasting aims to exploit complementary numerical, visual, and semantic information for more accurate predictions. However, existing methods often treat temporal observations as equally reliable, rely on limited visual representations to characterize temporal structures, and introduce generic textual descriptions that are weakly aligned with individual forecasting instances. To address these limitations, **EviTime** is proposed as an **Evi**dence-guided vision-language learning framework for multimodal **Time** series forecasting. First, EviTime introduces evidential reasoning into temporal representation learning by combining Dempster-Shafer Theory (DST), nonlinear Basic Probability Assignment (BPA) generation, and Deng entropy to characterize observation reliability and suppress ambiguous information. Second, numerical sequences are further transformed into three-channel visual representations using Gramian Angular Summation Fields (GASF), Gramian Angular Difference Fields (GADF), and amplitude information, enabling a pretrained Vision-Language Model (VLM) to capture complementary structural patterns. Third, hierarchical metadata prompts encode dataset-, task-, and sample-level semantics, while a cross-attention metadata router extracts instance-relevant language information. Temporal, visual, and semantic representations are then aligned and adaptively fused for forecasting. Finally, extensive experiments on multiple benchmarks demonstrate consistent improvements over competitive baselines, with Mean Squared Error (MSE) and Mean Absolute Error (MAE) reduced by 18.4% and 13.6%, respectively, compared with the average performance of 10 state-of-the-art methods. The source code is publicly available at `https://anonymous.4open.science/r/EvidentialTime-CD2F`.

1 INTRODUCTION

Time series forecasting has emerged as a fundamental machine learning task in critical domains such as energy management and traffic scheduling (Chowdhury et al., 2026; Wang et al., 2026c). By modeling temporal patterns embedded in historical observations, forecasting methods enable the estimation of future states, providing quantitative support for resource optimization (Shu & Lampos, 2026) and risk warning (Chang et al., 2026). However, with the widespread deployment of sensing devices, traditional time series data have increasingly evolved into continuous data streams (Zhang et al., 2026a; Fan et al., 2026b). This evolution has drastically amplified the complexity of the data, imposing unprecedented demands on the generalization capability of models (Wang et al., 2026d; Shi et al., 2026). Consequently, designing effective forecasting frameworks for time series streams has become a key research topic (Ye et al., 2027; Wang et al., 2026a).

The rapid advancement of deep learning (Zhang et al., 2026c; Cetina et al., 2026) has significantly propelled the evolution of time series forecasting methods (Chen et al., 2025; Guo et al., 2026). Early statistical models such as ARIMA (Shahriari et al., 2025) are limited by linear assumptions and struggle to capture nonlinear patterns. Subsequently, architectures specifically designed for time series have emerged, including CNN-based TimesNet (Wu et al., 2023), Transformer-based iTransformer (Liu et al., 2024a), and MLP-based DLinear (Zeng et al., 2023). These models have achieved substantial performance improvements. Building upon these advances, time se-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

ries foundation models (Li et al., 2025b; Liu et al., 2024b) have emerged, drawing inspiration from pretraining paradigms in natural language processing (Lyu et al., 2026). By pretraining on multi-domain time series corpora, time series foundation models learn transferable representations (Gu et al., 2026) and achieve strong zero-shot generalization (Qin et al., 2025; Ma et al., 2026).

The aforementioned forecasting models rely primarily on numerical sequences, leaving structural and semantic information underexplored (Savchenko & Kachan, 2026). Multimodal forecasting integrates complementary features from diverse information sources to construct data representations (Jia et al., 2026; Park et al., 2026). Therefore, multimodal learning opens new paradigms for time series forecasting (Wang et al., 2026b). In particular, VLMs (Zhang et al., 2026b; Zhong et al., 2025) enable cross-modal alignment between visual and textual semantics, as shown in Figure 1.

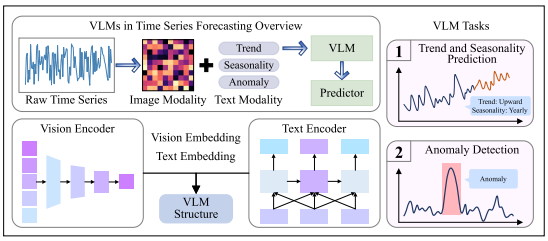

Figure 1: Overview of VLMs in time series forecasting. VLMs process time series represented through image and text modalities, supporting the prediction.

Despite recent progress, existing time series forecasting methods still face three key limitations (Kui et al., 2026). First, most methods encode noisy or ambiguous observations into deterministic representations, making it difficult to distinguish reliable information from ambiguous or conflicting observations (Liu et al., 2025). Second, image-based approaches typically rely on a single encoding scheme, which captures only partial temporal structures and limits the joint representation of similarity patterns, directional variations, and amplitude dynamics (Fan et al., 2026a). Third, textual information is often introduced through generic descriptions, leading to insufficient alignment between semantic knowledge and individual time series instances (Wang et al., 2026e).

To address these limitations, this paper proposes EviTime. The main contributions are summarized as follows. (1) An evidence-aware temporal representation scheme models observation reliability using DST-based BPAs from temporal and variable perspectives, with Deng-entropy-guided adaptive fusion integrating the two evidential views. (2) EviTime incorporates complementary structural and semantic information into evidence-aware forecasting. Three-channel GAF representations capture temporal similarity, directional variation, and amplitude patterns, while hierarchical metadata prompts provide dataset-, task-, and sample-level semantics. These heterogeneous representations are jointly aligned and fused with temporal features. (3) Extensive experiments covering long-term and short-term forecasting, robustness to input perturbations, few-shot forecasting, and zero-shot cross-dataset transfer demonstrate the effectiveness of EviTime.

2 Related Work

This paper introduces relevant research on VLMs, DST, and metadata. Detailed theoretical foundations are provided in Appendix A.

2.1 Vision-Language Model (VLM)

The rapid development of VLMs has introduced a new paradigm for cross-modal representation learning. Representative models include CLIP (Radford et al., 2021), BLIP-2 (Li et al., 2023), and ViLT (Kim et al., 2021). Pretrained on large-scale image-text pairs, these models learn rich cross-modal associations. This cross-modal representation capability offers unique advantages for time series forecasting (He et al., 2026). On the one hand, image representations enable VLM vision encoders to capture periodic patterns and local fluctuations. On the other hand, incorporating constructed text prompts allows text encoders to provide semantic guidance.

2.2 Dempster-Shafer Theory (DST)

DST (Shafer, 1976; Dempster, 1967) provides a flexible framework for representing uncertainty (Li et al., 2025a). In this paper, the frame of discernment is defined as $\Theta = \{e_1, e_2, e_3\}$, where

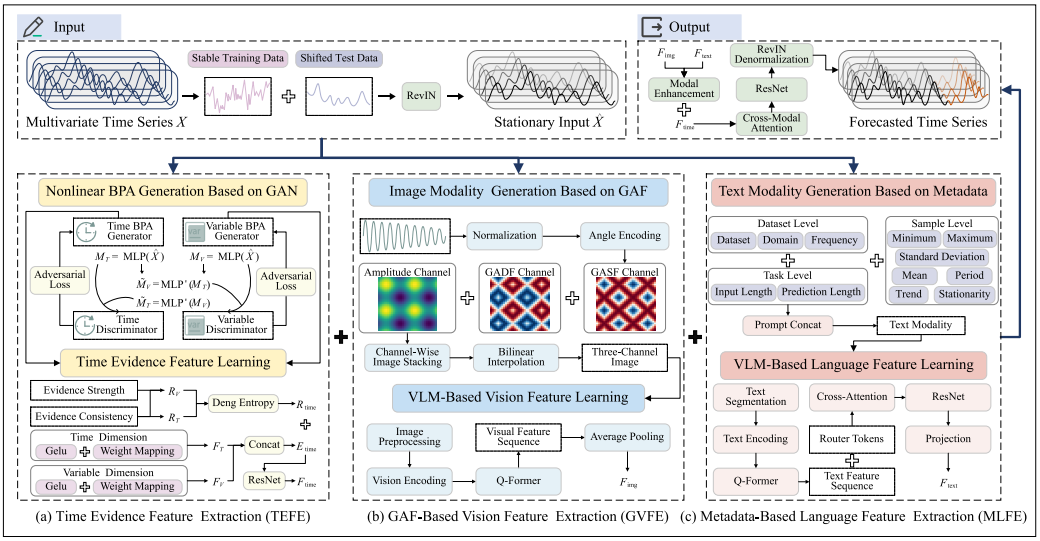

Figure 2: The framework of EviTime. EviTime extracts evidence-aware features through DST and then introduces image and text modalities to capture periodic patterns. EviTime, along with the time and space complexity analysis, is provided in Appendix B.

e_1, e_2, and e_3 denote three latent reliability states of a time series observation: reliable, locally perturbed, and unreliable, respectively. These hypotheses describe mutually exclusive observation states, while ambiguity between them is represented by composite subsets. In DST, belief masses are assigned only to non-empty subsets of Θ. Therefore, the focal-element set is $\mathcal{F} = 2^{\Theta} \setminus \{\emptyset\} = \{A_1, A_2, \ldots, A_7\}$, yielding $K = |\mathcal{F}| = 2^{|\Theta|} - 1 = 7$. A BPA is defined as $m : 2^{\Theta} \to [0, 1]$, satisfying $m(\emptyset) = 0$ and $\sum_{A \in \mathcal{F}} m(A) = 1$. A mass assigned to a singleton represents support for a specific reliability state, whereas a mass assigned to a composite focal element expresses unresolved evidence among multiple states (Deng, 2015; Zhan et al., 2025).

2.3 Metadata

Metadata refers to information that describes other data and is commonly defined as "data about data" (Riley, 2017). It records data characteristics, contextual information, and semantic descriptions, thereby providing supplementary explanations for raw observations. In time series analysis, metadata has been widely used at both the data management and analytical levels. On the one hand, metadata records information such as data sources, acquisition settings, and statistical summaries, facilitating data organization and retrieval (Li et al., 2020; He et al., 2025). On the other hand, it captures the contextual semantics of forecasting scenarios, helping models better understand the characteristics and requirements of different forecasting tasks (Wang et al., 2024).

3 Methodology

This paper proposes EviTime for time series forecasting, as illustrated in Figure 2.

3.1 Problem Statement

For time series forecasting, given $X \in \mathbb{R}^{B \times L \times C}$, the normalized series $\hat{X} \in \mathbb{R}^{B \times L \times C}$ is obtained through RevIN, where B, L, and C denote the batch size, sequence length, and number of variables, respectively. The task is to predict the next PL steps, denoted as $Y' \in \mathbb{R}^{B \times PL \times C}$.

3.2 Time Evidence Feature Extraction (TEFE)

This paper proposes TEFE, which performs uncertainty-aware probabilistic representation learning.

3.2.1 EVIDENTIAL UNCERTAINTY REPRESENTATION

The model integrates DST and Deng entropy for modeling uncertainty. Sinusoidal positional encoding is introduced to distinguish observations at different temporal positions. Specifically, the normalized historical input $\hat{X} \in \mathbb{R}^{B\times L\times C}$ is first projected along the temporal dimension to obtain an extended representation $X' \in \mathbb{R}^{B\times(L+PL)\times C}$. The first L positions represent the encoded historical context, while the additional PL positions serve as latent forecasting positions generated solely from the historical input. No future ground-truth observations are involved in this projection. The position-enhanced representation is then defined as $X_{pe} = X' + PE$. For $t = 0, 1, \ldots, (L+PL)-1$ and $i = 0, 1, \ldots, \lfloor C/2 \rfloor - 1$, $PE \in \mathbb{R}^{1\times(L+PL)\times C}$ is computed as $PE_{(t,2i)} = \sin(\frac{t}{10000^{2i/C}})$ and $PE_{(t,2i+1)} = \cos(\frac{t}{10000^{2i/C}})$. This encoding preserves the relative ordering of observations.

Ambiguous observations may arise from temporal fluctuations or inconsistent dependencies among variables. Therefore, two dedicated multilayer perceptrons (MLPs) are employed to generate evidential representations from the temporal and variable perspectives, respectively: $M_T = \mathrm{Softmax}(\mathrm{MLP}_{\text{time}}(X_{\text{pe}}^\top)^\top) \in \mathbb{R}^{B\times(L+PL)\times C\times K}$, $M_V = \mathrm{Softmax}(\mathrm{MLP}_{\text{var}}(X_{\text{pe}}^\top)) \in \mathbb{R}^{B\times C\times(L+PL)\times K}$. Here, the Softmax operation is applied along the K-dimensional focal-element space, converting the nonlinear outputs into BPAs.

Since reliable evidence assignments should remain locally stable, evidence consistency is measured by the normalized deviation between each BPA and its local average: $\mathrm{EC}_T = 1 - \frac{|M_T - \mathcal{A}(M_T)|}{\mathbb{E}[|M_T - \mathcal{A}(M_T)|]+\epsilon}$, while EC_V is computed analogously from M_V along the variable dimension. Here, $\mathcal{A}(\cdot)$ denotes the corresponding 1D average pooling, $\mathbb{E}[\cdot]$ is the global mean, and ϵ prevents division by zero. The BPAs are retained as evidence strength: $\mathrm{ES}_T = M_T$ and $\mathrm{ES}_V = M_V$.

Evidence consistency reflects the local stability of BPAs, while evidence strength retains their original belief information. Their concatenation is used to estimate the contribution of each focal element: $R_T = \sum_{k=1}^{K} w_T^{(k)} M_T^{(k)}$, where $w_T = \sigma(\mathrm{MLP}_{\text{weight}}(\mathrm{concat}(\mathrm{ES}_T, \mathrm{EC}_T)))$. R_V and w_V are computed analogously from ES_V and EC_V.

To further distinguish concentrated and dispersed BPAs, Deng entropy is used to quantify evidence uncertainty: $H_d(M) = -\sum_{k=1}^{K} M^{(k)} \log_2 \frac{M^{(k)}}{2^{|A_k|}-1}$, with $H_T = H_d(M_T)$ and $H_V = H_d(M_V)$. A learnable gate combines the two evidential views according to their uncertainty:

$$R_{\text{time}} = g \odot R_T + (1 - g) \odot R_V, \tag{1}$$

where $g = \sigma(\mathrm{MLP}_{\text{gate}}([H_T, H_V]))$. Before fusion, R_V and H_V are rearranged to match the temporal-first order of R_T and H_T. Thus, focal-element weighting evaluates reliability within each view, while the entropy balances the two views.

Adversarial Evidence Learning: To refine evidence representations, two generators construct synthetic evidence through cross-dimensional mappings. Each generator first transforms the evidence from the opposite view through an MLP and then applies Softmax over the non-empty focal elements: $\widetilde{M}_T = \mathrm{Softmax}\big(\mathrm{MLP}'_{\text{time}}(M_V)\big)$, $\widetilde{M}_V = \mathrm{Softmax}\big(\mathrm{MLP}'_{\text{var}}(M_T)\big)$. Thus, $\widetilde{M}_T$ and $\widetilde{M}_V$ are normalized synthetic BPAs defined on the same focal-element space as M_T and M_V. The discriminators distinguish the original evidence M_q from the synthetic evidence $\widetilde{M}_q$, where $q \in \{T, V\}$ and $D_q(\cdot) = \tanh\big(w_2\big(\mathrm{ReLU}(w_1(\cdot))\big)\big)$. The adversarial losses are $\mathcal{L}_D = \frac{1}{2}\sum_{q\in\{T,V\}} \mathbb{E}[(D_q(M_q)-1)^2 + (D_q(\widetilde{M}_q)+1)^2]$ and $\mathcal{L}_G = \frac{1}{2}\sum_{q\in\{T,V\}} \mathbb{E}[(D_q(\widetilde{M}_q)-1)^2]$. This adversarial process encourages the generators to produce evidence that better follows the intrinsic patterns of time series. Sparsity and orthogonality regularizations reduce cross-view redundancy: $\mathcal{L}_{\text{sparse}} = \frac{1}{2}(\mathbb{E}[H_T] + \mathbb{E}[H_V])$, $\mathcal{L}_{\text{ortho}} = \mathbb{E}[(\frac{\mathbb{E}_{\text{var}}[M_T]\mathbb{E}_{\text{var}}[M_V]}{\|\mathbb{E}_{\text{var}}[M_T]\|\|\mathbb{E}_{\text{var}}[M_V]\|+\epsilon})^2]$. The overall loss is:

$$\mathcal{L}_{\text{TEFE}} = \lambda_s \mathcal{L}_{\text{sparse}} + \lambda_o \mathcal{L}_{\text{ortho}} + \lambda_{\text{adv}}(\mathcal{L}_D + \mathcal{L}_G), \tag{2}$$

where λ_s, λ_o, and λ_{adv} are the weights corresponding to the loss functions.

3.2.2 TIME SERIES FEATURE LEARNING

The preceding module focuses on the reliability of temporal observations, while reliable evidence alone cannot fully describe temporal evolution and cross-variable dependencies. There-

fore, two complementary views are constructed from the extended representation X': the temporal view $X_T = X' \in \mathbb{R}^{B\times(L+PL)\times C}$, which preserves the temporal structure, and the variable view $X_V = \text{permute}(X', (0, 2, 1)) \in \mathbb{R}^{B\times C\times(L+PL)}$. Two parallel projections are applied: $F_T = X_T W_1 + b_1 \in \mathbb{R}^{B\times(L+PL)\times d}$, $F_V = X_V W_2 + b_2 \in \mathbb{R}^{B\times C\times d}$, where $W_1 \in \mathbb{R}^{C\times d}$, $W_2 \in \mathbb{R}^{(L+PL)\times d}$, and $b_1, b_2 \in \mathbb{R}^d$. Since F_T and F_V have different structural dimensions, F_T is expanded along the variable dimension, while F_V is expanded along the temporal dimension: $\bar{F}_T = \text{Broadcast}_C(F_T)$, $\bar{F}_V = \text{Broadcast}_{L+PL}(F_V) \in \mathbb{R}^{B\times(L+PL)\times C\times d}$. The aligned features are fused as

$$E'_{\text{time}} = \psi([\bar{F}_T \| \bar{F}_V]), \quad E_{\text{time}} = \sum_{i=1}^{d} E'^{(i)}_{\text{time}}, \tag{3}$$

where $\psi(\cdot)$ denotes a learnable linear mapping.

Finally, residual fusion integrates evidence with temporal and variable dependencies to obtain time series features:

$$F_{\text{time}} = R_{\text{time}} + E_{\text{time}}. \tag{4}$$

3.3 GAF-Based Vision Feature Extraction (GVFE)

A one-dimensional sequence mainly represents point-wise variations and does not explicitly expose the relationships between time steps. GVFE maps the sequence into a two-dimensional relational manifold, where each position describes the interaction between a pair of time points. Specifically, GASF captures pairwise angular similarity, GADF characterizes directional changes, and the amplitude channel preserves the original magnitude information.

3.3.1 Three-Channel Image Modality Generation

Given the normalized univariate time series $\hat{X}_v = \{\hat{x}_t\}_{t=1}^{L}$ with $\hat{x}_t \in [-1, 1]$, each value is first mapped to the polar angle $\phi_t = \arccos(\hat{x}_t) \in [0, \pi]$, so that pairwise temporal relations can be represented on a 2D plane. Based on this angle encoding, GASF and GADF are defined as $\mathcal{I}^{(1)}_{\text{GAF}} = \text{GASF}_{t_1,t_2} = \cos(\phi_{t_1} + \phi_{t_2}) = \hat{x}_{t_1}\hat{x}_{t_2} - \sqrt{1-\hat{x}^2_{t_1}}\sqrt{1-\hat{x}^2_{t_2}}$, and $\mathcal{I}^{(2)}_{\text{GAF}} = \text{GADF}_{t_1,t_2} = \sin(\phi_{t_1} - \phi_{t_2}) = \hat{x}_{t_2}\sqrt{1-\hat{x}^2_{t_1}} - \hat{x}_{t_1}\sqrt{1-\hat{x}^2_{t_2}}$, which capture complementary similarity and transition patterns, respectively. To avoid losing the original numerical magnitude after angular transformation, a third channel is further introduced as $\mathcal{I}^{(3)}_{\text{GAF}} = (\hat{x}_{t_1} + \hat{x}_{t_2})/2$. Therefore, the three-channel image is constructed by stacking $\mathcal{I}^{(1)}_{\text{GAF}}$, $\mathcal{I}^{(2)}_{\text{GAF}}$, and $\mathcal{I}^{(3)}_{\text{GAF}}$.

For multivariate series, channel-wise GAF images are computed independently and aggregated as follows: $I_{\text{GAF}} = \frac{1}{C}\sum_{v=1}^{C} \text{Stack}(\mathcal{I}^{(1)}_{\text{GAF},v}, \mathcal{I}^{(2)}_{\text{GAF},v}, \mathcal{I}^{(3)}_{\text{GAF},v})$, which reduces redundancy across variables while preserving shared temporal structures. Finally, I_{GAF} is resized via bilinear interpolation to $\mathbb{R}^{B\times 3\times h\times w}$ for compatibility with VLMs.

3.3.2 Vision Feature Learning

The three-channel GAF image is fed into the vision encoder and Q-Former of the pretrained VLM to extract patch-level structural patterns and aggregate them into compact visual semantics: $\text{VF} = \text{VLM}_{\text{vision}}(I_{\text{GAF}}) \in \mathbb{R}^{B\times n\times d}$, where n denotes the number of image patches and d is the feature dimension. Mean pooling is applied over the query dimension to obtain an image representation:

$$F_{\text{img}} = \text{AvgPool}(\text{VF}) \in \mathbb{R}^{B\times d}. \tag{5}$$

3.4 Metadata-Based Language Feature Extraction (MLFE)

The image modality describes structural patterns in time series, but it does not explicitly specify the data domain, forecasting setting, or statistical state of each input sample. These factors affect how similar visual patterns should be interpreted across different datasets and tasks. Therefore, MLFE converts metadata into textual prompts and uses the language branch of the VLM to provide contextual semantics.

3.4.1 METADATA-BASED TEXT MODALITY GENERATION

To organize metadata, MLFE constructs the text modality from three complementary granularities. (1) Dataset-Level Metadata $\mathcal{P}_{\text{dataset}}$: It describes the dataset domain and sampling frequency, providing global context for interpreting the observations. (2) Task-Level Metadata $\mathcal{P}_{\text{task}}$: It specifies the input length and prediction length, defining the temporal scope and forecasting objective. (3) Sample-Level Metadata $\mathcal{P}_{\text{sample}}$: It characterizes the current input using minimum, maximum, mean, standard deviation, period, trend, and stationarity, all of which are computed exclusively from the historical input window, thereby reflecting sample-specific dynamics. These metadata are converted into structured descriptions and concatenated to form prompts: $\mathcal{P} = \text{concat}(\mathcal{P}_{\text{dataset}}, \mathcal{P}_{\text{task}}, \mathcal{P}_{\text{sample}})$.

3.4.2 LANGUAGE FEATURE LEARNING

The prompt $\mathcal{P}$ is tokenized and encoded by the text encoder and Q-Former of the VLM: $\text{TF} = \text{VLM}_{\text{text}}(\mathcal{P}) \in \mathbb{R}^{B\times m\times d}$, where m denotes the number of query tokens and d is the feature dimension. Since the relevance of different metadata levels varies across samples, three learnable router tokens $\text{RO} \in \mathbb{R}^{3\times d}$ are introduced to selectively aggregate the language sequence: $G = \text{CrossAttn}(Q = \text{RO}, K = \text{TF}, V = \text{TF}) \in \mathbb{R}^{B\times 3\times d}$. The routed features are finally flattened and projected into a compact text representation:

$$F_{\text{text}} = \text{MLP}_{\text{text}}(\text{Flatten}(G)) \in \mathbb{R}^{B\times d}. \tag{6}$$

3.5 FEATURE FUSION AND FORECASTING

The forecasting stage integrates the time series feature F_{time} with the visual and textual representations $F_{\text{img}}, F_{\text{text}}$. First, F_{img} and F_{text} are concatenated and projected into a shared multimodal context, denoted as F_{multi}, which is expanded along the variable dimension. Meanwhile, the last PL latent forecasting positions of F_{time} are projected as the variable temporal representation F_{temp}. Cross-attention aligns F_{temp} with F_{multi}, producing the aligned multimodal feature F_{align}. A learnable gate then combines F_{align} with the temporal prediction, followed by residual refinement to obtain the fused representation H. The forecasting output is generated as

$$\hat{Y} = \text{MLP}_{\text{pre}}(H) \in \mathbb{R}^{B\times PL\times C}. \tag{7}$$

Finally, $\hat{Y}$ is denormalized through RevIN to obtain Y', and the model is optimized by

$$\mathcal{L} = \mathcal{L}_{\text{MSE}}(Y', Y) + \mathcal{L}_{\text{TEFE}}. \tag{8}$$

Importantly, the future ground-truth sequence is used only for loss computation during training.

4 EXPERIMENTS

This section evaluates EviTime in terms of long- and short-term forecasting performance, hyperparameter sensitivity, ablation, robustness, and few-shot experiments. The source code and experimental data have been uploaded to the submission system for reproducibility.

4.1 EXPERIMENTAL SETTINGS

4.1.1 DATASETS

Experiments are conducted on 7 forecasting datasets: ETTh1, ETTh2, ETTm1, ETTm2, Weather, ECL, and Traffic. These datasets cover energy, weather, and traffic scenarios, with sampling intervals ranging from 10 minutes to 1 hour. Further details are shown in Appendix C.

4.1.2 SOTA BASELINES

EviTime is compared with 10 SOTA forecasting models, including (1) evidence-based models: TEFN (Zhan et al., 2025); (2) frequency-based models: TimeEmb (Xia et al., 2025); (3) linear models: DLinear (Zeng et al., 2023) and LightTS (Zhang et al., 2022); (4) Transformer-based variants: PatchTST (Nie et al., 2023), FEDformer (Zhou et al., 2022), Stationary (Liu et al., 2022), ETSformer (Woo et al., 2022), and Autoformer (Wu et al., 2021); and (5) 2D structured models: TimesNet (Wu et al., 2023). Further details are provided in Appendix D.

Table 1: Long-term forecasting results averaged over $PL \in \{96, 192, 336, 720\}$ across 7 datasets. Best and second-best results are marked in bold and underlined, respectively. Full results are provided in Appendix F.

Methods	Ours		TEFN		TimeEmb		DLinear		LightTS		PatchTST		FEDformer		Stationary		ETSformer		Autoformer		TimesNet	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.414**	0.430	0.441	0.429	0.425	**0.425**	0.422	0.437	0.491	0.479	0.469	0.454	0.440	0.460	0.570	0.537	0.542	0.510	0.496	0.487	0.458	0.450
ETTh2	**0.344**	0.392	0.380	0.402	0.362	**0.390**	0.431	0.446	0.602	0.543	0.387	0.407	0.437	0.449	0.526	0.516	0.439	0.452	0.450	0.459	0.414	0.427
ETTm1	**0.356**	**0.377**	0.403	0.398	0.368	0.384	0.357	0.378	0.435	0.437	0.387	0.400	0.448	0.452	0.481	0.456	0.429	0.425	0.588	0.517	0.400	0.406
ETTm2	**0.254**	0.313	0.286	0.327	0.265	**0.308**	0.267	0.333	0.409	0.436	0.281	0.326	0.305	0.349	0.306	0.347	0.293	0.342	0.327	0.371	0.291	0.333
Weather	**0.225**	**0.256**	0.260	0.283	0.237	0.262	0.248	0.300	0.261	0.312	0.259	0.281	0.309	0.360	0.288	0.314	0.271	0.334	0.338	0.382	0.259	0.287
ECL	**0.161**	**0.254**	0.215	0.292	0.167	0.260	0.166	0.263	0.229	0.329	0.205	0.290	0.214	0.327	0.193	0.296	0.208	0.323	0.227	0.338	0.192	0.295
Traffic	**0.413**	**0.281**	0.623	0.372	0.454	0.293	0.433	0.295	0.622	0.392	0.481	0.304	0.610	0.376	0.624	0.340	0.621	0.396	0.628	0.379	0.620	0.336

4.1.3 Experimental Computational Resources

All experiments are conducted on a high-performance workstation equipped with an Intel Core i7-14700 CPU, 64 GB of system memory, and a single 40 GB NVIDIA A100 GPU. The model is implemented in PyTorch using pretrained BLIP-2 as the vision-language backbone. Further implementation details are provided in Appendix E.

4.2 Long-Term Forecasting Performance

EviTime is evaluated against 10 SOTA baselines on 7 datasets, and Table 1 reports the average results over all prediction lengths.

As shown in Table 1, EviTime achieves the best MSE on all 7 datasets and the best MAE on 4 datasets. It also ranks second in MAE on ETTh2 and ETTm2, placing among the top 2 methods on 6 datasets. Compared with TimeEmb, the strongest overall baseline, EviTime obtains lower average MSE and MAE across the 7 datasets, demonstrating more consistent forecasting performance. The complete results for $PL \in \{96, 192, 336, 720\}$ are provided in Appendix F. To further examine its prediction behavior beyond aggregate metrics, detailed case studies on ETTm2 and ECL are provided in Appendix G.

4.3 Short-Term Forecasting Performance

EviTime is evaluated on three representative frequency subsets of the M4 short-term forecasting benchmark, including yearly, quarterly, and monthly series. Table 2 compares it with 5 representative baselines from different categories using SMAPE, MASE, and OWA.

As shown in Table 2, EviTime achieves the best results in 8 of the 9 frequency-specific settings, including all metrics on the quarterly and monthly subsets. In terms of average performance, it obtains the lowest SMAPE of 10.975 and OWA of 0.802, reducing them by 6.5% and 6.0% over the second-best results, respectively. Although PatchTST achieves the best average MASE, EviTime ranks second with 1.687. These results demonstrate its effectiveness in capturing recurring seasonal patterns.

Table 2: Short-term forecasting on M4 across yearly, quarterly, and monthly frequencies.

Methods		Ours	TEFN	TimeEmb	DLinear	PatchTST	TimesNet
Yea	SMAPE	**12.216**	14.023	13.316	16.965	13.477	15.378
	MASE	3.021	3.039	**3.012**	4.283	3.019	3.554
	OWA	**0.754**	0.823	0.767	1.058	0.792	0.918
Qua	SMAPE	**8.867**	11.597	9.318	12.145	10.380	10.465
	MASE	**1.123**	1.501	1.155	1.520	1.233	1.227
	OWA	**0.812**	1.089	0.904	1.106	0.921	0.923
Mon	SMAPE	**11.841**	13.176	12.584	13.514	12.959	13.513
	MASE	**0.916**	1.008	0.953	1.037	0.970	1.039
	OWA	**0.841**	0.928	0.887	0.956	0.905	0.957
Ave	SMAPE	**10.975**	12.932	11.739	13.639	12.059	12.880
	MASE	1.687	1.849	1.701	2.095	**1.623**	1.836
	OWA	**0.802**	0.947	0.853	1.051	0.869	0.955

4.4 Hyperparameter Sensitivity

This paper analyzes the sensitivity of EviTime to the input sequence length L and model dimension d_{model}. Experiments are conducted on 4 ETT subsets under different configurations, with MSE and MAE reported in Figure 3.

Impact of Input Sequence Length (L): The input sequence length L determines the size of the historical context available to the model and is varied in the range $\{96, 192, 336, 512, 672, 1024\}$. Figure 3 indicates that increasing the input sequence length from 96 to 512 generally leads to a consistent decrease in

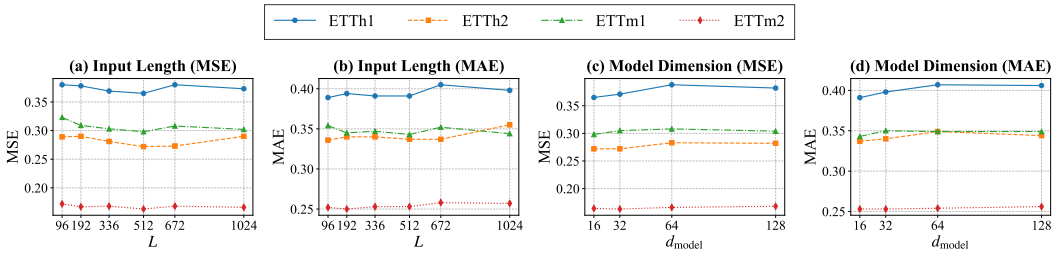

Figure 3: Hyperparameter sensitivity to the input sequence length $L \in \{96, 192, 336, 512, 672, 1024\}$ and model dimension $d_{\text{model}} \in \{16, 32, 64, 128\}$ on ETT datasets, evaluated using MSE and MAE.

Table 3: Ablation results of EviTime at the forecasting horizon $PL = 96$. The best results are highlighted in bold.

Datasets	ETTh1		ETTh2		ETTm1		ETTm2	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o DST	0.373	0.398	0.285	0.345	0.308	0.352	0.169	0.259
w/o GAF	0.378	0.402	0.291	0.349	0.315	0.358	0.172	0.262
w/o Metadata	0.392	0.415	0.324	0.368	0.338	0.375	0.178	0.268
Ours	**0.365**	**0.391**	**0.272**	**0.337**	**0.298**	**0.343**	**0.163**	**0.253**

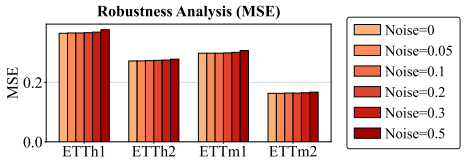

Figure 4: Robustness of EviTime on 4 ETT datasets under noise levels $\{0, 0.05, 0.1, 0.2, 0.3, 0.5\}$.

both MSE and MAE across most datasets. However, when the input length is further increased to 1024, the performance experiences a slight degradation. This phenomenon suggests that excessively long input sequences may introduce distant temporal noise or redundant information, which distracts the cross-modal attention mechanism and dilutes the extraction of task-relevant semantics. Therefore, $L = 512$ provides a favorable balance between temporal context and information redundancy.

Impact of Model Dimension (d_{model}): The model dimension d_{model} controls the representational capacity of the fused multimodal features. This paper evaluates the sensitivity of EviTime to d_{model} across the set $\{16, 32, 64, 128\}$. Figure 3 reveals that the proposed framework achieves its best performance at a relatively compact dimension of $d_{\text{model}} = 16$. As the feature dimension increases toward 128, both MSE and MAE exhibit an upward trend. Consequently, further increasing the hidden dimension introduces redundant degrees of freedom.

4.5 Ablation Experiments

To assess the contribution of each component, ablation studies are conducted on 4 ETT datasets using three variants: (1) **w/o DST:** Remove the evidential representation; (2) **w/o GAF:** Remove the GAF-based vision feature; (3) **w/o Metadata:** Remove the metadata-based language feature.

As shown in Table 3, removing any component degrades performance, confirming their complementary contributions. The w/o Metadata variant causes the largest decline, with MSE increasing by up to 19.1% on ETTh2, highlighting the importance of hierarchical metadata in providing semantic context for the pretrained VLM. Removing GAF also increases errors, indicating that the vision encoder benefits from two-dimensional structural patterns. Moreover, removing DST consistently degrades all metrics, confirming that evidence reasoning strengthens temporal representations despite the moderate decline. Overall, EviTime performs best when evidence modeling, visual patterns, and language semantics are jointly integrated.

4.6 Robustness Experiments

Real-world time series data are inevitably corrupted by sensor noise, transmission errors, and environmental interference. To evaluate the robustness of EviTime against such disturbances, this paper simulates noisy real-world conditions by injecting random perturbations into the input sequences. Specifically, the model's performance is evaluated on 4 ETT datasets under varying noise levels of $\{0, 0.05, 0.1, 0.2, 0.3, 0.5\}$, as shown in Figure 4.

Table 4: Few-shot forecasting results using 10% and 5% of the training data. The results are averaged over all forecasting horizons $PL \in \{96, 192, 336, 720\}$, and the best and second-best results under each training-data setting are marked in bold and underlined, respectively. For Traffic, $PL = 720$ is not reported. The complete horizon-specific results are provided in Appendix H.

Training Data	Datasets	EviTime		TEFN		TimeEmb		DLinear		PatchTST		TimesNet	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
10%	Weather	0.243	0.275	0.267	0.288	0.246	**0.272**	**0.241**	0.283	0.242	0.279	0.279	0.301
	ECL	**0.173**	**0.269**	0.222	0.298	0.178	0.270	0.180	0.280	0.180	0.273	0.323	0.392
	Traffic	0.425	0.300	0.630	0.376	0.456	**0.295**	0.434	0.305	**0.415**	0.296	0.773	0.438
5%	Weather	**0.250**	0.288	0.274	0.295	0.256	**0.281**	0.263	0.308	0.269	0.303	0.298	0.318
	ECL	0.194	0.286	0.229	0.306	0.188	0.283	**0.176**	**0.275**	0.181	0.277	0.402	0.453
	Traffic	0.470	0.334	0.663	0.393	0.487	0.335	0.450	0.317	**0.418**	**0.296**	0.867	0.493

Table 5: Zero-shot forecasting results. The results are averaged over all forecasting horizons $PL \in \{96, 192, 336, 720\}$, and the best and second-best results are marked in bold and underlined. The notation "source $\rightarrow$ target" indicates that the model is trained on the source dataset and directly evaluated on the target dataset. The complete results are provided in Appendix H.

Methods	Ours		DLinear		PatchTST		Autoformer		TimesNet	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1 $\rightarrow$ ETTh2	**0.349**	**0.390**	0.493	0.488	0.380	0.405	0.582	0.548	0.421	0.431
ETTh1 $\rightarrow$ ETTm2	**0.308**	**0.358**	0.415	0.452	0.314	0.360	0.457	0.483	0.327	0.361
ETTh2 $\rightarrow$ ETTh1	**0.495**	**0.484**	0.703	0.574	0.565	0.513	0.757	0.608	0.865	0.621
ETTh2 $\rightarrow$ ETTm2	**0.299**	**0.355**	0.328	0.386	0.325	0.365	0.366	0.411	0.342	0.376

Based on Figure 4, EviTime demonstrates strong stability across all tested noise levels. Even under severe perturbations at a noise level of 0.5, the degradation in forecasting accuracy remains moderate. On ETTh1, the MSE increases only marginally from 0.365 at a baseline noise level of 0 to 0.377 at the highest noise level 0.5. On ETTm2, the MSE rises slightly from 0.163 to 0.167.

4.7 Few-Shot and Zero-Shot Experiments

To evaluate generalization under limited supervision and distribution shifts, few-shot experiments are conducted on Weather, ECL, and Traffic using 10% and 5% of the training data, while zero-shot experiments directly transfer models between ETT datasets without fine-tuning. The reported results are averaged over $PL \in \{96, 192, 336, 720\}$, and the complete results are provided in Appendix H.

As shown in Tables 4 and 5, EviTime remains competitive under limited training data and zero-shot transfer. With 10% of the training data, it achieves the best MSE and MAE on ECL, while under the 5% setting, it obtains the best MSE and the second-best MAE on Weather, with larger degradation on ECL and Traffic. For zero-shot transfer, EviTime achieves the lowest MSE and MAE in all 4 source-to-target settings, reducing MSE by 8.2%, 1.9%, 12.4%, and 8.0% over the second-best PatchTST, respectively. These results indicate that EviTime retains effective forecasting capability under data scarcity and exhibits cross-dataset transferability across different sampling granularities and temporal distributions.

5 Conclusion

This paper proposes EviTime, a multimodal framework for time series forecasting that integrates evidential reasoning, visual modeling, and metadata-guided semantic learning. DST constructs evidence-aware temporal representations, three-channel GAF images capture temporal structures, and hierarchical metadata prompts provide contextual semantics. Despite these advantages, BLIP-2 and variable-wise GAF construction introduce additional computational and memory costs. Future work will investigate lightweight multimodal backbones, model compression, shared multivariate GAF representations, and sparse attention mechanisms to reduce redundant computation while preserving complementary structural information.

AI USE STATEMENT

In this work, we used generative AI tools to improve the readability and language quality of the manuscript. We did not use generative AI tools for other tasks requiring disclosure, and the remaining required-disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. AI-assisted suggestions, including grammar corrections and writing improvements, were manually reviewed and validated by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study does not involve human participants, human behavioral analysis, personally identifiable information, or sensitive personal data. All experimental evaluations are conducted on publicly available benchmark datasets for time series forecasting, including energy, weather, and traffic datasets. The benchmark datasets consist of aggregated numerical observations collected from sensors or public monitoring infrastructures and contain no information associated with individual identities, personal attributes, or private activities.

REPRODUCIBILITY STATEMENT

The reproducibility of EviTime is supported by comprehensive descriptions of the model architecture, training strategy, benchmark datasets, evaluation metrics, and experimental configurations provided in the main manuscript and supplementary materials. The methodology section presents the complete framework design, including TEFE, GVFE, and MLFE. The experimental evaluation is conducted on seven publicly available time series forecasting benchmarks, including ETTh1, ETTh2, ETTm1, ETTm2, Weather, ECL, and Traffic. The benchmark datasets cover multiple application scenarios, including energy management, weather monitoring, and traffic forecasting. Detailed dataset information, preprocessing procedures, forecasting settings, and evaluation protocols are provided to support consistent experimental reproduction. The implementation of EviTime is developed using the PyTorch framework with a pretrained vision-language backbone. The source code, including data preprocessing pipelines, model implementation, training scripts, and evaluation procedures, is publicly available at `https://anonymous.4open.science/r/EvidentialTime-CD2F`. The complete implementation is organized to enable reproduction of the reported experimental results.

REFERENCES

Alan Gabriel Paredes Cetina, Kaouther Benguessoum, Raoni Lourenco, and Sylvain Kubler. Counterfactual explainable AI (XAI) method for deep learning-based multivariate time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 17393–17400, 2026.

Wenxia Chang, Chao Zhang, Wentao Li, and Deyu Li. MIGC-CMamba: Cross-domain Mamba with multi-scale imaging and granular-ball computing for traffic flow prediction. In *Proceedings of the ACM Web Conference 2026*, pp. 7167–7177, 2026.

Xiwen Chen, Peijie Qiu, Wenhui Zhu, Huayu Li, Hao Wang, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. Sequence complementor: Complementing transformers for time series forecasting with learnable sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15913–15921, 2025.

Abdul Monaf Chowdhury, Rabeya Akter, and Safaeid Hossain Arib. T3Time: Tri-modal time series forecasting via adaptive multi-head alignment and residual fusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20597–20605, 2026.

A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2):325–339, 1967.

Yong Deng. Generalized evidence theory. *Applied Intelligence*, 43(3):530–543, 2015.

Wei Fan, Jingru Fei, Jindong Han, Jie Lian, Hangting Ye, Xiaozhuang Song, Xin Lv, Kun Yi, and Min Li. MedViA: Empowering medical time series classification with vision augmentation and multimodal fusion. *Information Fusion*, 127:103659, 2026a.

Xiaoxuan Fan, Jiaqi Sun, Xianjun Deng, Qiankun Zhang, Wei Xiang, Shenghao Liu, and Lingzhi Yi. Automated model selection for multivariate time series forecasting. In *Proceedings of the ACM Web Conference 2026*, pp. 5322–5333, 2026b.

Wenbo Gu, Ruizhe Ma, Yang Yu, and Zongmin Ma. FPSO-Time: A fuzzy time series forecasting method based on fuzzy particle swarm optimization and reprogrammed large language models. *IEEE Transactions on Fuzzy Systems*, 34(2):567–579, 2026.

Qi Guo, Baolin Zhao, Mingchen Song, and Guoqiang Zhong. A survey of deep learning for time series forecasting: Taxonomy, analysis and future directions. *IEEE Transactions on Knowledge and Data Engineering*, 38(5):2541–2560, 2026.

Fang He, Jiaqi Fan, Yang Deng, Xiaoyang Zhang, and Dan Wang. MetaCloze: a schema-guided automated building metadata model generation system via information extraction. *Knowledge-Based Systems*, 330:114596, 2025.

Zelin He, Sarah Alnegheimish, and Matthew Reimherr. Harnessing vision-language models for time series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21690–21698, 2026.

Shuning Jia, Baijun Song, Canming Ye, and Chun Yuan. M3Time: LLM-enhanced multi-modal, multi-scale, and multi-frequency multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22265–22273, 2026.

Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 5583–5594, 2021.

Xiaoyan Kui, Canwei Liu, Qinsong Li, Zhipeng Hu, Yangyang Shi, Weixin Si, and Beiji Zou. TFKAN: Time-frequency KAN for long-term time series forecasting. *Neurocomputing*, 701: 134558, 2026.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 19730–19742, 2023.

Rui Li, Chao Zhang, Deyu Li, Wentao Li, and Jianming Zhan. Improved evidential three-way decisions in incomplete multi-scale information systems. *International Journal of Approximate Reasoning*, 181:109417, 2025a.

Tianan Li, Xiangdong Huang, Jianmin Wang, Dongfang Mao, Yi Xu, and Jun Yuan. The design of Apache IoTDB distributed framework. *Scientia Sinica Informationis*, 50(5):621–636, 2020.

Zhe Li, Xiangfei Qiu, Peng Chen, Yihang Wang, Hanyin Cheng, Yang Shu, Jilin Hu, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, and Bin Yang. TSFM-Bench: A comprehensive and unified benchmark of foundation models for time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 5595–5606, 2025b.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *Proceedings of the International Conference on Learning Representations*, 2024a.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 32369–32399, 2024b.

Zhiding Liu, Mingyue Cheng, Guanhao Zhao, Jiqian Yang, Qi Liu, and Enhong Chen. Improving time series forecasting via instance-aware post-hoc revision. In *Advances in Neural Information Processing Systems*, volume 38, pp. 35604–35629, 2025.

Sisuo Lyu, Siru Zhong, Tiegang Chen, Weilin Ruan, Qingxiang Liu, Taiqiang Lv, Qingsong Wen, Raymond Chi-Wing Wong, and Yuxuan Liang. TS-Memory: Plug-and-play memory for time series foundation models. *arXiv preprint arXiv:2602.11550*, 2026.

Shichao Ma, Zhengyang Zhou, Qihe Huang, Binwu Wang, and Yang Wang. QuiZSF: A retrieval-augmented framework for zero-shot time series forecasting. In *Proceedings of the ACM Web Conference 2026*, pp. 1947–1958, 2026.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *Proceedings of the International Conference on Learning Representations*, 2023.

Sehyuk Park, Caren Han, and Hyunsuk Chung. MulTiCast: A multimodal time series forecasting system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 41655–41657, 2026.

Jianyang Qin, Chaoyang Li, Jinhao Cui, Lingzhi Wang, Zhao Liu, and Qing Liao. Bridging time and linguistics: LLMs as time series analyzer through symbolization and segmentation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 124618–124650, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021.

Jenn Riley. *Understanding Metadata: What is Metadata, and What is it For?: A Primer*. National Information Standards Organization (NISO), Baltimore, MD, 2017.

Andrey Savchenko and Oleg Kachan. HN-MVTS: Hypernetwork-based multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25200–25208, 2026.

Glenn Shafer. *A mathematical theory of evidence*. Princeton University Press, Princeton, NJ, 1976.

Siroos Shahriari, Scott A Sisson, and Taha Rashidi. Modelling time series with temporal and spatial correlations in transport planning using hierarchical ARIMA-copula model: A bayesian approach. *Expert Systems with Applications*, 274:126977, 2025.

Han Shi, Yangyang Wu, Daozhan Pan, Jing Li, Guoqiang Qi, Xiaoye Miao, Jianwei Yin, and Gang Yu. Effective and efficient incomplete multivariate time series forecasting. *Expert Systems with Applications*, 323:132412, 2026.

Yuxuan Shu and Vasileios Lampos. Sonnet: Spectral operator neural network for multivariable time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 25419–25427, 2026.

Anna Wang, Chao Zhang, Mingwei Lin, Junbo Zhang, Zeshui Xu, Wentao Li, Pengfei Zhang, and Oscar Castillo. Towards vision-spatiotemporal fusion in traffic forecasting: A survey on cross-modal alignment. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 8032–8040, 2026a.

Bingjie Wang, Chao Zhang, Wentao Li, and Deyu Li. MF3: Multimodal federated learning with dual-path Mamba-Transformer for metro flow prediction. In *Proceedings of the ACM Web Conference 2026*, pp. 5451–5462, 2026b.

Guohong Wang, Xianhan Tan, Zengming Lin, Binli Luo, Shangjian Zhong, and Kele Xu. FreqEvo: Enhancing time series forecasting with multi-level frequency domain feature extraction. *IEEE Transactions on Knowledge and Data Engineering*, 38(2):1099–1113, 2026c.

Jieting Wang, Huimei Shi, Feijiang Li, and Xiaolei Shang. Beyond MSE: Ordinal cross-entropy for probabilistic time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26269–26276, 2026d.

Ning Wang, Haiyan Wang, Jian Tan, Panpan Feng, Shihua Li, Zongmin Wang, and Bing Zhou. ECG-text multi-modal learning for zero-shot detection via time-frequency alignment and medical prompt learning. *Expert Systems with Applications*, 307:131064, 2026e.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Chen Wang, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*, 2024.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. ETSformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *Proceedings of the International Conference on Learning Representations*, 2023.

Mingyuan Xia, Chunxu Zhang, Zijian Zhang, Hao Miao, Qidong Liu, Yuanshao Zhu, and Bo Yang. TimeEmb: A lightweight static-dynamic disentanglement framework for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 38, pp. 76227–76258, 2025.

Jiexia Ye, Yongzi Yu, Weiqi Zhang, Le Wang, Jia Li, and Fugee Tsung. Empowering time series analysis with foundation models: A comprehensive survey. *Information Fusion*, 137:104601, 2027.

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

Tianxiang Zhan, Yuanpeng He, Yong Deng, Zhen Li, Wenjie Du, and Qingsong Wen. Time evidence fusion network: Multi-source view in long-term time series forecasting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11220–11233, 2025.

Boya Zhang, Shuaijie Yin, Huiwen Zhu, and Xing He. FreqCycle: A multi-scale time-frequency analysis method for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 28159–28167, 2026a.

Hongsheng Zhang, Zhong Ji, Jingren Liu, Yanwei Pang, and Jungong Han. Multi-stage knowledge integration of vision-language models for continual learning. *IEEE Transactions on Image Processing*, 35:615–628, 2026b.

Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented MLP structures. *arXiv preprint arXiv:2207.01186*, 2022.

Yan Zhang, Yihong Song, Manzhou Li, Qiushi Li, Qi Li, and Ju Ren. PeriNet: Periodic deep learning framework for modality-agnostic privacy preserving. In *Proceedings of the ACM Web Conference 2026*, pp. 3426–3437, 2026c.

Siru Zhong, Weilin Ruan, Ming Jin, Huan Li, Qingsong Wen, and Yuxuan Liang. Time-VLM: Exploring multimodal vision-language models for augmented time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 78478–78497, 2025.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 27268–27286, 2022.

A THEORETICAL SUPPORT

This appendix provides the theoretical foundations underlying the proposed EviTime framework. Appendix A.1 introduces the DST for uncertainty reasoning. Appendix A.2 presents the GAF for time series-to-image transformation. Appendix A.3 describes the BLIP-2 vision-language model architecture.

A.1 DST

DST is a mathematical framework for handling uncertainty and evidential reasoning. DST is widely applied in multi-source information fusion, decision support, and uncertainty modeling. Unlike traditional Bayesian probability, which requires complete prior probabilities and assigns probabilities only to single mutually exclusive events, DST can assign probabilities to sets of events.

Frame of Discernment Let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ be a finite and mutually exhaustive set of hypotheses, referred to as the frame of discernment. All possible propositions are represented by the power set of Θ, denoted as 2^Θ. The frame of discernment defines the universe of all possible outcomes for a given problem. The power set 2^Θ includes every possible combination of these outcomes. For example, if a medical diagnosis frame is $\{$Disease X, Disease Y$\}$, the power set includes Disease X, Disease Y, Disease X or Disease Y, and the empty set. This allows the model to say "the evidence points to either X or Y, but I don't know which".

BPA A function $m : 2^\Theta \to [0, 1]$ is called a BPA if it satisfies the following conditions:

$$m(\emptyset) = 0, \sum_{A \subseteq \Theta} m(A) = 1. \tag{9}$$

Here, $m(A)$ represents the exact belief mass assigned strictly to proposition A, and any subset A where $m(A) > 0$ is called a focal element. The BPA is not the same as a classical probability. If $m(A) = 0.6$, it means we have exactly 60% belief dedicated specifically to set A as a whole, not distributed among its individual subsets. The remaining belief could be assigned to the entire frame Θ to represent "unknown".

Belief and Plausibility Functions The belief function represents the total degree of support for proposition A:

$$Bel(A) = \sum_{B \subseteq A} m(B). \tag{10}$$

The plausibility function represents the degree to which A is not contradicted by the evidence:

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) = 1 - Bel(\bar{A}). \tag{11}$$

These two functions create a probability interval $[Bel(A), Pl(A)]$ that bounds the true probability of a hypothesis. $Bel(A)$ is the lower bound: the sum of all evidence that strictly guarantees A. $Pl(A)$ is the upper bound: the sum of all evidence that could potentially support A, i.e., evidence that does not explicitly contradict A. The gap between them, $Pl(A) - Bel(A)$, quantifies the exact degree of uncertainty regarding proposition A.

Dempster's Rule of Combination Let m_1 and m_2 be two independent BPAs over the same frame of discernment. Their combination is calculated as:

$$(m_1 \oplus m_2)(A) = \frac{1}{1 - K} \sum_{B \cap C = A} m_1(B) m_2(C), \forall A \neq \emptyset, \tag{12}$$

where the conflict coefficient K is:

$$K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C). \tag{13}$$

If $K = 1$, the two pieces of evidence are in complete conflict and cannot be combined. This rule is the primary engine of DST, used to fuse independent pieces of evidence to reach a stronger

conclusion. The factor $\frac{1}{1-K}$ is a normalization constant that ensures the new mass assignments still sum to 1 by redistributing any mass assigned to the impossible empty set. While highly effective, researchers must be careful: if K is very close to 1, the rule can sometimes yield counterintuitive results.

Deng Entropy Deng Entropy is used to quantify the uncertainty of a BPA:

$$H_d(m) = -\sum_{A \subseteq \Theta} m(A) \log_2 \frac{m(A)}{2^{|A|} - 1}, \tag{14}$$

where $|A|$ denotes the cardinality of the focal element A. A higher entropy value indicates a greater degree of uncertainty in the evidence. Deng entropy is a generalization of traditional Shannon entropy designed specifically for the DST framework. Unlike Shannon entropy, which only measures probabilistic uncertainty, Deng entropy captures two dimensions of uncertainty:

(1) Discord: Uncertainty caused by conflicting evidence.

(2) Non-specificity: Uncertainty caused by the lack of precision in the evidence.

This makes it highly robust for evaluating the quality and reliability of evidence before feeding it into downstream models.

A.2 GAF

The GAF is a transformation technique that encodes one-dimensional time series into two-dimensional image representations. By mapping each time step into a polar coordinate system and leveraging trigonometric relationships, GAF captures temporal dependencies and pairwise correlations in a spatially structured form. This enables the application of mature vision models to time series analysis.

Traditional time series models operate in the temporal or frequency domain, where capturing long-range dependencies and structural patterns remains challenging. In contrast, visual models excel at extracting spatial features and recognizing patterns. GAF bridges this gap by transforming temporal dynamics into image-like structures, allowing time series to benefit from advances in computer vision.

Normalization Given a time series $X = \{x_1, x_2, \ldots, x_L\}$, it is first normalized into the interval $[-1, 1]$:

$$\tilde{x}_i = \frac{2(x_i - \min(X))}{\max(X) - \min(X)} - 1, \tag{15}$$

where $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_L\}$ ensures valid input for the subsequent angular transformation.

Polar Coordinate Transformation Each normalized value $\tilde{x}_i$ is mapped into polar coordinates (ϕ_i, r_i):

$$\phi_i = \arccos(\tilde{x}_i), \quad r_i = \frac{i}{L}, \tag{16}$$

where $\phi_i \in [0, \pi]$ encodes the magnitude information, and $r_i \in (0, 1]$ preserves temporal ordering. This mapping is bijective, ensuring no information loss.

Gramian Matrix Construction After polar mapping, GAF constructs an $L \times L$ Gramian matrix, where each entry encodes the trigonometric relationship between two time points. Two variants are commonly used, depending on the trigonometric function.

- **GASF:** GASF is defined using the cosine of the sum of angles:

$$\text{GASF}_{i,j} = \cos(\phi_i + \phi_j). \tag{17}$$

  Using the identity $\cos(\phi_i + \phi_j) = \cos\phi_i \cos\phi_j - \sin\phi_i \sin\phi_j$, together with $\cos\phi_i = \tilde{x}_i$ and $\sin\phi_i = \sqrt{1 - \tilde{x}_i^2}$, we obtain:

$$\text{GASF}_{i,j} = \tilde{x}_i \tilde{x}_j - \sqrt{1 - \tilde{x}_i^2}\sqrt{1 - \tilde{x}_j^2}, \tag{18}$$

which captures the similarity or co-movement between two time points. GASF measures the similarity between two time points. When both $\tilde{x}_i$ and $\tilde{x}_j$ are close to 1, $\mathrm{GASF}_{i,j} \approx 1$. Similarly, when both values are close to -1, $\mathrm{GASF}_{i,j} \approx 1$. In contrast, when one value is close to 1 and the other is close to -1, $\mathrm{GASF}_{i,j} = -1$. Therefore, larger GASF values indicate stronger directional alignment between the two time points.

- **GADF:** GADF is defined using the sine of the difference of angles:

$$\mathrm{GADF}_{i,j} = \sin(\phi_i - \phi_j). \tag{19}$$

Using the identity $\sin(\phi_i - \phi_j) = \sin\phi_i \cos\phi_j - \cos\phi_i \sin\phi_j$, we obtain:

$$\mathrm{GADF}_{i,j} = \tilde{x}_j\sqrt{1-\tilde{x}_i^2} - \tilde{x}_i\sqrt{1-\tilde{x}_j^2}, \tag{20}$$

which emphasizes directional differences and local temporal variations. GADF captures the dynamic difference between two time points. Unlike GASF, GADF is anti-symmetric, i.e., $\mathrm{GADF}_{i,j} = -\mathrm{GADF}_{j,i}$. This property makes GADF particularly sensitive to local temporal variations and directional changes.

Properties and Interpretation GASF produces a symmetric matrix encoding global similarity patterns, while GADF yields an antisymmetric matrix sensitive to temporal changes. The diagonal elements of GASF satisfy:

$$\mathrm{GASF}_{i,i} = \cos(2\phi_i) = 2\tilde{x}_i^2 - 1, \tag{21}$$

whereas $\mathrm{GADF}_{i,i} = 0$.

From a geometric perspective, ϕ_i represents the angular position on the unit circle. GASF measures angular co-alignment, while GADF captures angular differences, enabling complementary representations of temporal structure.

GAF preserves the temporal ordering of the original time series by implicitly encoding sequential information through both the radial component r_i and the spatial arrangement of matrix indices. At the same time, it captures pairwise correlations between all time points, enabling the modeling of long-range dependencies. Furthermore, the resulting image-like representation can be directly leveraged by standard computer vision architectures, such as convolutional neural networks and Vision Transformers, facilitating effective feature extraction and multimodal integration.

A.3 BLIP-2 Vision-Language Model

BLIP-2 is an efficient vision-language pre-training framework designed to reduce the computational cost and inference latency of traditional large-scale VLMs. By introducing a lightweight Querying Transformer (Q-Former) as an intermediate bridge, BLIP-2 connects a frozen image encoder with a frozen Large Language Model (LLM), enabling efficient cross-modal alignment and parameter-efficient fine-tuning.

Model Architecture BLIP-2 consists of three core components:

- **Frozen image encoder:** Pre-trained vision encoders such as Vision Transformer (ViT) are employed to extract high-level semantic features from images. The parameters of this module remain fixed during training.
- **Frozen language model:** Pre-trained LLMs such as OPT, Flan-T5, or Flan-UL2 are used for text understanding and generation, and are kept frozen.
- **Trainable Q-Former:** The only trainable component, which uses cross-attention to extract task-relevant visual features and transform them into representations compatible with the LLM.

Q-Former Structure The Q-Former is a lightweight Transformer composed of two sub-modules:

- **Vision-text encoder:** Takes image features and text embeddings as input for multimodal representation learning.

- **Vision-text decoder:** Applies a causal attention mask for vision and text generation.

To control computational complexity, the Q-Former introduces n learnable query tokens: $Q \in \mathbb{R}^{n \times d}$, which selectively attend to image features instead of passing all visual tokens to the LLM.

Training Objectives BLIP-2 is pre-trained using three objective functions:

- **Image-text contrastive loss:** $\mathcal{L}_1 = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\text{sim}(Z_i^I, Z_i^T)/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(Z_i^I, Z_j^T)/\tau)}$, where B denotes the batch size; Z_i^I and Z_i^T represent the normalized image and text embeddings of the i-th pair, respectively; $\text{sim}(\cdot, \cdot)$ denotes cosine similarity; τ is a temperature parameter controlling the concentration level of the distribution; and j indexes all samples within the batch.
- **Image-text matching loss:** $\mathcal{L}_2 = -\mathbb{E}_{(I,T)}\left[y \log p(I,T) + (1-y)\log(1-p(I,T))\right]$, where (I, T) denotes an image-text pair sampled from the data distribution; $p(I, T)$ is the predicted probability that the pair is semantically aligned; $y \in \{0, 1\}$ is the ground-truth label (1 for matched pairs and 0 for mismatched pairs); and $\mathbb{E}[\cdot]$ denotes expectation over the data distribution.
- **Image-conditioned language modeling loss:** $\mathcal{L}_3 = -\sum_{t=1}^{|T|} \log P(t_t \mid t_{<t}, I)$, where $T = \{t_1, t_2, \ldots, t_{|T|}\}$ denotes the target text sequence; $|T|$ is the sequence length; t_t is the token at time step t; $t_{<t}$ represents all preceding tokens $\{t_1, t_2, \ldots, t_{t-1}\}$; I denotes the input image; and $P(\cdot)$ is the conditional probability modeled by the language decoder.

BLIP-2 achieves high parameter efficiency by training only the lightweight Q-Former while keeping both the image encoder and LLM frozen, thereby significantly reducing computational cost. Meanwhile, the cross-attention mechanism enables effective alignment between visual and textual modalities, facilitating seamless cross-modal interaction. Benefiting from large-scale pre-training, the model exhibits strong zero-shot and few-shot generalization capabilities across diverse multimodal tasks. In addition, the flexible design allows easy integration with various mainstream LLMs, such as OPT and Flan-T5, making it adaptable to a wide range of downstream applications.

B Algorithm Details

This paper provides the detailed algorithmic workflow of EviTime, followed by a comprehensive theoretical analysis of time and space complexities.

B.1 Algorithm Analysis

The overall procedure of EviTime is summarized in Algorithm 1.

B.2 Time Complexity Analysis

The time complexity of EviTime is primarily determined by three parallel encoding branches and the computationally intensive VLM backbone. For rigor, the key notations are defined as follows:

- B: Batch size.
- L: Input sequence length.
- PL: Prediction horizon.
- C: Number of variables in the multivariate time series.
- K: Number of focal elements in the evidential theory.
- d: Hidden dimension of the kernel-based projections.
- d_{VLM}: Hidden dimension of the pre-trained VLM backbone.
- N_v: Number of visual patches processed by the vision encoder.
- N_t: Number of text tokens processed by the text encoder.

Algorithm 1 Training Procedure of EviTime

Input: Multivariate time series sequence $X \in \mathbb{R}^{B\times L\times C}$, where B is batch size, L is sequence length, and C is the number of variables; Multi-granular metadata prompt $\mathcal{P}$; Prediction horizon PL.

Output: Forecasted future sequence $Y' \in \mathbb{R}^{B\times PL\times C}$.

Initialize pre-trained VLM backbone BLIP-2, and learnable parameters for TEFE, GVFE, MLFE, and fusion layers.
while not converged **do**
  Preprocessing
  Normalize input X via RevIN to obtain $\hat{X}$.
  Apply linear projection and add sinusoidal positional encoding to obtain X_{pe}.
  Stage I: TEFE
  Generate nonlinear BPA: $M_T, M_V \leftarrow \text{MLPs}(X_{pe})$.
  Calculate evidence consistency EC_T and EC_V, and evidence strength ES_T and ES_V.
  Compute adaptive weights via MLP and obtain weighted evidence: R_T, R_V.
  Fuse via Deng entropy H_T and H_V, and learnable gate g: $R_{time} = g \odot R_T + (1-g) \odot R_V$.
  Extract variable and temporal dependencies via Kernel projections: F_T, F_V.
  Fuse temporal and variable dependencies to obtain E_{time}, then add residual evidence feature to get uncertainty-aware temporal feature: $F_{time} = R_{time} + E_{time}$.
  Stage II: GVFE
  for each variable $v \in \{1, 2, \ldots, C\}$ **do**
    Transform 1D sequence into polar coordinates ϕ.
    Compute Gramian Angular Summation/Difference Fields: $GASF_v, GADF_v$.
  end for
  Stack aggregated GASF, GADF, and original amplitude to form 3-channel image $I_{GAF} \in \mathbb{R}^{B\times 3\times h\times w}$.
  Extract visual feature via VLM: $F_{img} \leftarrow \text{AvgPool}(\text{VLM}_{vision}(I_{GAF}))$.
  Stage III: MLFE
  Construct hierarchical prompt $\mathcal{P}$ by concatenating dataset-, task-, and sample-level metadata.
  Extract language feature via VLM and metadata router: $F_{text} \leftarrow \text{Router}(\text{VLM}_{text}(\mathcal{P}))$.
  Stage IV: Multimodal Fusion and Forecasting
  Construct shared multimodal context: $F_{multi} \leftarrow \text{MLP}_{fuse}([F_{img} \| F_{text}])$.
  Select the last PL latent forecasting positions $F_{\text{time}}[:, L : L + PL, :]$ and project them to obtain F_{temp}.
  Perform cross-modal attention $H \leftarrow \text{CrossAttn}(F_{temp}, F_{multi})$.
  Predict $\hat{Y} \leftarrow \text{MLP}_{pre}(H)$ and denormalize via RevIN to obtain final prediction Y'.
  Optimization
  Compute reconstruction loss $\mathcal{L}_{MSE}(Y', Y)$ and auxiliary TEFE loss $\mathcal{L}_{TEFE}$.
  Update network parameters by minimizing the total objective $\mathcal{L} = \mathcal{L}_{MSE} + \mathcal{L}_{TEFE}$.
end while

The overall computational complexity is decomposed according to the core architectural modules:

- **TEFE:** The nonlinear BPA generation is implemented via MLPs, exhibiting linear scaling with respect to both sequence length and variable dimension. The corresponding complexity is bounded by $\mathcal{O}(B \cdot (L + PL) \cdot C \cdot K)$. In addition, the kernel-based temporal and variable projections incur a complexity of $\mathcal{O}(B \cdot (L + PL) \cdot C \cdot d)$.
- **GVFE:** The transformation from 1D time series to 2D topological manifolds requires pairwise trigonometric computations across time steps. For C variables, this induces a quadratic complexity with respect to the sequence length, i.e., $\mathcal{O}(B \cdot C \cdot L^2)$. Subsequently, the VLM-based vision encoder processes the resulting multi-channel images with complexity $\mathcal{O}(B \cdot N_v^2 \cdot d_{\text{VLM}})$.
- **MLFE:** The text encoder processes multi-granular metadata prompts. Due to the self-attention mechanism, the computational complexity follows the standard Transformer formulation, i.e., $\mathcal{O}(B \cdot N_t^2 \cdot d_{\text{VLM}})$.

Overall Time Complexity Aggregating the above components, the overall time complexity is asymptotically dominated by the GAF construction and the VLM encoders:

$$\mathcal{O}\left(B \cdot C \cdot L^2 + B \cdot d_{\mathrm{VLM}}(N_v^2 + N_t^2)\right). \tag{22}$$

B.3 Space Complexity Analysis

The space complexity, or memory footprint, of EviTime is primarily dominated by two factors: the static parameters of the pre-trained foundational model and the intermediate topological representations generated during modality transformation. The memory overhead of each key component is detailed as follows:

- **Static model parameters:** The incorporation of the pre-trained BLIP-2 architecture introduces a substantial parameter space, requiring $\mathcal{O}(|\theta_{\mathrm{VLM}}|)$ static memory, typically in the billion-parameter regime, where θ_{VLM} denotes the set of all learnable parameters of VLMs.
- **TEFE:** The generation of BPA, specifically M_T and M_V, requires storing intermediate activation tensors of shape $\mathbb{R}^{B\times(L+PL)\times C\times K}$. Accordingly, the memory complexity allocated for evidence reasoning is bounded by $\mathcal{O}(B \cdot (L + PL) \cdot C \cdot K)$.
- **GVFE:** To preserve cross-variable independence prior to the VLM encoding phase, channel-wise GAF representations are computed independently. This operation generates intermediate activation matrices of size $\mathbb{R}^{B\times C\times L\times L}$ before channel aggregation. Therefore, the memory complexity for this visual transformation is bounded by $\mathcal{O}(B \cdot C \cdot L^2)$.
- **MLFE:** The MLFE module processes hierarchical text prompts through the text encoder of VLMs. This procedure involves storing tokenized embeddings and intermediate activation states of shape $\mathbb{R}^{B\times N_t\times d_{\mathrm{VLM}}}$. Furthermore, the metadata router utilizes three learnable query tokens to aggregate semantic features, introducing a negligible spatial overhead of $\mathcal{O}(B \cdot 3 \cdot d_{\mathrm{VLM}})$. Consequently, the overall memory footprint for the language feature extraction is linearly bounded by $\mathcal{O}(B \cdot N_t \cdot d_{\mathrm{VLM}})$.

C Supplementary Dataset Descriptions

In this paper, eight widely used real-world benchmark datasets are employed for time series forecasting, including ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, Traffic, and M4. The first seven datasets cover sampling intervals ranging from 10 minutes to 1 hour and represent application domains such as energy, meteorology, and transportation. In contrast, M4 is a large-scale benchmark containing univariate time series with multiple sampling frequencies and predefined forecast horizons, and is commonly used to evaluate short-term forecasting performance. The sampling characteristics and data-splitting strategies of the above-mentioned datasets are described below.

C.1 ETT

The Electricity Transformer Temperature (ETT) dataset consists of four subsets collected from two electricity transformers. ETTh1 and ETTh2 are sampled hourly, whereas ETTm1 and ETTm2 are sampled every 15 minutes. Each subset contains seven variables, including six power-load-related variables and the oil temperature of the corresponding transformer. The data cover the period from July 2016 to July 2018. Following the standard protocol adopted by Informer, the datasets are chronologically divided as follows:

- **Training set**: the first 12 months.
- **Validation set**: the subsequent 4 months.
- **Test set**: the following 4 months.

C.2 Electricity

The Electricity dataset contains the hourly electricity consumption, measured in kWh, of 321 customers. The benchmark version covers the period from 2012 to 2014 and contains 321 variables.

The dataset exhibits multiple seasonal patterns, including daily, weekly, and annual variations. The dataset is chronologically divided as follows:

- **Training set**: the first 70% of the observations.
- **Validation set**: the subsequent 10% of the observations.
- **Test set**: the final 20% of the observations.

C.3 TRAFFIC

The Traffic dataset is widely used for traffic forecasting. The dataset contains hourly road occupancy rates collected by 862 sensors on freeways in the San Francisco Bay Area from 2015 to 2016, resulting in 862 variables. The dataset is chronologically divided as follows:

- **Training set**: the first 70% of the observations.
- **Validation set**: the subsequent 10% of the observations.
- **Test set**: the final 20% of the observations.

C.4 WEATHER

The Weather dataset contains 21 meteorological variables recorded in Germany throughout 2020. The observations are collected at 10-minute intervals, providing high-resolution measurements of meteorological conditions. The dataset is chronologically divided as follows:

- **Training set**: the first 70% of the observations.
- **Validation set**: the subsequent 10% of the observations.
- **Test set**: the final 20% of the observations.

C.5 M4

The M4 dataset was introduced in the M4 forecasting competition and contains 100,000 univariate time series collected from diverse real-world domains, including finance, industry, demography, macroeconomics, microeconomics, and other application areas. The time series are organized into six frequency groups: yearly, quarterly, monthly, weekly, daily, and hourly. This diversity makes M4 a comprehensive benchmark for evaluating forecasting models across different temporal granularities. Unlike the other datasets, M4 does not adopt a unified chronological train/validation/test ratio. Instead, M4 provides an official training set and a test set with a frequency-specific forecast horizon. The evaluation protocol is defined as follows:

- **Training set**: all observations available before the forecast origin for each time series.
- **Test set**: the observations following the forecast origin, whose length is determined by the predefined forecast horizon H of each frequency group.

D SUPPLEMENTARY BASELINE DESCRIPTIONS

To ensure a comprehensive and fair evaluation, EviTime is compared with ten state-of-the-art (SOTA) time series forecasting models. These baselines cover five representative model families, as described below.

D.1 EVIDENCE-BASED MODELS

- **TEFN** Zhan et al. (2025): A time evidence fusion network addresses the joint requirements of forecasting accuracy and efficiency from an information fusion perspective. A basic probability assignment module models uncertainty along the temporal and channel dimensions, while a multisource fusion mechanism integrates the

resulting evidence for prediction. The lightweight architecture provides an interpretable forecasting framework with low computational complexity and stable performance. The detailed implementation is available at `https://github.com/ztxtech/Time-Evidence-Fusion-Network`.

D.2 Frequency-Based Models

- **TimeEmb** Xia et al. (2025): A lightweight static dynamic decomposition framework addresses temporal nonstationarity by separating persistent patterns from changing temporal variations. A global embedding module captures time invariant representations, while frequency domain filtering models time varying components through full spectrum analysis. The modular design supports efficient forecasting and can be integrated into existing time series models with limited computational overhead.

D.3 Linear and MLP-Based Models

- **DLinear** Zeng et al. (2023): A decomposition based linear model addresses long term time series forecasting by directly learning temporal relationships without self attention. Moving average decomposition separates each series into trend and residual components, which are projected independently through linear layers. The simple architecture preserves temporal ordering while avoiding the information loss and computational complexity associated with permutation invariant attention.
- **LightTS** Zhang et al. (2022): A lightweight multivariate time series forecasting framework that reduces computational cost through light sampling-oriented MLP structures. The model employs interval sampling and continuous sampling to capture long- and short-range temporal dependencies, respectively, providing efficient forecasting with a compact architecture.

D.4 Transformer-Based Models

- **PatchTST** Nie et al. (2023): A channel independent patch time series Transformer improves multivariate forecasting by representing each univariate channel as a sequence of local patches. Subseries patches preserve local temporal information, reduce attention computation and memory consumption, and enable the model to utilize longer historical contexts. Shared embedding and Transformer parameters across channels further support efficient forecasting and transferable self supervised representations.
- **FEDformer** Zhou et al. (2022): A frequency enhanced decomposed Transformer combines seasonal trend decomposition with frequency domain learning to address global pattern modeling and computational cost in long term forecasting. The decomposition component captures overall temporal profiles, while the Transformer models detailed structures. Sparse frequency representations further reduce attention complexity and support efficient long range prediction.
- **Stationary** Liu et al. (2022): A nonstationary Transformer framework addresses changing temporal distributions by balancing series predictability with model capacity. Series stationarization normalizes input statistics and restores output statistics, while destationary attention reintroduces intrinsic nonstationary information into temporal dependencies. The two modules mitigate overstationarization and help different series retain distinguishable attention patterns.
- **ETSformer** Woo et al. (2022): An exponential smoothing Transformer architecture improves long term time series forecasting through interpretable decomposition and efficient temporal modeling. Exponential smoothing attention and frequency attention replace conventional self attention to capture evolving trends and periodic patterns. Modular decomposition blocks further separate time series into level, growth, and seasonal components.
- **Autoformer** Wu et al. (2021): An Autoformer decomposition architecture addresses long term forecasting by combining progressive series decomposition with an autocorrelation mechanism. Series decomposition serves as an internal model block to progressively separate complex temporal components, while the autocorrelation mechanism discovers periodic dependencies and aggregates representations at the subseries level. The design im-

proves long range information utilization and reduces the efficiency limitations of pointwise self attention.

D.5 2D-Structured Models

- **TimesNet** Wu et al. (2023): A temporal two dimensional variation modeling network represents complex time series patterns by transforming one dimensional sequences according to multiple dominant periods. The transformed tensors organize intraperiod and interperiod variations along different dimensions, while a parameter efficient inception block extracts the resulting temporal structures. Adaptive period discovery supports a general backbone for forecasting, imputation, classification, and anomaly detection.

E Computational Resource Details

To facilitate reproducibility, this section describes the computational environment, model configuration, and training protocol used in the experiments.

E.1 Computational Environment

All experiments are conducted on a workstation equipped with an Intel Core i7-14700 CPU, 64 GB of system memory, and a single NVIDIA A100 GPU with 40 GB of memory. The proposed framework is implemented in PyTorch. Automatic mixed-precision training is enabled using `torch.cuda.amp` to reduce GPU memory consumption and accelerate training.

E.2 Vision-Language Backbone

A pretrained BLIP-2 model is adopted as the vision-language backbone, and the maximum prompt length is set to 1024 tokens. The hidden dimension of the language-model component is 2560.

The Q-Former converts the visual features extracted by the image encoder into a fixed number of query representations. A customized cross-attention module, referred to as the metadata router, is further employed to integrate the visual representations with variable-length textual metadata. The resulting multimodal features are projected into fixed-dimensional semantic embeddings and subsequently passed to the forecasting head.

E.3 Warmup and Early Stopping

A linear warmup strategy is applied to the metadata router during the first 10% of the training steps. The model's learning rate is gradually increased from zero to 10^{-6}. The maximum number of training epochs is set to 30. Early stopping is performed according to the validation MSE, with a patience of five epochs. The checkpoint achieving the lowest validation MSE is retained for final evaluation.

F Supplementary Long-Term Forecasting Results

This section reports the complete horizon-specific long-term forecasting results that are summarized in the main paper. EviTime is compared with ten representative forecasting baselines on seven real-world datasets, including ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity (ECL), and Traffic. The prediction length is selected from $PL \in \{96, 192, 336, 720\}$ to evaluate forecasting performance under different temporal horizons. Mean Squared Error (MSE) and Mean Absolute Error (MAE) are adopted as evaluation metrics, where lower values indicate better forecasting accuracy. For each dataset, the **Avg** row reports the arithmetic mean over the four prediction horizons.

As shown in Table 6, EviTime achieves the best or tied-best MSE in 32 of the 35 reported settings, where the 35 settings consist of four prediction horizons and one average result for each of the seven datasets. When only the horizon-specific results are considered, EviTime obtains the best or tied-best MSE in 25 of the 28 settings and ranks second in the remaining three settings. Moreover, EviTime

Table 6: Detailed long-term forecasting results of EviTime and ten baseline models on seven real-world datasets under prediction lengths $PL \in \{96, 192, 336, 720\}$. The **Avg** row reports the average performance across the four prediction horizons. Lower MSE and MAE values indicate better forecasting performance. The best and second-best results are marked in bold and underlined, respectively. The final row reports the number of best or tied-best results across all horizon-specific and average settings.

Methods		Ours		TEFN		TimeEmb		DLinear		LightTS		PatchTST		FEDformer		Stationary		ETSformer		Autoformer		TimesNet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.365**	0.391	0.383	0.391	0.366	**0.387**	0.375	0.399	0.424	0.432	0.414	0.419	0.376	0.419	0.513	0.491	0.494	0.479	0.449	0.459	0.384	0.402
	192	0.406	0.419	0.433	0.419	0.417	**0.416**	**0.405**	**0.416**	0.475	0.462	0.460	0.445	0.420	0.448	0.534	0.504	0.538	0.504	0.500	0.482	0.436	0.429
	336	**0.434**	0.441	0.475	0.441	0.457	**0.436**	0.439	0.443	0.518	0.488	0.501	0.466	0.459	0.465	0.588	0.535	0.574	0.521	0.521	0.496	0.491	0.469
	720	**0.449**	0.467	0.475	0.464	0.459	**0.460**	0.472	0.490	0.547	0.533	0.500	0.488	0.506	0.507	0.643	0.616	0.562	0.535	0.514	0.512	0.521	0.500
	Avg	**0.414**	0.430	0.441	0.429	0.425	**0.425**	0.422	0.437	0.491	0.479	0.469	0.454	0.440	0.460	0.570	0.537	0.542	0.510	0.496	0.487	0.458	0.450
ETTh2	96	**0.272**	0.337	0.288	0.337	0.277	**0.328**	0.289	0.353	0.397	0.437	0.302	0.348	0.358	0.397	0.476	0.458	0.340	0.391	0.346	0.388	0.340	0.374
	192	**0.345**	0.389	0.375	0.392	0.356	**0.379**	0.383	0.418	0.520	0.504	0.388	0.400	0.429	0.439	0.512	0.493	0.430	0.439	0.456	0.452	0.402	0.414
	336	**0.360**	**0.405**	0.423	0.434	0.400	0.417	0.448	0.465	0.626	0.559	0.426	0.433	0.496	0.487	0.552	0.551	0.485	0.479	0.482	0.486	0.452	0.452
	720	**0.398**	**0.436**	0.434	0.446	0.416	0.437	0.605	0.551	0.863	0.672	0.431	0.446	0.463	0.474	0.562	0.560	0.500	0.497	0.515	0.511	0.462	0.468
	Avg	**0.344**	0.392	0.380	0.402	0.362	**0.390**	0.431	0.446	0.602	0.543	0.387	0.407	0.437	0.449	0.526	0.516	0.439	0.452	0.450	0.459	0.414	0.427
ETTm1	96	**0.298**	**0.343**	0.343	0.367	0.304	**0.343**	0.299	**0.343**	0.374	0.400	0.329	0.367	0.379	0.419	0.386	0.398	0.375	0.398	0.505	0.475	0.338	0.375
	192	**0.334**	**0.363**	0.381	0.383	0.354	0.373	0.335	0.365	0.400	0.407	0.367	0.385	0.426	0.441	0.459	0.444	0.408	0.410	0.553	0.496	0.374	0.387
	336	**0.363**	**0.382**	0.414	0.404	0.379	0.393	0.369	0.386	0.438	0.438	0.399	0.410	0.445	0.459	0.495	0.464	0.435	0.428	0.621	0.537	0.410	0.411
	720	0.428	**0.419**	0.475	0.438	0.435	0.428	**0.425**	0.421	0.527	0.502	0.454	0.439	0.543	0.490	0.585	0.516	0.499	0.462	0.671	0.561	0.478	0.450
	Avg	**0.356**	**0.377**	0.403	0.398	0.368	0.384	0.357	0.378	0.435	0.437	0.387	0.400	0.448	0.452	0.481	0.456	0.429	0.425	0.588	0.517	0.400	0.406
ETTm2	96	**0.163**	0.253	0.181	0.264	**0.163**	**0.242**	0.167	0.269	0.209	0.308	0.175	0.259	0.203	0.287	0.192	0.274	0.189	0.280	0.255	0.339	0.187	0.267
	192	**0.222**	0.295	0.246	0.304	0.226	**0.285**	0.224	0.303	0.311	0.382	0.241	0.302	0.269	0.328	0.280	0.339	0.253	0.319	0.281	0.340	0.249	0.309
	336	**0.273**	0.327	0.307	0.343	0.286	**0.324**	0.281	0.342	0.442	0.466	0.305	0.343	0.325	0.366	0.334	0.361	0.314	0.357	0.339	0.372	0.321	0.351
	720	**0.357**	**0.377**	0.407	0.398	0.383	0.381	0.397	0.421	0.675	0.587	0.402	0.400	0.421	0.415	0.417	0.413	0.414	0.413	0.433	0.432	0.408	0.403
	Avg	**0.254**	0.313	0.286	0.327	0.265	**0.308**	0.267	0.333	0.409	0.436	0.281	0.326	0.305	0.349	0.306	0.347	0.293	0.342	0.327	0.371	0.291	0.333
Weather	96	**0.149**	0.192	0.182	0.227	0.150	**0.190**	0.176	0.237	0.182	0.242	0.177	0.218	0.217	0.296	0.173	0.223	0.197	0.281	0.266	0.336	0.172	0.220
	192	**0.194**	**0.236**	0.227	0.262	0.200	0.238	0.220	0.282	0.227	0.287	0.225	0.259	0.276	0.336	0.245	0.285	0.237	0.312	0.307	0.367	0.219	0.261
	336	**0.240**	**0.269**	0.279	0.298	0.259	0.282	0.265	0.319	0.282	0.334	0.278	0.297	0.339	0.380	0.321	0.338	0.298	0.353	0.359	0.395	0.280	0.306
	720	**0.318**	0.327	0.352	0.344	0.339	0.336	0.333	0.362	0.352	0.386	0.354	0.348	0.403	0.428	0.414	0.410	0.352	**0.288**	0.419	0.428	0.365	0.359
	Avg	**0.225**	**0.256**	0.260	0.283	0.237	0.262	0.248	0.300	0.261	0.312	0.259	0.281	0.309	0.360	0.288	0.314	0.271	0.334	0.338	0.382	0.259	0.287
Electricity	96	0.137	0.232	0.197	0.273	**0.136**	**0.231**	0.140	0.237	0.207	0.307	0.181	0.270	0.193	0.308	0.169	0.273	0.187	0.304	0.201	0.317	0.168	0.272
	192	**0.149**	**0.240**	0.197	0.276	0.153	0.246	0.153	0.249	0.213	0.316	0.188	0.274	0.201	0.315	0.182	0.286	0.199	0.315	0.222	0.334	0.184	0.289
	336	**0.165**	**0.259**	0.212	0.292	0.170	0.264	0.169	0.267	0.230	0.333	0.204	0.293	0.214	0.329	0.200	0.304	0.212	0.329	0.231	0.338	0.198	0.300
	720	**0.193**	**0.285**	0.253	0.325	0.208	0.297	0.203	0.301	0.265	0.360	0.246	0.324	0.246	0.355	0.222	0.321	0.233	0.345	0.254	0.361	0.220	0.320
	Avg	**0.161**	**0.254**	0.215	0.292	0.167	0.260	0.166	0.263	0.229	0.329	0.205	0.290	0.214	0.327	0.193	0.296	0.208	0.323	0.227	0.338	0.192	0.295
Traffic	96	**0.394**	**0.270**	0.645	0.383	0.432	0.279	0.410	0.282	0.615	0.391	0.462	0.295	0.587	0.366	0.612	0.338	0.607	0.392	0.613	0.388	0.593	0.321
	192	**0.401**	**0.281**	0.598	0.360	0.442	0.289	0.423	0.287	0.601	0.382	0.466	0.296	0.604	0.373	0.613	0.340	0.621	0.399	0.616	0.382	0.617	0.336
	336	**0.411**	**0.277**	0.605	0.363	0.456	0.295	0.436	0.296	0.613	0.386	0.482	0.304	0.621	0.383	0.618	0.328	0.622	0.396	0.622	0.337	0.629	0.336
	720	**0.445**	**0.294**	0.644	0.382	0.487	0.311	0.466	0.315	0.658	0.407	0.514	0.322	0.626	0.382	0.653	0.355	0.632	0.396	0.660	0.408	0.640	0.350
	Avg	**0.413**	**0.281**	0.623	0.372	0.454	0.293	0.433	0.295	0.622	0.392	0.481	0.304	0.610	0.376	0.624	0.340	0.621	0.396	0.628	0.379	0.620	0.336
1st Count		32	20	0	0	2	15	2	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0

achieves the best average MSE on all seven datasets, indicating that forecasting performance is not restricted to a particular dataset or prediction horizon.

For MAE, EviTime achieves the best or tied-best result in 20 of the 35 settings. In terms of the results averaged over the four prediction horizons, EviTime obtains the best MAE on ETTm1, Weather, Electricity, and Traffic, and the second-best MAE on ETTh2 and ETTm2. Although TimeEmb achieves a lower average MAE on ETTh1, EviTime obtains the lowest average MSE on this dataset. These results reflect a metric-dependent difference between the methods while showing that EviTime maintains competitive performance across most datasets.

As the prediction horizon increases, the forecasting errors of most methods generally increase because longer-term prediction involves greater temporal uncertainty and weaker dependence on recent observations. Nevertheless, at the longest prediction horizon of $PL = 720$, EviTime achieves the best MSE on six of the seven datasets and the best MAE on five datasets. This result suggests that the proposed method can preserve useful temporal information when the forecasting horizon becomes longer, rather than relying only on short-term local variations.

The performance differences are particularly apparent on the high-dimensional Electricity and Traffic datasets. These datasets contain 321 and 862 variables, respectively, and exhibit complex cross-variable dependencies and temporal variations. EviTime achieves the best average MSE and MAE on both datasets. This observation is consistent with the design of the proposed framework: TEFE characterizes the reliability of temporal and variable observations, GVFE represents pairwise tem-

poral similarity, directional variation, and amplitude information through three-channel images, and MLFE introduces dataset-, task-, and sample-level semantic context. The joint use of TEFE, GVFE, and MLFE provides complementary information for forecasting complex multivariate time series.

Overall, the complete horizon-specific results support the averaged results reported in the main paper. EviTime maintains stable performance across different datasets and prediction horizons, while the advantages are more evident in settings involving long prediction horizons, high-dimensional variables, and complex temporal dynamics.

G Case Studies

To provide an intuitive illustration of the forecasting behavior of EviTime, ETTm2 and Electricity are selected as representative case studies because they exhibit distinct temporal characteristics. ETTm2 represents a relatively low-dimensional and high-frequency forecasting scenario with periodic dependencies and nonlinear temporal variations, allowing the model's ability to capture fine-grained temporal patterns to be examined. In contrast, Electricity represents a high-dimensional forecasting scenario characterized by multiple seasonal patterns and pronounced fluctuations across variables. Together, these two datasets provide complementary examples for qualitatively examining the performance of EviTime under periodic and high-dimensional temporal dynamics. Figures 5 and 6 compare the predicted time series with the corresponding ground-truth observations on ETTm2 and Electricity, respectively.

As shown in Figure 5, EviTime follows the overall evolutionary trend of the ETTm2 series while reproducing a substantial proportion of the local nonlinear variations. The predicted sequence generally remains close to the ground-truth trajectory across both rising and falling intervals, indicating that the model can represent periodic dependencies and short-term temporal changes in this case.

Figure 6 presents the forecasting results on the Electricity dataset, which contains a larger number of variables and more pronounced temporal fluctuations. The predicted curves generally follow the trajectories of the observed sequences, including several rapid transitions and local peaks. Although deviations remain in some highly fluctuating regions, the overall visualization suggests that EviTime can track major temporal patterns in a high-dimensional forecasting setting.

The application of EviTime may provide broader societal benefits in sustainable energy management and power grid operation. Forecasting on datasets such as ETTm2 and Electricity can support power demand estimation, load scheduling, generation planning, and real time operational coordination. More reliable multivariate forecasts may help system operators identify upcoming demand changes, allocate generation resources more effectively, and reduce mismatches between electricity supply and consumption. Improved forecasting can also support the integration of renewable energy by providing earlier estimates of fluctuations in demand and system load. In addition, more accurate temporal predictions may assist grid operators in preparing for peak consumption periods, reducing unnecessary reserve allocation, and improving the utilization of existing energy infrastructure. Therefore, EviTime has the potential to contribute to more efficient, stable, and resilient energy management systems while supporting the transition toward low carbon power networks.

H Supplementary Few-Shot and Zero-Shot Forecasting Results

This section reports the complete horizon-specific results of the few-shot and zero-shot experiments summarized in the main paper. The few-shot experiments evaluate EviTime under limited supervision by retaining only 10% or 5% of the training data on Weather, Electricity, and Traffic. EviTime is compared with TEFN, TimeEmb, DLinear, PatchTST, and TimesNet under prediction lengths $PL \in \{96, 192, 336, 720\}$. The zero-shot experiments further evaluate cross-dataset transferability by training the models on a source ETT dataset and directly evaluating them on a target ETT dataset without fine-tuning. MSE and MAE are used as evaluation metrics, with lower values indicating better forecasting performance.

As shown in Table 7, the effect of limited training data varies across datasets when 10% of the original training samples are retained. EviTime achieves the lowest average MSE and MAE on Electricity. On Weather, EviTime obtains the second-lowest average MAE, while the average MSE remains close to that of DLinear and PatchTST. On Traffic, EviTime achieves the second-lowest

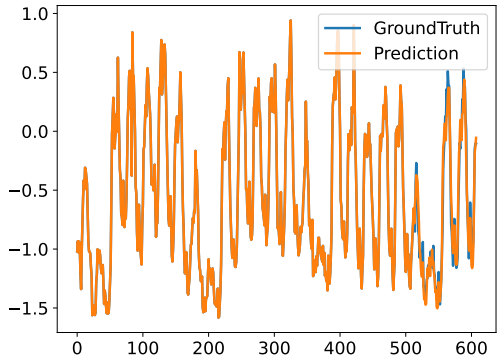

Figure 5: Qualitative results on ETTm2, comparing the predicted values with the ground-truth observations.

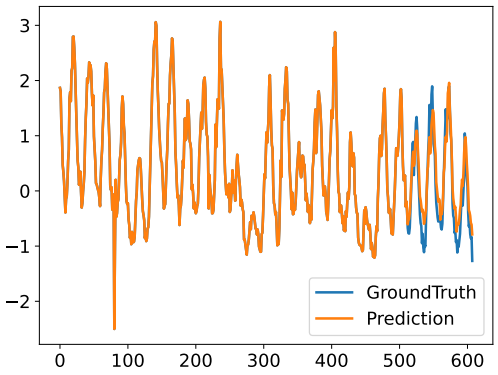

Figure 6: Qualitative results of Electricity, comparing the predicted values with the ground-truth observations.

Table 7: Detailed few-shot forecasting results of EviTime and five baseline models on Weather, Electricity, and Traffic using 10% of the original training data. Results are reported under prediction lengths $PL \in \{96, 192, 336, 720\}$ using MSE and MAE. The **Avg** row reports the average across the available prediction horizons. For Traffic, the result at $PL = 720$ is not reported; therefore, the average is calculated over $PL \in \{96, 192, 336\}$. The best and second-best results are marked in bold and underlined, respectively.

Methods		Ours		TEFN		TimeEmb		DLinear		PatchTST		TimesNet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	**0.165**	0.212	0.188	0.231	**0.165**	**0.210**	0.171	0.224	**0.165**	0.215	0.184	0.230
	192	**0.208**	0.249	0.233	0.268	0.210	**0.245**	0.215	0.263	0.210	0.257	0.245	0.283
	336	**0.255**	**0.285**	0.283	0.303	0.265	0.288	0.258	0.299	0.259	0.297	0.305	0.321
	720	0.344	0.354	0.362	0.350	0.345	**0.343**	**0.320**	0.346	0.332	0.346	0.381	0.371
	Avg	0.243	0.275	0.267	0.288	0.246	**0.272**	**0.241**	0.283	0.242	0.279	0.279	0.301
Electricity	96	0.146	0.246	0.204	0.279	0.148	**0.238**	0.150	0.253	**0.140**	**0.238**	0.299	0.373
	192	**0.159**	0.256	0.205	0.282	0.165	0.262	0.164	0.264	0.160	**0.255**	0.305	0.379
	336	**0.177**	**0.274**	0.219	0.299	0.182	0.278	0.181	0.282	0.180	0.276	0.319	0.391
	720	**0.208**	**0.299**	0.261	0.332	0.215	0.304	0.223	0.321	0.241	0.323	0.369	0.426
	Avg	**0.173**	**0.269**	0.222	0.298	0.178	0.270	0.180	0.280	0.180	0.273	0.323	0.392
Traffic	96	0.409	0.291	0.660	0.390	0.445	**0.286**	0.419	0.298	**0.403**	0.289	0.719	0.416
	192	0.430	0.302	0.610	0.368	0.455	0.297	0.434	0.305	**0.415**	**0.296**	0.748	0.428
	336	0.437	0.306	0.618	0.371	0.468	**0.303**	0.449	0.313	**0.426**	0.304	0.853	0.471
	720	–	–	–	–	–	–	–	–	–	–	–	–
	Avg	0.425	0.300	0.630	0.376	0.456	**0.295**	0.434	0.305	**0.415**	0.296	0.773	0.438

average MSE, whereas TimeEmb and PatchTST obtain lower average MAE values. These results indicate that EviTime retains useful forecasting capability when only a limited proportion of the training data is available, although the relative performance differs across datasets and evaluation metrics.

As reported in Table 8, further reducing the training ratio to 5% leads to more pronounced differences among the forecasting methods. EviTime achieves the lowest average MSE and the second-lowest average MAE on Weather, indicating that EviTime can still capture the main meteorological patterns under severe data scarcity. However, on Electricity and Traffic, DLinear and PatchTST achieve lower average errors. This observation suggests that high-dimensional datasets are more sensitive to substantial reductions in training samples, and relatively lightweight forecasting architectures may be easier to optimize in extremely limited-data settings.

A comparison between Tables 7 and 8 shows that the forecasting errors of most methods generally increase as the available training data decrease from 10% to 5%. This degradation is relatively lim-

Table 8: Detailed few-shot forecasting results of EviTime and five baseline models on Weather, Electricity, and Traffic using 5% of the original training data. Results are reported under prediction lengths $PL \in \{96, 192, 336, 720\}$ using MSE and MAE. The **Avg** row reports the average across the available prediction horizons. For Traffic, the result at $PL = 720$ is not reported; therefore, the average is calculated over $PL \in \{96, 192, 336\}$. The best and second-best results are marked in bold and underlined, respectively.

Methods		Ours		TEFN		TimeEmb		DLinear		PatchTST		TimesNet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	96	0.174	**0.222**	0.195	0.236	0.182	0.223	0.184	0.242	**0.171**	0.224	0.207	0.253
	192	0.216	**0.257**	0.241	0.275	**0.214**	0.258	0.228	0.283	0.230	0.277	0.272	0.307
	336	**0.262**	**0.292**	0.293	0.312	0.273	0.296	0.279	0.322	0.294	0.326	0.313	0.328
	720	**0.349**	0.382	0.368	0.358	0.353	**0.350**	0.364	0.388	0.384	0.387	0.400	0.385
	Avg	**0.250**	0.288	0.274	0.295	0.256	**0.281**	0.263	0.308	0.269	0.303	0.298	0.318
Electricity	96	0.160	0.265	0.211	0.287	0.159	0.245	0.150	0.251	**0.145**	**0.244**	0.315	0.389
	192	0.170	0.270	0.212	0.290	0.176	0.276	**0.163**	0.263	**0.163**	**0.260**	0.318	0.396
	336	0.192	0.291	0.226	0.306	0.194	0.298	**0.175**	**0.278**	0.183	0.281	0.340	0.415
	720	0.255	0.317	0.268	0.339	**0.222**	**0.311**	0.269	0.331	0.233	0.323	0.635	0.613
	Avg	0.194	0.286	0.229	0.306	0.188	0.283	**0.176**	**0.275**	0.181	0.277	0.402	0.453
Traffic	96	0.436	0.318	0.689	0.407	0.456	0.303	0.427	0.304	**0.404**	**0.286**	0.854	0.492
	192	0.482	0.338	0.642	0.384	0.496	0.343	0.447	0.315	**0.412**	**0.294**	0.894	0.517
	336	0.491	0.346	0.659	0.387	0.510	0.359	0.478	0.333	**0.439**	**0.310**	0.853	0.471
	720	–	–	–	–	–	–	–	–	–	–	–	–
	Avg	0.470	0.334	0.663	0.393	0.487	0.335	0.450	0.317	**0.418**	**0.296**	0.867	0.493

ited on Weather but is more evident on Electricity and Traffic. Since Electricity and Traffic contain substantially more variables, learning stable temporal and cross-variable dependencies from only 5% of the original training data becomes more challenging. Therefore, the few-shot results demonstrate that EviTime remains applicable under limited supervision, while the advantage becomes less consistent in extremely data-scarce and high-dimensional settings.

Table 9 reports the complete zero-shot forecasting results under four source-to-target transfer settings. EviTime achieves the lowest average MSE and MAE in all four settings. Across the 16 horizon-specific transfer settings, EviTime obtains the best or tied-best MSE in 15 settings and the best or tied-best MAE in 14 settings. The main exceptions occur under ETTh1 $\rightarrow$ ETTm2: PatchTST achieves lower MSE and MAE at $PL = 96$, while TimesNet obtains a slightly lower MAE at $PL = 336$.

As further shown in Table 9, EviTime maintains competitive transfer performance when the source and target datasets have different sampling granularities, such as ETTh1 $\rightarrow$ ETTm2 and ETTh2 $\rightarrow$ ETTm2. This result indicates that the learned representations are not entirely dependent on the temporal resolution of the source dataset. The evidence temporal features, two-dimensional structural representations, and metadata-guided semantic information provide complementary descriptions of the input series, which may facilitate forecasting knowledge transfer across related datasets.

Overall, Tables 7–9 provide a detailed analysis of the generalization performance summarized in the main paper. Tables 7 and 8 show that EviTime maintains competitive forecasting capability when the available training data are reduced to 10% and 5%, respectively. The performance on Weather remains relatively stable under severe data scarcity, whereas the results on Electricity and Traffic reveal greater sensitivity to limited supervision because of their higher dimensionality and more complex cross-variable dependencies. Table 9 further demonstrates that EviTime achieves consistent zero-shot transfer across different ETT datasets and sampling granularities without target-domain fine-tuning. The complementary integration of evidence-aware temporal representations, two-dimensional structural patterns, and metadata-guided semantics may contribute to such cross-dataset adaptability. Nevertheless, the results in Table 8 indicate that improving sample efficiency and reducing the dependence on training data remain important directions for high-dimensional forecasting scenarios.

Table 9: Detailed zero-shot forecasting results of EviTime and four baseline models under four source-to-target transfer settings. The notation "source → target" indicates that a model is trained on the source dataset and directly evaluated on the target dataset without fine-tuning. Results are reported under prediction lengths $PL \in \{96, 192, 336, 720\}$ using MSE and MAE, and the **Avg** row presents the average across the four prediction horizons. The best and second-best results are marked in bold and underlined, respectively.

Methods		Ours		DLinear		PatchTST		Autoformer		TimesNet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1 → ETTh2	96	**0.281**	**0.343**	0.347	0.400	0.304	0.350	0.469	0.486	0.358	0.387
	192	**0.342**	**0.380**	0.447	0.460	0.386	0.400	0.634	0.567	0.427	0.429
	336	**0.380**	**0.414**	0.515	0.505	0.414	0.428	0.655	0.588	0.449	0.451
	720	**0.392**	**0.423**	0.665	0.589	0.419	0.443	0.570	0.549	0.448	0.458
	Avg	**0.349**	**0.390**	0.493	0.488	0.380	0.405	0.582	0.548	0.421	0.431
ETTh1 → ETTm2	96	0.217	0.307	0.255	0.357	**0.215**	**0.304**	0.352	0.432	0.239	0.313
	192	**0.267**	**0.334**	0.338	0.413	0.275	0.339	0.413	0.460	0.291	0.342
	336	**0.328**	0.372	0.425	0.465	0.334	0.373	0.465	0.489	0.342	**0.371**
	720	**0.419**	**0.419**	0.640	0.573	0.431	0.424	0.599	0.551	0.434	**0.419**
	Avg	**0.308**	**0.358**	0.415	0.452	0.314	0.360	0.457	0.483	0.327	0.361
ETTh2 → ETTh1	96	**0.425**	**0.437**	0.689	0.555	0.485	0.465	0.693	0.569	0.848	0.601
	192	**0.537**	**0.501**	0.707	0.568	0.565	0.509	0.760	0.601	0.860	0.610
	336	**0.529**	**0.504**	0.710	0.577	0.581	0.515	0.781	0.619	0.867	0.626
	720	**0.490**	**0.493**	0.704	0.596	0.628	0.561	0.796	0.644	0.887	0.648
	Avg	**0.495**	**0.484**	0.703	0.574	0.565	0.513	0.757	0.608	0.865	0.621
ETTh2 → ETTm2	96	**0.208**	**0.301**	0.240	0.336	0.226	0.309	0.263	0.352	0.248	0.324
	192	**0.272**	**0.340**	0.295	0.369	0.289	0.345	0.326	0.389	0.296	0.352
	336	**0.322**	**0.369**	0.345	0.397	0.348	0.379	0.387	0.426	0.353	0.383
	720	**0.394**	**0.408**	0.432	0.442	0.439	0.427	0.487	0.478	0.471	0.446
	Avg	**0.299**	**0.355**	0.328	0.386	0.325	0.365	0.366	0.411	0.342	0.376

I Appendix Summary

In summary, the appendix provides comprehensive supplementary materials supporting the theoretical foundations, methodological implementation, computational analysis, experimental reproducibility, and extended empirical evaluation of EviTime. Appendix A introduces the theoretical foundations underlying the framework, including Dempster-Shafer Theory for evidential reasoning, Gramian Angular Fields for time-series-to-image transformation, and the BLIP-2 architecture for vision-language representation learning. Appendix B presents the complete training procedure of EviTime and analyzes the time and space complexities of TEFE, GVFE, MLFE, and the pretrained vision-language backbone, providing a detailed view of the computational characteristics of the overall framework. Appendices C–E describe the benchmark datasets, baseline methods, computational environment, vision-language backbone configuration, warmup strategy, and early-stopping protocol, thereby providing the experimental settings required for reproducibility. Appendix F reports the complete horizon-specific long-term forecasting results under multiple prediction lengths across ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, and Traffic, offering a detailed comparison between EviTime and representative forecasting baselines beyond the averaged results reported in the main text. Appendix G provides qualitative case studies on ETTm2 and Electricity, illustrating forecasting behavior under periodic, nonlinear, and high-dimensional temporal dynamics through comparisons between predicted sequences and ground-truth observations. Appendix H further reports detailed few-shot and zero-shot forecasting results, including performance under 10% and 5% training-data settings and cross-dataset transfer between ETT benchmarks without target-domain fine-tuning. The supplementary materials collectively provide additional evidence for the effectiveness of evidence-aware temporal modeling, multimodal representation learning, forecasting performance across different horizons, applicability under limited supervision, and cross-dataset transferability of EviTime.