Markovian Linguistic-Temporal Bridge: Unlocking the Potential of LLMs for Time Series Forecasting

Siming Sun, Kai Zhang, Xuejun Jiang, Wenchao Meng*, Qinmin Yang
Zhejiang University, Hangzhou, China
{smsun, kzhangnesc, jiangxuejun, wmengzju, qmyang}@zju.edu.cn

Abstract

Adapting pretrained Large Language Models (LLMs) for time series forecasting primarily relies on token-level linguistic-temporal alignment, leading to the stacking of logically disjointed tokens as input. While empirically effective, these methods overlook a fundamental capability of LLMs: modeling linguistic logic and structure, rather than merely processing token features. To address this limitation, we propose the **M**arkovian-**G**uided **S**tructure-**A**ware **A**lignment (**MGSAA**). Our core contribution is a framework that transcends pointwise feature matching to achieve global structural isomorphism between the linguistic and temporal domains. Specifically, MGSAA distills latent evolutionary patterns of language within LLMs into a Markovian state transition graph, which is transferred as a structural prior to the time series domain. Under this prior, time series patches are decoded into latent states and then aligned via state-constrained cross-attention. Ultimately, MGSAA generates a token sequence topologically isomorphic to the LLM's inherent mental structure, reactivating its reasoning capabilities for forecasting. Comprehensive evaluations across multiple benchmarks demonstrate that MGSAA achieves state-of-the-art performance, providing an innovative solution for cross-modal alignment in LLM for time series forecasting. Code is available at https://github.com/sunzju/MGSAA.

1 Introduction

Time series forecasting (TSF) is a fundamental but challenging task that plays a critical role in domains such as healthcare, finance, and energy. Although deep learning models have achieved considerable success in TSF (Liu et al., 2022; Nie et al., 2023; Wang et al., 2024), they often generalize poorly across domains (Zerveas et al., 2021). In contrast, large language models (LLMs) (Devlin et al., 2019; Raffel et al., 2020; Touvron et al., 2023) have recently demonstrated impressive cross-domain generalization capabilities (Bommasani et al., 2021). Pretrained on massive natural language corpora, LLMs encode rich linguistic and semantic knowledge, enabling them to excel in a variety of downstream tasks with minimal adaptation. Motivated by both this generalization ability and the shared sequential nature of language and time series data, recent research has surged in adapting LLMs for TSF (Zhou et al., 2023; Xue and Salim, 2024; Liu et al., 2025b). However, despite promising empirical results, consensus on two fundamental questions remains elusive:

1. **Model Degradation:** Are we truly leveraging the LLM's inherent language modeling and reasoning capabilities, or merely relegating it to a feature extractor?
2. **Input Misalignment:** Are we providing the model with a comprehensible sequence, or feeding it a disjointed "bag-of-words" that disrupts semantic continuity?

These questions highlight two critical limitations in current paradigms. First, from a model perspective, prevailing approaches (Zhou et al., 2023; Liu et al., 2025c,a), whether prompting, fine-tuning, or reprogramming, predominantly leverage the LLM's vast and well-initialized parameters for time series representation, while failing to activate the LLM as a generative reasoner. Second, from an input perspective, current methods (Jin et al., 2024; Sun et al., 2024; Pan et al., 2024; Qin et al., 2025) rely on token-level alignment (e.g., pointwise retrieval) that matches time-series patches to text prototypes based on local similarity, which neglects the temporal dynamics inherent in sequential data, leading to semantically disjointed tokens. For instance, a continuous signal trajec-

*Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 22165–22180

tory of "Surging → Collapsing" might be mapped to incongruent tokens like "Growth → Apple" due to accidental feature similarity. Such fragmented inputs act more as a collection of keywords rather than logically coherent narratives, failing to trigger the LLMs' reasoning capabilities.

Crucially, the two issues are coupled: resolving the input misalignment, by restoring a coherent, language-like input is a prerequisite for preventing LLMs' degradation. Consequently, our research pivots to a core challenge: *How to transcend simple token-level retrieval and achieve cross-modal alignment that respects global dynamical evolution?*

It is well-established that the essence of global evolution, whether in physical systems or linguistic narratives, is governed by latent state transitions (Rabiner, 1989; Jelinek, 1998), specifically, the dependency of the current state on historical trajectories. This dependency can be effectively approximated as a Markovian property, allowing us to abstract sequential evolution into a probabilistic graph structure. Driven by this insight, we propose the **M**arkovian-**G**uided **S**tructure-**A**ware **A**lignment (**MGSAA**) framework, which elevates alignment from pointwise feature matching to global structural isomorphism. MGSAA comprises two synergistic mechanisms: **Language Structure Distillation in LLM:** To discern the input structures comprehensible to the LLM, we first seek to uncover the state evolutionary regularities embedded in its latent space on the source (linguistic) domain. Specifically, by clustering token representations from intermediate hidden layers, we distill its latent evolution patterns into a Markovian transition matrix, which serves as a global structural prior that characterizes evolution paths compatible with the frozen LLM. **Structure-Aware Semantic Alignment:** Building on this prior, we implement a structure-constrained alignment strategy. We transfer the linguistic structural prior to the time series domain and perform state decoding for patch sequences under the joint guidance of global structural constraints and local feature patterns. Each patch retrieves text embeddings within its assigned latent states rather than from an open-ended pool, producing a token sequence topologically isomorphic to the LLM's inherent mental structure and thereby reactivating its generative reasoning for forecasting without fine-tuning.

Our main contributions are summarized as follows:

- We identify model degradation and input misalignment as key limitations of current paradigms, demonstrating that point-wise retrieval disrupts semantic continuity, which hinders the full reasoning potential of LLMs in time series forecasting.
- We propose the MGSAA framework, which introduces a Markovian structural prior to elevate the alignment paradigm from local matching to global structure-aware alignment, enabling LLMs to process time series inputs that align with their intrinsic semantic space dynamics for accurate forecasting.
- Empirical evaluations across long-term, short-term, few-shot, and zero-shot forecasting across diverse benchmarks demonstrate state-of-the-art performance, exceptional generalization, and superior efficiency.

2 Related Works

2.1 LLMs for Time Series

LLMs have shown great potential for time series modeling due to their pattern recognition and generalization capabilities (Jiang et al., 2024). GPT4TS (Zhou et al., 2023) first explored LLMs in this domain, followed by works such as LLM4TS (Chang et al., 2023), which proposes a two-stage fine-tuning strategy to adapt LLMs to time series. However, these methods simply project time series into input-compatible formats for LLMs and rely heavily on fine-tuning. Time-LLM (Jin et al., 2024) reprograms LLMs by aligning time series patches with textual prototypes, while TEST (Sun et al., 2024) adopts a contrastive learning strategy to align the time series embedding space with that of LLMs. CALF (Liu et al., 2025c) further introduces dual branches for textual and temporal inputs to minimize the modality distribution gap in input, feature, and output spaces. FSCA (Hu et al., 2025) treats time series as linguistic components, performing context alignment within prompt statements. Although these methods attempt to bridge the modality gap, most perform token-level feature alignment, ignoring the global sequential dynamics.

2.2 Cross-Modal Alignment in LLMs

Cross-modal alignment aims to bridge modalities by projecting them into a shared representation space (Shen et al., 2023). Current research pre-

dominantly focuses on implicit alignment, where the relationship between modalities is learned automatically through training objectives, such as contrastive learning. CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), and CoCa (Yu et al., 2022) learn imagetext correspondences through large-scale contrastive objectives. Explicit alignment, in contrast, relies on expert-defined mappings, offering stronger interpretability. BLIP (Li et al., 2022) attempts to integrate supervised matching and contrastive training. However, unlike image-text pairs, where semantic correspondences can be visually verified, time-series data lacks such inherent interpretability. Consequently, most existing approaches resort to purely implicit alignment strategies, potentially limiting their explainability (Sun et al., 2024; Pan et al., 2024; Liu et al., 2025c; Hu et al., 2025; Liu et al., 2025a). In contrast, our method combines explicit Markovian priors with semantic cross-attention, forming a unified alignment framework that bridges explicit and implicit alignment paradigms.

3 Methodology

Our model architecture is given in Figure 1, which comprises four interlinked components, i.e., Language Structure Distillation in LLM, Structure-Constrained State Decoding, State-Conditioned Semantic Alignment, and LLM for Forecasting. The workflow commences with Language Structure Distillation, which operates as an independent preprocessing module. It distills a linguistic Markovian state transition graph within the LLM's latent space, establishing a global structural constraint for subsequent alignment. In the temporal domain, consider a multivariate time series input $\mathbf{X} \in \mathbb{R}^{C \times T}$ with C variates and T time steps. Following the prevalent channel-independence paradigm (Nie et al., 2023), we treat each channel independently and apply instance normalization to mitigate distribution shifts (Kim et al., 2022). Specifically, we segment the sequence using a patch length l_p and stride s_p, yielding $P = \lfloor \frac{T-l_p}{s_p} \rfloor + 1$ patches. Accordingly, the time series is represented as C patch sequences $\mathbf{X}_P^{(i)} = (\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \cdots, \mathbf{x}_P^{(i)})$, where $i = 1, \cdots, C$. Subsequently, these patch sequences undergo Structure-Constrained State Decoding. Under the joint guidance of temporal features and global linguistic structure prior, the module infers the optimal sequence of latent states for the patches. Thereafter, State-Conditioned Semantic Alignment enforces the patch embeddings to compute cross-attention with text prototypes strictly within their assigned latent state spaces, generating a structure-aware and semantically aligned token sequence. Finally, this sequence, combined with the original temporal features via a residual connection, is fed into the Frozen LLM for inference, which is then projected to yield the final forecast $\hat{\mathbf{Y}} \in \mathbb{R}^{C \times T'}$.

3.1 Language Structure Distillation in LLM

To make the evolutionary logic of language in LLMs explicit and tractable, we model the hidden representations sequence induced by autoregressive token processing as a Hidden Markov Process over a dynamical corpus. Specifically, inspired by recent works on latent variable distillation for large deep generative models (Liu et al., 2023; Zhang et al., 2023), we cluster context-aware token embeddings from LLM's intermediate layers and distill the latent transition behavior into a Hidden Markov Model (HMM) (Rabiner, 1989), where the state transition matrix explicitly encodes admissible evolution patterns in the LLM's latent space.

Dynamical Subspace Construction Directly distilling transition structures from the full pretraining corpus of LLMs is impractical. On the one hand, general-purpose corpora contain highly heterogeneous logical patterns, which induces an extremely sparse and complex transition graph that is difficult to estimate reliably. On the other hand, natural language permits abrupt semantic shifts, topic jumps, and abstract concepts, whereas physical time-series processes typically exhibit smoother evolution. This mismatch makes a general linguistic transition structure poorly suited as a prior for time-series modeling.

To address these issues, we propose constructing a *Dynamical Subspace* $\mathcal{D}$. Within this restricted space, our objective is to derive a linguistic latent transition structure that encodes generic smoothness and continuity properties transferable to the time-series domain, rather than simulating specific temporal dynamics such as periodicity or monotonicity. Specifically, smoothness and continuity refer to the predictable evolution of states characterized by clear causal or sequential relationships. For instance, a sentence like *"He slammed nine shiny coins onto the worn ma-*

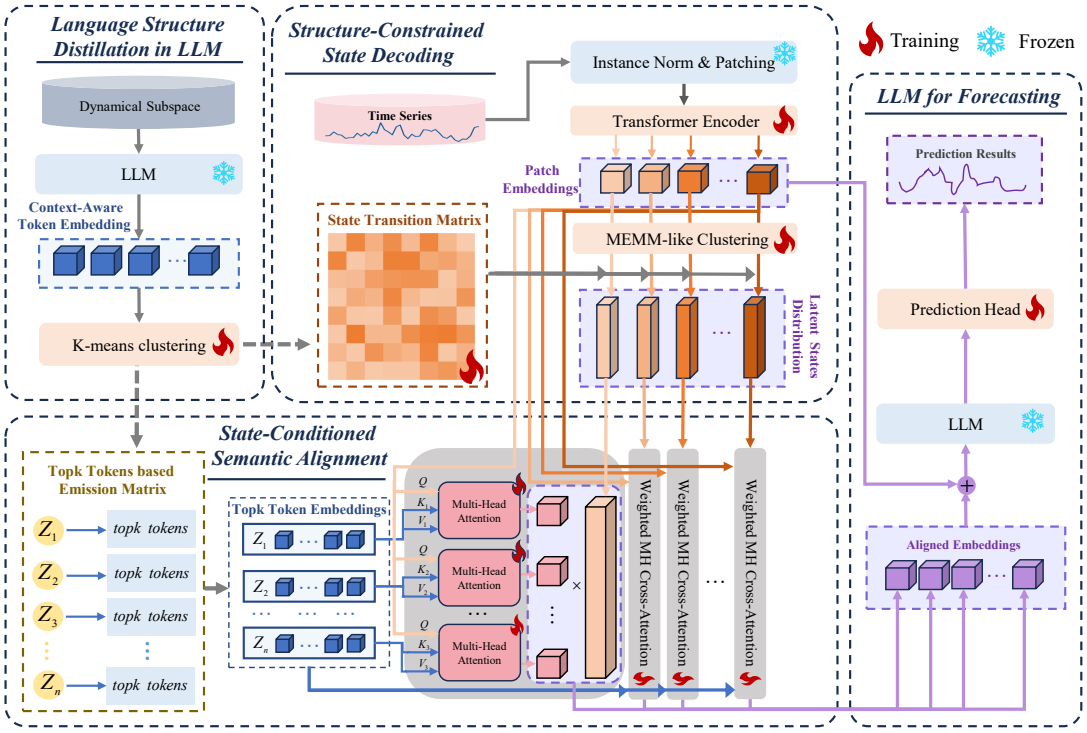


Figure 1: An overview of MGSAA. The pipeline begins with *Language Structure Distillation in LLM*, which is executed independently and outputs structure prior for downstream modules. Based on the prior, *Structure-Constrained State Decoding* is performed first to assign latent states to patches, followed by *State-Conditioned Semantic Alignment* to execute alignment via the latent states, finally, the *LLM for Forecasting* module produces prediction.

hogany counter" is grammatically flawless but highly "unsmooth" in terms of state evolution, it contains unpredictable abrupt actions and leaping subjective descriptions, strictly contradicting the continuous dynamics of physical time series.

Therefore, we deliberately utilize abstract physical and mathematical narratives, as they naturally exhibit a high degree of causal rigor and logical continuity. We design four families of prompt templates $\mathcal{P}$ with diverse narrative forms via a controlled Cartesian product of specific semantic components (such as subjects, verbs, and contexts). Based on each $P \in \mathcal{P}$, we sample a subspace corpus $\mathcal{D}$ consisting of M text sequences using the LLM's causal generation process $G(\cdot)$:

$$\mathcal{D} = \{S^{(i)} \mid S^{(i)} = G(P_i), P_i \in \mathcal{P}\}_{i=1}^{M} \quad (1)$$

This approach strictly confines the model to a smooth and coherent descriptive space. During autoregressive generation, the model naturally follows the path of least resistance to extend the smooth syntactic structure of P to generate a logically coherent sentence $S^{(i)} = \{w_1, w_2, \ldots, w_L\}$, effectively filtering out abrupt semantic leaps. More details on the dynamical subspace construction are provided in Appendix 5

Context-Aware Representation and Latent State Clustering For each token w_t in a sequence $S^{(i)}$, we extract a context-aware hidden representation $\mathbf{e}_t$ from the l-th layer of the LLM under causal masking. This ensures that each representation encodes only historical information available up to time step t, allowing the current state to implicitly summarize relevant past context in a Markovian manner. To mitigate the well-known anisotropy of LLM representations (Ethayarajh, 2019; Gao et al., 2019), we apply feature standardization followed by ℓ_2 normalization:

$$\tilde{\mathbf{e}}_t = \text{L2-Norm}\left(\frac{\mathbf{e}_t - \mu}{\sigma + \epsilon}\right) \quad (2)$$

where μ and σ denote the mean and standard deviation computed over $\mathcal{D}$.

We then assume that the latent evolution of the LLM within the dynamical subspace can be discretized into a finite set of states. Accordingly, we perform K-means clustering on the normalized representations $\tilde{\mathbf{e}}_t$ to identify N latent states $\mathcal{Z} = \{z_1, \ldots, z_N\}$, using N cluster centers $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_N\}$, by minimizing intra-cluster

variance:

$$\min_{\mathbf{C}} \sum_{t} \min_{n} |\tilde{\mathbf{e}}_t - \mathbf{c}_n|^2 \tag{3}$$

Each token w_t is then assigned a latent state $z_t \in \{z_1, \ldots, z_N\}$ according to $z_t = \arg\min_{c_n} \|\tilde{\mathbf{e}}_t - \mathbf{c}_n\|$, yielding a discrete state sequence for Markovian structure estimation.

HMM Parameter Estimation Given the discrete latent state sequences $\{z_t\}$ and the corresponding token sequences $\{w_t\}$, we estimate the HMM parameters $\Phi = \{\mathbf{A}^{\text{text}}, \mathbf{B}^{\text{text}}, \boldsymbol{\pi}^{\text{text}}\}$ via frequency-based counting with Laplace smoothing parameter δ.

Transition Matrix $\mathbf{A}^{\text{text}} \in \mathbb{R}^{N\times N}$: Encodes transition patterns between latent states.

$$A_{ij} = P(z_{t+1} = j \mid z_t = i) = \frac{\text{Count}(z_t = i, z_{t+1} = j) + \delta}{\sum_{n=1}^{N}(\text{Count}(z_t = i, z_{t+1} = n) + \delta)} \tag{4}$$

Emission Matrix $\mathbf{B}^{\text{text}} \in \mathbb{R}^{N\times|\mathcal{V}|}$: Maps latent states to semantic distributions over vocabulary $\mathcal{V}$.

$$B_{nv} = P(w_t = v \mid z_t = n) = \frac{\text{Count}(z_t = n, w_t = v) + \delta}{\sum_{v'\in\mathcal{V}}(\text{Count}(z_t = n, w_t = v') + \delta)} \tag{5}$$

Initial Distribution $\boldsymbol{\pi}^{\text{text}} \in \mathbb{R}^{N}$: Encodes starting state activation.

$$\pi_n = P(z_1 = n) = \frac{\text{Count}(z_1 = n) + \delta}{\sum_{j=1}^{N}(\text{Count}(z_1 = j) + \delta)} \tag{6}$$

The estimated HMM provides a compact representation of transition regularities in the LLM's latent space and will be used as a structural prior in subsequent time series modeling stages.

3.2 Structure-Constrained State Decoding

After obtaining the linguistic structural prior, Sections 3.2 and 3.3 form a coherent process, with the goal of achieving structure-constrained semantic alignment on the time series. This requires transforming the continuous patch sequence into a discrete structured variable that encodes global evolution constraints, enabling the subsequent alignment in Section 3.3 to be structurally guided. To this end, in this section, we introduce a structure-constrained state decoding procedure that generates a sequence of latent state distributions for time-series patches, considering both local patch features and global state transition structure.

Feature-driven Soft Clustering This step provides a feature-based estimation of the state probability for each patch. We first encode the patch sequence $\mathbf{X}_P$ using a shallow Transformer encoder:

$$\mathbf{H}^{\text{time}} = (\mathbf{h}_1, \ldots, \mathbf{h}_P) = \textit{Enc}(\textit{Embed}(\mathbf{X}_P)) \in \mathbb{R}^{P\times d} \tag{7}$$

where d is the model dimension.

Then, we introduce N learnable cluster prototypes $\{\boldsymbol{\mu}_n \in \mathbb{R}^d\}_{n=1}^{N}$, where N is chosen to match the state space size of the Markovian prior. Each patch embedding $\mathbf{h}_P$ is softly assigned to these prototypes based on feature similarity, resulting in a local observation evidence:

$$\boldsymbol{\gamma}_p = \left[\gamma_p^n\right]_{n=1}^{N} \tag{8}$$

where

$$\gamma_p^k = \frac{\exp(-\|\mathbf{h}_p - \boldsymbol{\mu}_n\|^2)}{\sum_{j=1}^{N}\exp(-\|\mathbf{h}_p - \boldsymbol{\mu}_j\|^2)} \tag{9}$$

Markovian-regularized Sequential Decoding To incorporate global structural constraints, we transfer initial distribution $\boldsymbol{\pi}^{\text{text}}$ and transition matrix $\mathbf{A}^{\text{text}}$ from the language domain to initialize the learnable parameters $\boldsymbol{\pi}^{\text{time}}$ and $\mathbf{A}^{\text{time}}$ in the temporal domain. This strategy leverages the linguistic prior while allowing for fine-tuning to time-series dynamics. We then recursively compute the posterior latent state probability $\tilde{\boldsymbol{\gamma}}_p$ of each patch p :

$$\begin{cases} \tilde{\boldsymbol{\gamma}}_1 = \text{Softmax}(\boldsymbol{\pi}^{\text{time}} \odot \boldsymbol{\gamma}_1), \quad p = 1 \\ \tilde{\boldsymbol{\gamma}}_p = \text{Softmax}\big((\tilde{\boldsymbol{\gamma}}_{p-1}^{\top}\mathbf{A}^{\text{time}})^{\top} \odot \boldsymbol{\gamma}_p\big), \\ \qquad\qquad\qquad\qquad p = 2, \ldots, P \end{cases} \tag{10}$$

where $\odot$ denotes the element-wise multiplication. This decoding process yields a sequence of structurally regularized latent state distributions:

$$\tilde{\boldsymbol{\Gamma}}^{\text{time}} = (\tilde{\boldsymbol{\gamma}}_1, \ldots, \tilde{\boldsymbol{\gamma}}_P), \quad \tilde{\boldsymbol{\gamma}}_p \in \Delta^N \tag{11}$$

where Δ^N denotes the N-dimensional probability simplex. The resulting distributions provide a structured scaffold for the subsequent state-constrained semantic alignment. The entire workflow of Markovian structure constrained latent state decoding is depicted in Figure 2.

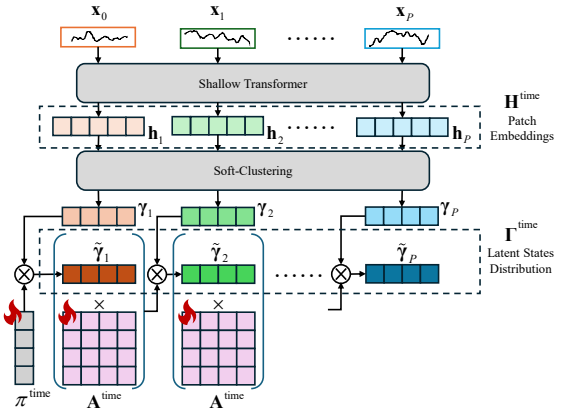


Figure 2: The workflow of Markovian structure constrained latent state decoding

3.3 State-Conditioned Semantic Alignment

In this stage, we operationalize the structurally regularized posterior state sequence $\tilde{\mathbf{\Gamma}}^{\text{time}} = (\tilde{\boldsymbol{\gamma}}_1, \ldots, \tilde{\boldsymbol{\gamma}}_P)$ into the cross-modal alignment process, by restricting semantic retrieval to be state-conditioned and then aggregate it according to $\tilde{\boldsymbol{\gamma}}_p$.

Specifically, for each latent state z_n, we select its top-k tokens from the language-domain emission matrix $\mathbf{B}^{\text{text}}$ and retrieve their corresponding embedding representations in the LLM embedding space to form a state-specific prototype set $\mathbf{E}_n \in \mathbb{R}^{k \times d_{\text{llm}}}$. Given the patch representation $\mathbf{h}_p$, we compute a candidate aligned embedding under state z_n via multi-head cross-attention:

$$\mathbf{h}_p^n = \mathit{CrossAttention}(\mathbf{h}_p, \mathbf{E}_n, \tau = \sqrt{d_h}) \quad (12)$$

where τ is the attention temperature and $d_h = d_{\text{llm}}/H$ with H denoting the number of heads. Finally, we combine the state-conditioned candidates by performing a weighted aggregation, using the posterior probabilities $\tilde{\boldsymbol{\gamma}}_p$:

$$\tilde{\mathbf{h}}_p = \sum_{n=1}^{N} \tilde{\gamma}_p^n \cdot \mathbf{h}_p^n \quad (13)$$

This produces an aligned input embedding sequence $\tilde{\mathbf{H}}^{\text{time}} = (\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2, \ldots, \tilde{\mathbf{h}}_P) \in \mathbb{R}^{P \times d_{\text{llm}}}$ that is not only semantically meaningful to the LLM, but also structurally consistent with its learned transition regularities, enabling direct integration with a frozen LLM for downstream forecasting.

3.4 LLM for Forecasting

The aligned semantic vector $\tilde{\mathbf{h}}_p$ captures the coarse structural semantics implied by the transferred Markov prior. However, mapping continuous signals into a discrete state space may attenuate instance-specific details. To preserve such information, we introduce a residual projection of the patch representation:

$$\mathbf{h}_p^{\text{in}} = \tilde{\mathbf{h}}_p + \mathbf{W}_r \mathbf{h}_p \quad (14)$$

where $\mathbf{W}_r$ is a lightweight linear layer.

The resulting sequence $\mathbf{H}_{\text{in}} = (\mathbf{h}_1^{\text{in}}, \ldots, \mathbf{h}_P^{\text{in}})$ lies in the native embedding space of the LLM and is structurally regularized by the decoded state posteriors. We feed $\mathbf{H}_{\text{in}}$ into a frozen LLM to obtain output representations, and map them to the forecasting horizon via a prediction head:

$$\hat{\mathbf{Y}} = \text{Head}(\text{LLM}(\mathbf{H}_{\text{in}})) \quad (15)$$

4 Experiments and Analysis

We evaluate MGSAA on a wide range of forecasting tasks, including long-term, short-term, few-shot and zero-shot settings. Our approach consistently achieves state-of-the-art performance. Ablation studies further validate the effectiveness of Markovian structure constrained alignment, which is sufficient to activate frozen LLMs for time series forecasting.

Baselines. We compare our model against several competitive baselines from recent literature, covering a wide range of modeling paradigms: **Transformer-based models**: PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), FEDformer (Zhou et al., 2022), Non-stationary Transformer (Liu et al., 2022), and Autoformer (Wu et al., 2021); **CNN-based models**: TimesNet (Wu et al., 2023); **MLP-based models**: DLinear (Zeng et al., 2023); and **LLM-based models**: GPT4TS (Zhou et al., 2023), TimeLLM (Jin et al., 2024) and FSCA (Hu et al., 2025). For short-term forecasting scenarios, we additionally compare with N-HiTS (Challu et al., 2023) and N-BEATS (Oreshkin et al., 2020).

4.1 Key Experimental Details

To ensure fair comparison, we adopt a unified experimental setup for all LLM-based models, using a 6-layer GPT-2 (Radford et al., 2019) as the backbone, including our implementation. For FSCA, we follow its original best-reported 4-layer GPT-2 configuration. For all other non-LLM baselines, we follow the experimental settings and protocols provided in (Wu et al., 2023). Our experimental setup is detailed in the appendix 5. All experiments are conducted three times, and the reported results are averaged.

Table 1: Long-term forecasting results. The input sequence length T is set to 96 for all baselines. All the results are averaged from 4 different prediction lengths $H \in \{96, 192, 336, 720\}$. **Bold**: best, underlined: second best.

Models	MGSAA		FSCA		TimeLLM		GPT4TS		PatchTST		iTransformer		FEDformer		Stationary		Autoformer		TimesNet		DLinear	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.378**	**0.383**	0.400	0.402	0.392	0.403	0.390	0.398	0.388	0.402	0.407	0.411	0.446	0.455	0.529	0.478	0.614	0.526	0.410	0.418	0.404	0.408
ETTm2	**0.280**	**0.320**	0.283	0.326	0.291	0.334	0.284	0.329	0.290	0.333	0.291	0.334	0.303	0.349	0.517	0.438	0.332	0.368	0.295	0.332	0.355	0.401
ETTh1	**0.433**	**0.427**	0.438	0.441	0.454	0.447	0.448	0.437	0.448	0.447	0.464	0.455	0.439	0.457	0.629	0.560	0.501	0.490	0.460	0.455	0.460	0.456
ETTh2	**0.370**	**0.392**	0.380	0.409	0.387	0.408	0.382	0.408	0.384	0.413	0.383	0.407	0.442	0.453	0.544	0.498	0.458	0.462	0.407	0.421	0.564	0.519
Weather	**0.253**	**0.271**	0.260	0.280	0.273	0.290	0.264	0.284	0.258	0.280	0.260	0.281	0.313	0.364	0.288	0.309	0.376	0.406	0.259	0.285	0.266	0.318
Electricity	0.189	0.271	0.195	0.279	0.197	0.285	0.206	0.291	0.204	0.294	**0.175**	**0.266**	0.221	0.332	0.194	0.295	0.238	0.346	0.197	0.297	0.215	0.303
Traffic	0.439	**0.268**	0.459	0.294	0.509	0.324	0.489	0.317	0.482	0.308	**0.422**	0.282	0.610	0.379	0.643	0.355	0.644	0.399	0.625	0.331	0.624	0.383

Table 2: Short-term forecasting results on M4 dataset. The prediction length is set to $[6, 48]$, and the input lengths are configured to be double the forecasting horizons. **Bold**: best, underlined: second best.

Models		MGSAA	FSCA	TimeLLM	GPT4TS	PatchTST	iTransformer	FEDformer	Stationary	Autoformer	TimesNet	DLinear	N-HiTS	N-BEATS
Average	SMAPE	**11.707**	11.828	12.494	12.365	12.059	12.142	13.160	12.780	12.909	12.880	13.639	12.035	12.250
	MASE	**1.567**	1.580	1.731	1.767	1.623	1.631	1.775	1.756	1.771	1.836	2.095	1.625	1.698
	OWA	**0.841**	0.850	0.913	0.918	0.869	0.874	0.949	0.930	0.939	0.955	1.051	0.869	0.896

4.2 Long-term Forecasting

Setups. For long-term forecasting, we assess our MGSAA framework on seven widely used real-world datasets (ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, Traffic). To ensure a fair comparison, we adopt a unified input sequence length of $T = 96$ and evaluate performance under four forecasting horizons: $H \in \{96, 192, 336, 720\}$, using mean squared error (MSE) and mean absolute error (MAE).

Results. Comprehensive results are presented in Table 1. Our method outperforms all baselines in most scenarios. Specifically, compared to the sub-optimal model FSCA, MGSAA achieves average reductions of 2.9% in MSE and 4.1% in MAE. Against other LLM-based baselines: TimeLLM and GPT4TS, our method reduces MSE/MAE by 5.0%/5.8% and 6.6%/6.9%, respectively. Compared to other traditional baselines, the improvements are even more pronounced, with reductions often exceeding 10%.

4.3 Short-term Forecasting

Setups. We utilize the well-recognized M4 datasets (Makridakis et al., 2018), which comprise univariate marketing data collected yearly, quarterly, and monthly. In this case, the forecasting horizons are relatively short, spanning $[6, 48]$, and the input lengths are configured to be double the forecasting horizons. The evaluation metrics are symmetric mean absolute percentage error (SMAPE), mean absolute scaled error (MASE), and overall weighted average (OWA).

Results. As shown in Table 2, our method outperforms all baselines across various metrics in short-term forecasting, achieving a 1.0% performance improvement over FSCA, the previously best model for short-term forecasting, 7.9% over TimeLLM, and 8.2% over GPT4TS.

4.4 Few / Zero-shot Forecasting

LLMs have exhibited exceptional performance in both few-shot and zero-shot scenarios (Brown et al., 2020; Kojima et al., 2022). Although current LLM-based models outperform traditional deep learning approaches in these tasks, they have not fully harnessed the potential of LLMs to maximize their predictive capabilities in the few-shot setting. To demonstrate the insightfulness and efficacy of our method, we perform experiments in few-shot and zero-shot learning scenarios. In few-shot learning, a limited portion, only 10% of the training data is used on four ETT datasets. As shown in Table 3, our method consistently outperforms strong baselines, achieving an average error reduction of 5.7% over both FSCA and TimeLLM. In zero-shot learning, a model trained on one dataset is directly applied to test on a different dataset without further training. Zero-shot results are provided in the appendix 11.

4.5 Model Analysis

Effectiveness of Markovian Structure Constraint To assess whether the Markovian transition structure distilled from the language domain is meaningfully reflected in the temporal domain, we perform a quantitative and visual analysis on the ETTm1 dataset. We first visualize the transition matrices from both domains as heatmaps, as shown in Figure 3, which exhibit strikingly similar patterns. Moreover, the L_1 distance between

Table 3: Few-shot forecasting results on 10% training data of ETT datasets. All the results are averaged from 4 different prediction lengths $H \in \{96, 192, 336, 720\}$.

Models	MGSAA		FSCA		TimeLLM		GPT4TS		PatchTST		iTransformer		FEDformer		Stationary		TimesNet		DLinear	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.505**	**0.445**	0.602	0.503	0.565	0.491	0.607	0.498	0.619	0.496	0.580	0.497	0.696	0.570	1.070	0.680	0.674	0.535	0.568	0.500
ETTm2	**0.297**	**0.331**	0.302	0.339	0.303	0.341	0.303	0.337	0.302	0.342	0.306	0.344	0.358	0.394	0.428	0.414	0.322	0.355	0.329	0.383
ETTh1	**0.598**	**0.520**	0.695	0.553	0.616	0.530	0.688	0.554	0.623	0.532	0.748	0.587	0.749	0.608	0.845	0.631	0.864	0.625	0.647	0.552
ETTh2	0.423	0.422	**0.403**	**0.416**	0.497	0.465	0.574	0.495	0.490	0.459	0.443	0.443	0.553	0.525	0.772	0.588	0.487	0.468	0.446	0.461

the two matrices is as low as 0.487, indicating a strong preservation of the original language transition structure.

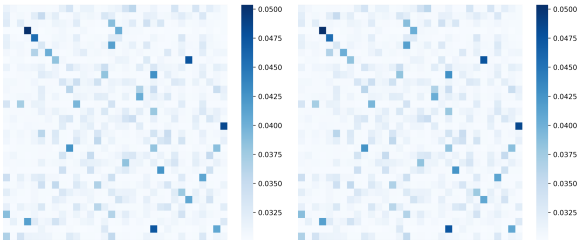


Figure 3: Markovian state transition heatmap. Left: language domain; Right: time series domain.

While Figure 3 demonstrates the successful transfer of the transition matrix at a macroscopic level, Figure 4 provides visual evidence at the sample level showing how this matrix fundamentally alters the decoding process. Specifically, without a structural prior, mapping continuous time series patches to text prototypes based merely on local feature similarity typically results in erratic and chaotic state jumping. In contrast, Figure 4 shows that as the series progresses, the assigned latent states do not transition randomly; instead, they evolve through smooth and structured patterns. This observation visually confirms that the global transition matrix effectively acts as a structural prior to constrain the latent state evolution.

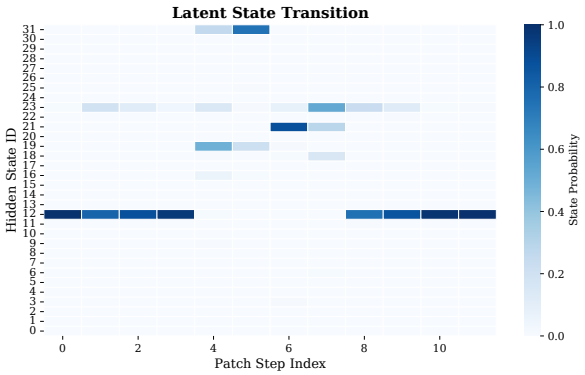


Figure 4: Latent state distributions over a single time series sample. The x-axis denotes the sequence of temporal patches, and the y-axis corresponds to latent state categories.

Effectiveness of Structure-Aware Semantic Alignment To evaluate the effectiveness of structure-aware semantic alignment, we designed three ablation variants: **w/o Alignment:** This variant removes the cross-modal alignment module, feeding raw time series embeddings directly into the LLM. This baseline evaluates the overall gain provided by our alignment strategy. **w/o Structural Prior:** This variant retains the alignment mechanism but replaces the Markovian prior with a randomly initialized transition matrix. It serves to validate the effectiveness of structural constraint. **w/o Residual:** This variant removes the instance-level residual connection, feeding only the retrieved structured text token sequence into the LLM. It isolates the contribution of structural alignment from numerical details. The results summarized in Table 4 show: Most significant drop in A: The removal of the alignment module leads to the sharpest performance decline, indicating that raw features alone fail to effectively reactivate the LLM's forecasting potential. Notable degradation in B: Even with the alignment mechanism intact, the absence of a proper Markovian structural prior results in a significant loss of accuracy. This confirms the necessity of utilizing global structural constraint. Marginal decline in C: Variant C exhibits a slight performance decrease. This suggests that while residual connections aid in refining numerical precision, the structured semantic sequence acts as the primary driver of forecasting accuracy.

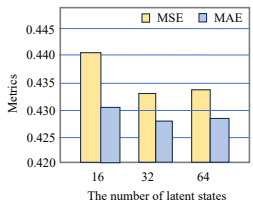
Table 4: Ablation study on the effectiveness of structure-aware semantic alignment on ETTm1 and Traffic datasets.

Ablation Setting	ETTh1		Traffic	
	MSE	MAE	MSE	MAE
MGSAA(Full Model)	**0.433**	**0.427**	**0.439**	**0.268**
A. w/o Alignment	0.464	0.447	0.461	0.285
B. w/o Structural Prior	0.448	0.439	0.454	0.278
C. w/o Residual	0.445	0.433	0.448	0.273

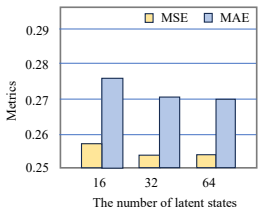
Hyperparameter Sensitivity We conduct a sensitivity analysis on two core hyperparameters of the MGSAA framework: the number of latent states $N \in \{16, 32, 64\}$ and the number of top-k tokens $k \in \{5, 50, 1000\}$. The experiments are performed on the ETTh1 and Weather datasets to evaluate their impact on forecasting accuracy.

The number of latent states N defines the granularity of the distilled Markovian structural prior. As illustrated in Figure 5a and 5b, $N = 16$ is insufficient for capturing the complex dynamics inherent in the language-temporal bridge. However, increasing N beyond 32 yields marginal gains, suggesting that 32 states provide sufficient representational capacity for achieving global isomorphism. We thus set $N = 32$ to balance modeling precision with computational efficiency.

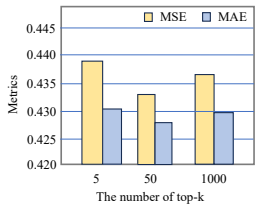
The hyperparameter k regulates the semantic richness of the prototypes used for alignment. Results in Figure 5c and 5d indicate that $k = 5$ provides inadequate linguistic context to reactivate the LLM's reasoning capabilities. While $k = 50$ achieves optimal performance, increasing k to 1000 introduces semantic noise from the emission matrix's long tail, leading to performance stagnation or slight degradation.



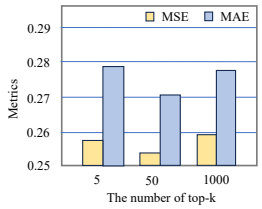
(a) ETTh1: latent states.



(b) Weather: latent states.



(c) ETTh1: top-k tokens.



(d) Weather: top-k tokens.

Figure 5: Hyperparameter sensitivity results on the number of latent states and top-k tokens.

4.6 Computational Cost

Computational cost is a critical factor in assessing the practicality of LLM-based models for time series forecasting. To evaluate this, we compared our MGSAA with state-of-the-art LLM-based models (Time-LLM and FSCA) regarding training and inference costs. For a fair comparison, all baseline models uniformly use a 6-layer GPT-2 backbone to match the architecture of MGSAA. The evaluation was conducted on the ETTh1 dataset with a batch size of 32, using a single NVIDIA L40s GPU.

Table 5: Computational cost comparison of LLM-based models.

Models	MGSAA	Time-LLM	FSCA
Total Params (M)	**87.20**	148.89	89.28
Trainable Params (M)	**4.06**	52.80	8.17
Training Time / Step (s)	**0.014**	0.058	0.070
Inference Time / Batch (s)	**0.004**	0.024	0.026

As summarized in Table 5, MGSAA operates with a significantly lower computational footprint and fewer trainable parameters compared to pure reprogramming method (Time-LLM) and context-alignment method (FSCA). This high efficiency further justifies the effectiveness of our framework. Integrating global structural constraints not only reactivates the LLM for superior predictive accuracy but also substantially reduces the computational cost, by limiting the attention computation to a small set of tokens associated with the decoded state. Furthermore, since the HMM distillation process is a single offline preprocessing step, it introduces zero computational overhead during the actual training and inference phases of downstream tasks.

5 Conclusion

In this work, we propose MGSAA, a novel framework that achieves linguistic-temporal cross-modal alignment that respects global dynamical evolution, grounded in Markovian structure. By transferring language-derived state transition patterns in LLM's latent space to guide the structural representation alignment, MGSAA transforms time series into topologically isomorphic sequences that are inherently interpretable by frozen LLMs. Comprehensive evaluations across diverse scenarios demonstrate our model's superior prediction performance and robust generalization capabilities, establishing a new state-of-the-art approach in time series forecasting.

Limitations

Despite the effectiveness of MGSAA in cross-modal structural alignment, several limitations remain. First, the framework relies on a predefined

number of latent states, N. While an N-state Markovian chain is sufficient for capturing most typical dynamical evolutions, it may encounter granularity bottlenecks when modeling extremely complex non-linear systems. Second, the construction of the dynamical subspace depends on specific linguistic prompt templates. Although the language structures embodied in the current templates capture general time-series dynamic patterns, expanding the template library would enable the framework to better accommodate more diverse and highly stochastic dynamics. Future work will investigate adaptive state selection and automated template generation to enhance topological granularity and linguistic prior diversity.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No.92367205, No. U24A20258) and the Natural Science Foundation of Zhejiang Province (Grant LR26F030002).

References

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, and 34 others. 2021. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramírez, Max Mergenthaler Canseco, and Artur Dubrawski. 2023. NHITS: neural hierarchical interpolation for time series forecasting. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 6989–6997. AAAI Press.

Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. 2023. LLM4TS: two-stage fine-tuning for time-series forecasting with pre-trained llms. *CoRR*, abs/2308.08469.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. 2025. Context-alignment: Activating and enhancing llms capabilities in time series. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Frederick Jelinek. 1998. *Statistical methods for speech recognition*. MIT press.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR.

Yushan Jiang, Zijie Pan, Xikun Zhang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. 2024. Empowering time series analysis with large language models: A survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 8095–8103. ijcai.org.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth*

International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2022. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR.

Anji Liu, Honghua Zhang, and Guy Van den Broeck. 2023. Scaling up probabilistic circuits by latent variable distillation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. 2025a. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 18780–18788. AAAI Press.

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

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. 2025c. CALF: aligning llms for time series forecasting via cross-modal fine-tuning. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 18915–18923. AAAI Press.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2024. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2018. The m4 competition: Results, findings, conclusion and way forward. *International Journal of forecasting*, 34(4):802–808.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2020. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zijie Pan, Yushan Jiang, Sahil Garg, Anderson Schneider, Yuriy Nevmyvaka, and Dongjin Song. 2024. S2IP-LLM: semantic space informed prompt learning with LLM for time series forecasting. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Jianyang Qin, Chaoyang Li, Jinhao Cui, Lingzhi Wang, Zhao Liu, and Qing Liao. 2025. Bridging time and linguistics: LLMs as time series analyzer through symbolization and segmentation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Lawrence R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286.

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. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Junhong Shen, Liam Li, Lucio M. Dery, Corey Staten, Mikhail Khodak, Graham Neubig, and Ameet Talwalkar. 2023. Cross-modal fine-tuning: Align then refine. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31030–31056. PMLR.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. 2024. TEST: text prototype aligned embedding to activate llm's ability for time series. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. 2024. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 22419–22430.

Hao Xue and Flora D. Salim. 2024. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Trans. Knowl. Data Eng.*, 36(11):6851–6864.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *Trans. Mach. Learn. Res.*, 2022.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting? In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 11121–11128. AAAI Press.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 2114–2124. ACM.

Honghua Zhang, Meihua Dang, Nanyun Peng, and Guy Van den Broeck. 2023. Tractable control for autoregressive language generation. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 40932–40945. PMLR.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 27268–27286. PMLR.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023. One fits all: Power general time series analysis by pretrained LM. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A Dataset Details

All datasets employed in this study are publicly available and can be accessed through open repositories.

A.1 Long-term Forecasting

We conduct comprehensive evaluations on seven widely-used time series datasets for long-term forecasting. Following established protocols in prior work (Wu et al., 2021), each dataset is split chronologically into training, validation, and test sets. Specifically, we adopt a 6:2:2 split for the ETT dataset and a 7:1:2 split for all others. The details of the datasets are as follows:

1. **ETT** (Electricity Transformer Temperature): This dataset contains measurements of power load and oil temperature from electrical transformers in two Chinese regions between 2016 and 2018. It is available in two resolutionsETTh (hourly) and ETTm (15-minute).
2. **Weather**: This dataset records 21 meteorological variables measured every 10 minutes across Germany throughout 2020, providing a comprehensive representation of local weather dynamics.

3. **Electricity**: This dataset reports hourly electricity consumption (in kWh) from 321 clients, collected between 2012 and 2014.
4. **Traffic**: Comprising hourly road occupancy data from 862 sensors deployed on San Francisco Bay Area freeways, this dataset spans 2015 to 2016 and reflects regional traffic patterns and congestion trends.

A.2 Short-term Forecasting

We use the M4 benchmark in our short-term forecasting experiments, which comprises 100,000 time series collected from diverse domains. These time series are categorized into six subsets based on their sampling frequencies, ranging from yearly to hourly. The detailed statistics for each subset are summarized in Table 6.

B Implementation Details

To ensure the reproducibility of our results, we provide the comprehensive experimental configurations and hyperparameter settings for the MGSAA framework below, and all experiments are conducted on a single NVIDIA L40s GPU (48GB memory).

B.1 Architecture Configurations

We utilize the pre-trained GPT-2 model as our generative backbone, specifically selecting the first six Transformer layers to balance reasoning capacity and computational efficiency. For the temporal feature extraction, we employ a two-layer Transformer encoder with a hidden dimensionality of $d = 128$. In the State-Conditioned Semantic Alignment module, the top-k value for state-specific token retrieval is set to 50. To mitigate overfitting and enhance model robustness, a dropout rate of 0.1 is applied across all modules.

B.2 Task-Specific Settings

Long-term Forecasting: All datasets are segmented into patches with a length $l_p = 16$ and a stride $s_p = 8$. The number of latent states for Markovian distillation is set to $N = 32$. We utilize the L_1 loss as the training objective for ETT, Weather, and Electricity datasets, while Smooth L_1 loss is adopted for the Traffic dataset.

Short-term Forecasting: For the M4 benchmark with diverse sampling frequencies, we apply a more granular patching strategy with $l_p = 4$ and $s_p = 2$. The latent state space is configured with $N = 32$ states. The training is guided by the SMAPE loss function to align with the evaluation metrics of the M4 competition.

C More Experiments

C.1 The Details of Dynamical Subspace Construction

To ensure that the distilled Markovian prior reflects generic smoothness and continuity, we construct the Dynamical Subspace $\mathcal{D}$ using a systematic prompt engineering and generation pipeline. The implementation details are as follows:

Prompt Template Design We utilize a combinatorial approach to generate 800 unique prompts across four abstract scientific scenarios. Each category is formed by taking the Cartesian product of predefined subjects, verbs, and contextual patterns to ensure a high degree of logical coherence within the LLM's latent space:

- **Periodic and Oscillatory Scenarios:**
  - **Subjects:** Pendulum, wave function, oscillating particle, etc.
  - **Verbs:** Oscillated, rotated, cycled, fluctuated, etc.
  - **Patterns:** Along a sinusoidal path, around the equilibrium point, etc.
- **Gradual Trend Evolution:**
  - **Subjects:** Temperature, population, system entropy, stock price, etc.
  - **Verbs:** Increased linearly, decayed exponentially, stabilized gradually, etc.
  - **Contexts:** Over the observed period, reaching a critical threshold, etc.
- **Causal and Conditional Reasoning:**
  - **Starts:** Given the initial condition, assuming the system is stable, etc.
  - **Actions:** The next state becomes, the variable transitions to, etc.
  - **Targets:** A higher energy level, the adjacent node, etc.
- **Continuous Geometric Trajectories:**
  - **Fragments:** Continuously evolving from, smoothly transitioning into, etc.
  - **Objects:** The initial phase, the linear segment, etc.
  - **Results:** The trajectory forms a curve, the manifold remains smooth, etc.

Table 6: Statistics of the M4 dataset.

Dataset	Time Steps	Frequency	Domain
M4-Yearly	23,000	Yearly	Demographic
M4-Quarterly	24,000	Quarterly	Finance
M4-Monthly	48,000	Monthly	Industry
M4-Weekly	359	Weekly	Macro
M4-Daily	4,227	Daily	Micro
M4-Hourly	414	Hourly	Other

Causal Generation and Representation Extraction Using the curated prompts, we sample 10,000 text sequences using a 6-layer GPT-2 model. The generation and extraction process is executed as follows: To prevent identical trajectories from identical prompts, we employ random sampling with $top\_p = 0.9$ and $temperature = 0.7$. Each sequence is limited to a maximum of 32 new tokens. At each generation step t, we extract the context-aware hidden representation from layer 6 of the GPT-2 backbone. And we specifically capture the representation of the latest generated token before the next sampling step. This ensures that each embedding encapsulates the entire preceding sequence context without looking ahead at future tokens, strictly adhering to the causal nature of temporal processes.

C.2 LLM Latent Variable Clustering Results

To ensure visual clarity while maintaining the diversity of the underlying topological structure, we project a representative subset consisting of 20 salient clusters into a low-dimensional manifold. We sample 60,000 token embeddings from these typical states and apply t-SNE to map the 768-D embeddings onto a 2-D plane. As shown in Figure 6, the context-aware representations exhibit clear spatial grouping. Tokens associated with the same latent state are highly concentrated, forming distinct regions on the manifold. Despite the clear separation, adjacent clusters often exhibit close proximity or shared boundaries. This spatial arrangement reflects the smooth transition between different semantic states in the linguistic domain, which provides a natural inductive bias for modeling the continuous dynamics of physical time series.

C.3 Testing with a larger LLM

To verify that our MGSAA framework can seamlessly scale and benefit from larger LLMs, we conducted additional experiments using the LLaMA-2(7B) backbone and compared it with Time-LLM (also equipped with LLaMA-2). As shown in the table below, when using LLaMA-2, MGSAA maintains highly competitive forecasting performance and consistently, significantly outperforms Time-LLM across all prediction lengths. This confirms that our methodology is not limited to the specific scale of GPT-2, but can robustly adapt to more powerful LLM architectures, continuously surpassing pure reprogramming methods under identical conditions.

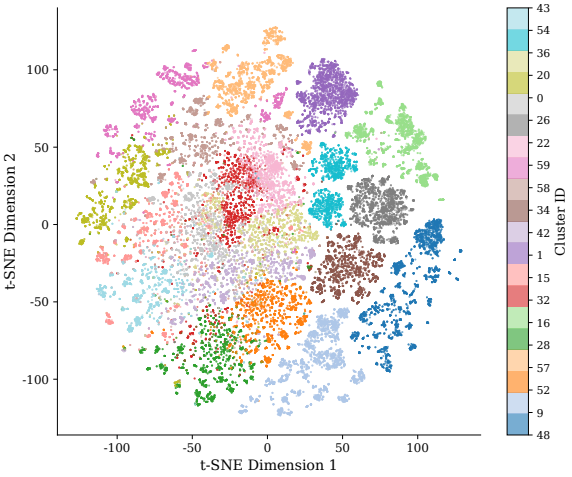


Figure 6: The result of LLM latent variable clustering.

Table 7: Performance comparison on ETT1 and ETT2 datasets using the LLaMA-2(7B) backbone.

Models	Metrics	ETT1			ETT2		
		96	192	336	96	192	336
Time-LLM(8)	MSE	0.383	0.440	0.480	0.316	0.397	0.430
	MAE	0.408	0.452	0.492	0.360	0.407	0.437
MGSAA(8)	MSE	**0.373**	**0.425**	**0.467**	**0.295**	**0.370**	**0.410**
	MAE	**0.391**	**0.423**	**0.442**	**0.339**	**0.386**	**0.427**

D Full Results

Table 8: Full results for long-term forecasting with different prediction lengths $H \in \{96, 192, 336, 720\}$. The input sequence length for all baseline methods is set to 96. Avg. denotes the average over all prediction lengths. **Bold**: best, underlined: second best.

Dataset	Horizon	MGSAA		FSCA		TimeLLM		GPT4TS		PatchTST		iTransformer		FEDformer		Stationary		Autoformer		TimesNet		DLinear	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.313**	**0.341**	0.331	0.368	0.331	0.370	0.330	0.365	0.324	0.365	0.341	0.376	0.366	0.411	0.428	0.428	0.558	0.499	0.334	0.375	0.346	0.374
	192	**0.359**	**0.370**	0.389	0.384	0.370	0.389	0.368	0.383	0.372	0.392	0.381	0.395	0.434	0.446	0.498	0.457	0.606	0.511	0.407	0.413	0.382	0.391
	336	**0.388**	**0.391**	0.407	0.406	0.411	0.413	0.401	0.404	0.398	0.409	0.419	0.419	0.476	0.472	0.584	0.498	0.714	0.568	0.415	0.422	0.415	0.415
	720	0.454	**0.431**	0.472	0.449	0.458	0.441	0.461	**0.439**	0.457	0.444	0.486	0.456	0.507	0.490	0.608	0.529	0.578	0.527	0.486	0.461	0.473	0.451
	Avg	**0.378**	**0.383**	0.400	0.402	0.392	0.403	0.390	0.398	0.388	0.402	0.407	0.411	0.446	0.455	0.529	0.478	0.614	0.526	0.410	0.418	0.404	0.408
ETTm2	96	**0.176**	**0.251**	0.181	0.263	0.181	0.267	0.178	0.264	0.186	0.268	0.184	0.267	0.191	0.282	0.280	0.328	0.253	0.323	0.190	0.267	0.193	0.293
	192	**0.239**	**0.294**	0.245	0.304	0.252	0.311	0.245	0.306	0.246	0.305	0.253	0.312	0.263	0.326	0.682	0.497	0.280	0.340	0.253	0.307	0.285	0.361
	336	0.307	**0.339**	**0.304**	0.340	0.315	0.352	0.309	0.346	0.311	0.348	0.315	0.352	0.331	0.367	0.467	0.418	0.332	0.367	0.321	0.349	0.385	0.429
	720	**0.399**	**0.396**	0.401	0.398	0.415	0.407	0.404	0.401	0.418	0.413	0.412	0.406	0.427	0.423	0.639	0.509	0.462	0.441	0.418	0.405	0.556	0.523
	Avg	**0.280**	**0.320**	0.283	0.326	0.291	0.334	0.284	0.329	0.290	0.333	0.291	0.334	0.303	0.349	0.517	0.438	0.332	0.368	0.295	0.332	0.355	0.401
ETTh1	96	**0.371**	**0.387**	0.389	0.410	0.394	0.407	0.377	0.398	0.377	0.397	0.395	0.410	**0.376**	0.416	0.505	0.482	0.464	0.466	0.389	0.412	0.396	0.411
	192	0.430	**0.421**	0.445	0.438	0.446	0.437	0.439	0.427	0.426	0.432	0.449	0.441	**0.419**	0.442	0.615	0.553	0.513	0.486	0.439	0.442	0.445	0.440
	336	0.469	**0.440**	**0.455**	0.449	0.485	0.459	0.480	0.447	0.469	0.457	0.492	0.465	0.457	0.465	0.754	0.632	0.518	0.499	0.494	0.471	0.487	0.465
	720	**0.461**	**0.459**	0.464	0.469	0.493	0.484	0.496	0.477	0.519	0.504	0.522	0.504	0.504	0.506	0.642	0.574	0.510	0.509	0.518	0.494	0.513	0.510
	Avg	**0.433**	**0.427**	0.438	0.441	0.454	0.447	0.448	0.437	0.448	0.447	0.464	0.455	0.439	0.457	0.629	0.560	0.501	0.490	0.460	0.455	0.460	0.456
ETTh2	96	**0.285**	**0.331**	0.306	0.358	0.298	0.346	0.294	0.347	0.309	0.359	0.300	0.350	0.349	0.389	0.403	0.423	0.373	0.408	0.337	0.371	0.341	0.395
	192	**0.362**	**0.381**	0.380	0.403	0.389	0.401	0.386	0.404	0.381	0.407	0.382	0.400	0.440	0.447	0.546	0.497	0.457	0.454	0.405	0.415	0.482	0.479
	336	0.408	**0.421**	**0.401**	0.427	0.431	0.438	0.423	0.435	0.412	0.429	0.424	0.432	0.499	0.490	0.622	0.530	0.476	0.478	0.450	0.449	0.592	0.542
	720	**0.424**	**0.436**	0.434	0.449	0.431	0.446	0.424	0.447	0.436	0.456	0.426	0.445	0.482	0.487	0.606	0.542	0.526	0.509	0.435	0.448	0.840	0.661
	Avg	**0.370**	**0.392**	0.380	0.409	0.387	0.408	0.382	0.408	0.384	0.413	0.383	0.407	0.442	0.453	0.544	0.498	0.458	0.462	0.407	0.421	0.564	0.519
Weather	96	0.171	**0.206**	0.178	0.217	0.195	0.234	0.182	0.224	0.176	0.218	0.175	0.216	0.221	0.300	0.180	0.230	0.329	0.368	**0.169**	0.219	0.197	0.256
	192	**0.217**	**0.250**	0.226	0.258	0.239	0.269	0.232	0.263	0.221	0.256	0.225	0.258	0.279	0.339	0.237	0.280	0.349	0.398	0.225	0.365	0.239	0.295
	336	**0.272**	**0.289**	0.281	0.298	0.291	0.306	0.283	0.299	0.280	0.298	0.280	0.298	0.343	0.385	0.318	0.334	0.369	0.400	0.282	0.304	0.283	0.337
	720	**0.351**	**0.341**	0.357	0.348	0.366	0.353	0.360	0.349	0.356	0.348	0.361	0.351	0.408	0.432	0.418	0.394	0.458	0.460	0.359	0.354	0.346	0.383
	Avg	**0.253**	**0.271**	0.260	0.280	0.273	0.290	0.264	0.284	0.258	0.280	0.260	0.281	0.313	0.364	0.288	0.309	0.376	0.406	0.259	0.285	0.266	0.318
Electricity	96	0.164	0.249	0.171	0.255	0.169	0.259	0.186	0.272	0.180	0.273	**0.148**	**0.240**	0.195	0.309	0.167	0.268	0.209	0.324	0.169	0.272	0.197	0.285
	192	0.173	**0.256**	0.181	0.266	0.173	0.266	0.189	0.277	0.187	0.280	**0.165**	**0.256**	0.202	0.315	0.183	0.283	0.222	0.332	0.187	0.289	0.202	0.288
	336	0.189	0.273	0.196	0.281	0.197	0.289	0.205	0.291	0.204	0.296	**0.178**	**0.270**	0.227	0.340	0.204	0.308	0.246	0.351	0.199	0.299	0.215	0.304
	720	0.229	0.306	0.234	0.313	0.249	0.328	0.244	0.324	0.246	0.328	**0.209**	**0.299**	0.260	0.363	0.222	0.320	0.276	0.376	0.232	0.329	0.248	0.337
	Avg	0.189	0.271	0.195	0.279	0.197	0.285	0.206	0.291	0.204	0.294	**0.175**	**0.266**	0.221	0.332	0.194	0.295	0.238	0.346	0.197	0.297	0.215	0.303
Traffic	96	0.412	**0.249**	0.432	0.281	0.487	0.307	0.468	0.308	0.459	0.298	**0.392**	0.268	0.581	0.363	0.625	0.349	0.670	0.401	0.589	0.315	0.649	0.397
	192	0.425	**0.263**	0.448	0.287	0.499	0.318	0.477	0.311	0.469	0.301	**0.412**	0.277	0.606	0.379	0.645	0.356	0.646	0.413	0.617	0.324	0.598	0.370
	336	0.443	**0.270**	0.462	0.295	0.516	0.329	0.489	0.317	0.483	0.307	**0.424**	0.283	0.612	0.380	0.644	0.355	0.609	0.382	0.635	0.338	0.605	0.373
	720	0.476	**0.289**	0.495	0.313	0.535	0.343	0.522	0.333	0.517	0.326	**0.460**	0.301	0.641	0.393	0.659	0.359	0.650	0.402	0.660	0.349	0.646	0.394
	Avg	0.439	**0.268**	0.459	0.294	0.509	0.324	0.489	0.317	0.482	0.308	**0.422**	0.282	0.610	0.379	0.643	0.355	0.644	0.399	0.625	0.331	0.624	0.383

Table 9: Full results for short-term forecasting on the M4 dataset. The input and prediction lengths are set to $[12, 96]$ and $[6, 48]$, respectively. The reported average represents the weighted average results across different datasets. **Bold**: best, underlined: second best.

Models		MGSAA	FSCA	TimeLLM	GPT4TS	PatchTST	iTransformer	FEDformer	Stationary	Autoformer	TimesNet	DLinear	N-HiTS	N-BEATS
Yearly	SMAPE	**13.186**	13.288	13.750	14.822	13.477	13.652	14.021	14.727	13.974	15.378	16.965	13.422	13.487
	MASE	**2.960**	2.974	3.055	3.618	3.019	3.095	3.036	3.078	3.134	3.554	4.283	3.056	3.036
	OWA	**0.776**	0.781	0.805	0.909	0.792	0.807	0.811	0.807	0.822	0.918	1.058	0.795	0.795
Quarterly	SMAPE	**10.029**	10.037	10.671	10.411	10.380	10.353	11.100	10.958	11.338	10.465	12.145	10.185	10.564
	MASE	**1.169**	1.174	1.276	1.232	1.233	1.209	1.350	1.325	1.365	1.227	1.520	1.180	1.252
	OWA	**0.882**	0.884	0.950	0.922	0.921	0.911	0.996	0.981	1.012	0.923	1.106	0.893	0.936
Monthly	SMAPE	**12.568**	12.762	13.416	12.902	12.959	13.079	14.403	13.917	13.958	13.513	13.514	13.059	13.089
	MASE	**0.928**	0.947	1.045	0.956	0.970	0.974	1.147	1.097	1.103	1.039	1.037	1.013	0.996
	OWA	**0.872**	0.897	0.957	0.897	0.905	0.911	1.038	0.998	1.002	0.957	0.956	0.929	0.922
Others	SMAPE	**4.688**	4.761	4.973	5.294	4.952	4.780	7.148	6.302	5.485	6.913	6.709	4.711	6.599
	MASE	3.215	3.207	3.412	3.610	3.347	3.231	4.064	4.064	3.865	4.507	4.953	**3.054**	4.430
	OWA	1.000	1.007	1.053	1.126	1.049	1.012	1.304	1.304	1.187	1.438	1.487	**0.977**	1.393
Average	SMAPE	**11.707**	11.828	12.494	12.365	12.059	12.142	13.160	12.780	12.909	12.880	13.639	12.035	12.250
	MASE	**1.567**	1.580	1.731	1.767	1.623	1.631	1.775	1.756	1.771	1.836	2.095	1.625	1.698
	OWA	**0.841**	0.850	0.913	0.918	0.869	0.874	0.949	0.930	0.939	0.955	1.051	0.869	0.896

Table 10: Full results for few-term forecasting on 10% training data of ETT datasets with different prediction lengths $H \in \{96, 192, 336, 720\}$. The input sequence length for all baseline methods is set to 96. Avg. denotes the average over all prediction lengths. **Bold**: best, underlined: second best.

Dataset	Horizon	MGSAA		FSCA		TimeLLM		GPT4TS		PatchTST		iTransformer		FEDformer		Stationary		TimesNet		DLinear	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	96	**0.457**	**0.415**	0.573	0.494	0.571	0.494	0.613	0.494	0.654	0.509	0.571	0.487	0.602	0.523	1.032	0.649	0.585	0.504	0.552	0.488
	192	**0.476**	**0.430**	0.574	0.487	0.509	0.457	0.596	0.489	0.617	0.481	0.557	0.480	0.641	0.545	1.025	0.656	0.609	0.516	0.546	0.487
	336	**0.512**	**0.452**	0.618	0.509	0.576	0.498	0.595	0.499	0.575	0.488	0.582	0.503	0.769	0.606	1.040	0.682	0.736	0.573	0.567	0.501
	720	**0.577**	**0.485**	0.644	0.521	0.606	0.517	0.623	0.511	0.631	0.507	0.612	0.518	0.772	0.605	1.182	0.734	0.768	0.548	0.606	0.523
	Avg	**0.505**	**0.445**	0.602	0.503	0.565	0.491	0.607	0.498	0.619	0.496	0.580	0.497	0.696	0.570	1.070	0.680	0.674	0.535	0.568	0.500
ETTm2	96	0.187	**0.261**	0.189	0.268	0.190	0.271	**0.186**	0.267	0.193	0.274	0.195	0.277	0.223	0.314	0.253	0.323	0.215	0.289	0.225	0.320
	192	0.255	**0.305**	0.266	0.320	0.258	0.314	**0.253**	0.308	0.254	0.313	0.262	0.318	0.285	0.352	0.327	0.363	0.271	0.325	0.291	0.362
	336	0.325	0.350	**0.317**	**0.348**	0.323	0.357	0.333	0.354	0.326	0.361	0.325	0.355	0.392	0.420	0.438	0.424	0.330	0.357	0.354	0.403
	720	**0.421**	**0.407**	0.437	0.420	0.442	0.421	0.439	0.419	0.436	0.419	0.444	0.425	0.533	0.489	0.694	0.545	0.473	0.448	0.447	0.448
	Avg	**0.297**	**0.331**	0.302	0.339	0.303	0.341	0.303	0.337	0.302	0.342	0.306	0.344	0.358	0.394	0.428	0.414	0.322	0.355	0.329	0.383
ETTh1	96	0.489	0.458	0.523	0.473	0.543	0.486	0.462	0.446	**0.455**	**0.444**	0.623	0.530	0.650	0.562	0.784	0.614	0.856	0.626	0.589	0.515
	192	0.545	**0.487**	0.552	0.491	0.566	0.499	0.549	0.495	**0.531**	**0.488**	0.694	0.562	0.674	0.569	0.723	0.577	0.786	0.587	0.631	0.540
	336	**0.587**	**0.513**	0.615	0.525	0.606	0.521	0.628	0.539	0.624	0.539	0.768	0.593	0.771	0.606	0.860	0.607	0.943	0.649	0.659	0.555
	720	0.772	0.621	1.089	0.724	0.750	0.615	1.113	0.738	0.881	0.657	0.909	0.662	0.901	0.695	1.014	0.725	0.872	0.639	**0.710**	**0.600**
	Avg	**0.616**	**0.528**	0.695	0.553	**0.616**	0.530	0.688	0.554	0.623	0.532	0.748	0.587	0.749	0.608	0.845	0.631	0.864	0.625	0.647	0.552
ETTh2	96	0.316	0.358	**0.305**	**0.347**	0.319	0.358	0.321	0.357	0.315	0.361	0.340	0.377	0.359	0.404	0.410	0.411	0.367	0.401	0.361	0.407
	192	0.394	0.404	**0.392**	**0.399**	0.436	0.435	0.399	0.403	0.467	0.441	0.438	0.432	0.461	0.461	0.523	0.478	0.508	0.475	0.445	0.453
	336	0.479	**0.453**	**0.460**	0.456	0.516	0.481	0.562	0.499	0.553	0.490	0.495	0.474	0.581	0.536	0.695	0.561	0.558	0.504	0.521	0.508
	720	0.465	0.468	**0.454**	**0.462**	0.716	0.587	1.013	0.721	0.624	0.545	0.501	0.488	0.812	0.698	1.461	0.901	0.514	0.494	0.457	0.475
	Avg	0.413	0.421	**0.403**	**0.416**	0.497	0.465	0.574	0.495	0.490	0.459	0.443	0.443	0.553	0.525	0.772	0.588	0.487	0.468	0.446	0.461

Table 11: Full results for zero-shot learning on ETT datasets with different prediction lengths $H \in \{96, 192, 336, 720\}$. 'h1', 'h2', 'm1', and 'm2' denote ETTh1, ETTh2, ETTm1, and ETTm2. A $\rightarrow$ B indicates training on dataset A and testing on dataset B. The input sequence length for all baseline methods is set to 96. **Bold**: best, underlined: second best.

Dataset	Horizon	MGSAA		FSCA		TimeLLM		GPT4TS		PatchTST		iTransformer		FEDformer		Stationary		TimesNet		DLinear	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
h1 → h2	96	0.308	0.346	**0.284**	**0.335**	0.300	0.343	0.301	0.345	0.302	0.346	0.301	0.348	0.367	0.410	0.487	0.474	0.333	0.373	0.313	0.370
	192	0.379	0.391	**0.363**	**0.386**	0.386	0.397	0.382	0.392	0.380	0.396	0.380	0.395	0.446	0.456	0.551	0.503	0.409	0.412	0.413	0.433
	336	0.418	0.424	**0.373**	**0.405**	0.429	0.435	0.420	0.427	0.423	0.431	0.421	0.430	0.482	0.490	0.723	0.579	0.478	0.461	0.490	0.485
	720	0.420	0.445	**0.406**	**0.428**	0.424	0.440	0.424	0.439	0.428	0.446	0.433	0.447	0.526	0.527	0.656	0.566	0.475	0.469	0.646	0.576
	Avg	0.381	0.401	**0.356**	**0.388**	0.385	0.404	0.382	0.401	0.383	0.405	0.384	0.405	0.455	0.471	0.604	0.530	0.424	0.429	0.465	0.466
h2 → h1	96	**0.456**	**0.446**	0.513	0.498	0.508	0.470	0.484	0.454	0.600	0.530	0.590	0.523	0.679	0.575	1.178	0.733	0.749	0.601	0.466	0.456
	192	0.539	0.496	0.527	0.498	0.534	0.486	0.578	0.514	0.750	0.593	0.668	0.565	0.780	0.618	1.430	0.812	0.802	0.619	**0.496**	**0.480**
	336	0.573	**0.508**	0.554	0.527	0.583	0.512	0.584	0.516	0.713	0.581	0.698	0.581	0.747	0.612	1.685	0.839	0.917	0.668	**0.544**	0.509
	720	**0.566**	**0.524**	0.634	0.579	0.577	0.529	0.593	0.543	0.912	0.676	0.699	0.599	0.721	0.615	3.349	1.070	1.087	0.699	0.570	0.550
	Avg	0.533	**0.493**	0.557	0.525	0.550	0.499	0.560	0.507	0.744	0.595	0.664	0.567	0.727	0.605	1.910	0.863	0.889	0.647	**0.519**	0.499
m1 → m2	96	**0.193**	**0.265**	**0.193**	0.271	0.206	0.268	0.206	0.275	0.199	0.275	0.203	0.281	0.327	0.415	0.256	0.326	0.230	0.307	0.220	0.313
	192	0.258	**0.308**	**0.256**	0.309	0.269	**0.308**	0.268	0.314	0.262	0.314	0.263	0.317	0.396	0.453	0.430	0.428	0.344	0.376	0.283	0.355
	336	0.328	0.354	**0.315**	**0.347**	0.327	0.445	0.326	0.352	0.319	0.349	0.319	0.351	0.436	0.467	0.506	0.470	0.361	0.383	0.359	0.406
	720	0.437	0.418	**0.410**	0.401	0.426	**0.400**	0.421	0.403	0.425	0.411	0.422	0.406	0.526	0.510	0.604	0.521	0.460	0.435	0.477	0.476
	Avg	0.304	0.336	**0.293**	**0.332**	0.307	0.355	0.305	0.336	0.301	0.337	0.302	0.339	0.421	0.461	0.449	0.436	0.349	0.375	0.335	0.387
m2 → m1	96	**0.493**	**0.420**	0.525	0.450	0.530	0.459	0.527	0.459	0.599	0.471	0.664	0.507	0.808	0.622	1.288	0.671	0.871	0.600	0.534	0.452
	192	**0.516**	**0.442**	0.559	0.486	0.583	0.485	0.568	0.488	0.562	0.482	0.702	0.536	0.795	0.612	1.347	0.707	0.724	0.558	0.555	0.471
	336	0.606	0.500	0.605	0.518	0.619	0.513	0.590	0.504	**0.559**	**0.495**	0.834	0.589	0.791	0.610	1.343	0.725	0.772	0.586	0.586	**0.495**
	720	**0.604**	**0.521**	0.674	0.555	0.698	0.552	0.644	0.542	0.853	0.621	0.815	0.592	0.805	0.616	2.049	0.853	0.998	0.650	0.617	**0.521**
	Avg	**0.555**	**0.471**	0.591	0.502	0.607	0.502	0.562	0.498	0.643	0.517	0.754	0.556	0.800	0.615	1.507	0.739	0.841	0.598	0.573	0.485