SSDA: BRIDGING SPECTRAL AND STRUCTURAL GAPS VIA DUAL ADAPTATION FOR VISION-BASED TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision models (LVMs) have recently proven to be surprisingly effective time series forecasters, simply by rendering temporal data as images. This success, however, rests on a largely unexamined premise: the rendered time series images are sufficiently close to natural images for knowledge in pre-trained models to transfer effectively. We argue that two gaps still remain, i.e., **spectral and structural gaps**, fundamentally limiting the potential of LVMs for time series forecasting. *Spectrally*, we systematically reveal that rendered time series images exhibit a markedly shallower power spectrum than the natural images LVMs are pre-trained to recognize. *Structurally*, reshaping 1D temporal sequences into 2D grids fabricates spurious spatial adjacencies while severing genuine temporal continuities, misleading the spatial inductive biases of pre-trained LVMs. To bridge these gaps, we propose **SSDA**, a dual-branch network that spectrally and structurally adapts to unlock the full potential of LVMs for time series forecasting. At the data level, a **Spectral Magnitude Aligner (SMA)** applies 2D FFT to selectively enhance the magnitude spectrum toward natural-image statistics while preserving phase. At the model level, a **Structural-Guided Low-Rank Adaptation (SG-LoRA)** injects position-aware temporal encodings into patch embeddings and adapts attention via low-rank updates. The two branches are further adaptively fused to produce the final forecast. Extensive experiments on seven real-world benchmarks demonstrate that SSDA consistently outperforms strong LVM- and LLM-based baselines under both full-shot and few-shot settings. Code is publicly available at `https://anonymous.4open.science/r/SSDA-8C5B`.

1 INTRODUCTION

Time series forecasting (TSF) is central to decision-making across a wide range of domains, including energy Fei et al. (2025), transportation Yang et al. (2025a), healthcare Chen et al. (2024), climate science Zhang et al. (2026), and economics Yu et al. (2023). Inspired by the success of foundation models in natural language processing (NLP) and computer vision (CV), a growing body of work adapts large pre-trained backbones, such as GPT-2, BERT, CLIP, and LLaMA, into universal forecasters Liang et al. (2024). Such backbones offer rich, transferable representations learned from large-scale data, motivating their reuse as general feature encoders. Early efforts, e.g., GPT4TS, UniTS, and Time-LLM, treat time series as text and feed them into large language models (LLMs), achieving superior gains over traditional methods Goswami et al. (2024); Niu et al. (2023); Zhou et al. (2023); Jin et al. (2024); You et al. (2026). More recently, large vision models (LVMs) have emerged as an especially promising direction. By rendering 1D time series as 2D images and processing them with a pre-trained vision encoder, methods such as VisionTS Chen et al. (2025) and DMMV Shen et al. (2025a) outperform LLM-based counterparts on many TSF benchmarks Ni et al. (2025); Kim et al. (2021). The rationale is intuitive: while both text and images differ from time series in modality, images align with the continuous structure of time series far more naturally. Tokenizing continuous values into discrete symbols, as required by LLM-based approaches, is inherently lossy and disrupts numerical continuity. Rendering time series as 2D images, by contrast, turns trends, periodicity, and fluctuations into the form of gradients, textures, and edges that pre-trained LVMs are already trained to recognize Chen et al. (2025).

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.

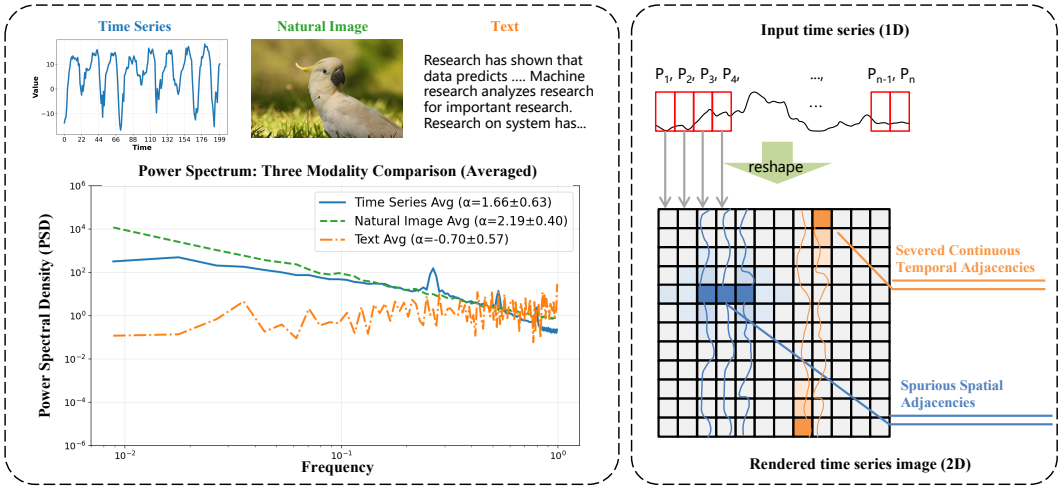


Figure 1: Illustration of the spectral and structural modality gaps. **Left:** Spectral gap. Top: one representative sample from three modalities, i.e., a time series from ETT, a natural image from ImageNet, and a text snippet from Wikipedia. Bottom: power spectra averaged over a large set of samples per modality, fitted with the power-law model $P(f) \propto f^{-\alpha}$. The fitted exponents Time Series ($\alpha = 1.66 \pm 0.63$), Natural Images ($\alpha = 2.19 \pm 0.40$), and Text ($\alpha = -0.70 \pm 0.57$) reveal substantial distributional differences across modalities. **Right:** Structural gap. Reshaping a 1D time series into a 2D image via periodic folding introduces spurious spatial adjacencies between temporally distant points and severs continuous temporal adjacencies at column boundaries.

However, the success of LVMs relies on a largely unexamined assumption: the rendered time series images are sufficiently similar to natural images for the pre-trained knowledge of LVMs to transfer effectively. Existing LVM-based forecasters typically resort to ad hoc strategies: directly feeding rendered time series images into a frozen Masked Autoencoder (MAE) Feichtenhofer et al. (2022), fusing them with textual or numerical views Zhong et al. (2025), or filtering training data to mitigate severe mismatches Shen et al. (2025b). None of these approaches, however, explicitly answer the fundamental question: *how far do rendered time series images deviate from natural images, and along which dimensions do such gaps hinder the reuse of LVM pre-training?* Without this understanding, efforts to adopt LVMs for time series forecasting remain largely heuristic and the full potential of LVMs remains out of reach.

We argue that this modality gap has two distinct facets: a **spectral gap** in the input data distribution and a **structural gap** in how time series are rendered to images.

The **spectral gap** concerns the low-level statistics of the image itself. Natural images are well known to follow a $1/f^{\alpha}$ power spectrum with $\alpha \approx 2.0$ Field (1987); Ruderman & Bialek (1993), a regularity that LVM pre-training implicitly exploits. To probe whether rendered time series images share this property, we adopt the Power Spectrum Slope (PSS) Zang et al. (2022); An et al. (2026), a compact frequency-domain descriptor that has been used to characterize signals across modalities Zhong et al. (2025); Zeng et al. (2023b); Lo et al. (2024); Nguyen et al. (2026). As shown in the left panel of Figure 1, samples from the ETT dataset exhibit an average power spectrum slope of $\alpha \approx 1.66$, significantly lower than that of natural images ($\alpha \approx 2.19$), while text rendered as a visual signal yields $\alpha \approx -0.70$. The three modalities thus occupy distinctly different regions of frequency space: the gap between rendered time series and natural images ($\Delta\alpha \approx 0.53$) is substantially smaller than that between time series and text ($\Delta\alpha \approx 2.36$), confirming that the visual modality aligns with time series far more closely than text. To our knowledge, this is the first cross-modal spectral comparison of time series, images, and text in the context of LVM-based forecasting. Full details are given in Appendix A.

The **structural gap** arises in how the model constructs and captures relations between patches. As illustrated in the right panel of Figure 1, reshaping a 1D sequence into a 2D grid via periodic folding creates two simultaneous distortions. First, it fabricates spurious spatial adjacencies. Existing methods usually arrange consecutive time steps as columns Chen et al. (2025), but pixels in the same

row yet different columns are treated as neighbors by the LVM's spatial attention, even though they may be hours or days apart in the original series. Second, it severs genuine temporal adjacencies. The last point of one column and the first point of the next are temporally consecutive yet spatially distant in the rendered image. A pre-trained LVM, whose inductive biases are calibrated to true spatial structure, fails to distinguish these distortions, leading to degraded forecasting performance.

Motivated by the above observations, we propose **SSDA**, a dual-branch network in which each branch is the natural remedy for one gap at the level where that gap actually arises. The **spectral branch** addresses the spectral gap at the *data level* through a **Spectral Magnitude Aligner (SMA)**: rendered images are decomposed by 2D Fast Fourier Transform (FFT), their magnitude spectrum is selectively enhanced by a lightweight learnable module while the phase is preserved to retain temporal structure, and the result is passed to a frozen MAE, maximally preserving the visual knowledge accumulated during pre-training. The **structural branch** addresses the structural gap at the *model level* through **Structural-Guided LoRA (SG-LoRA)**: a Temporal Grounding Adapter (TGA) maps each spatial patch back to its original 1D temporal index, and injects a sinusoidal temporal encoding into the patch embedding via a learnable fusion gate, and LoRA adapters fine-tune the attention projections so that the model reconciles spatial layout with true temporal order. The two branches are fused by a learnable scalar weight and the result is denormalized to produce the final forecast.

Our contributions are threefold:

- We revisit a largely unexamined premise behind LVM-based time series forecasting and present the first systematic characterization of the modality gap between rendered time series images and natural images, identifying two distinct issues: a **spectral** gap in data statistics and a **structural** gap in rendering time series images.
- We propose **SSDA**, a dual-branch network with two complementary adapters: at the data level, the **SMA** reshapes the Fourier magnitude spectrum of rendered images while preserving its phase to retain temporal periodicity; at the model level, **SG-LoRA** injects temporal awareness into attention via low-rank adaptation to restore true continuity.
- Extensive experiments on seven real-world benchmarks demonstrate that SSDA achieves state-of-the-art performance in both full-shot and few-shot settings, with consistent improvements over 11 strong baselines spanning LVM-based, LLM-based, and classical methods.

2 Related Work

2.1 LVM-based Time Series Forecasting

The success of vision models has spurred their application in TSF. Early approaches applied CNNs such as ResNet He et al. (2016) and VGG-19 Simonyan & Zisserman (2015) to time series for feature extraction Li et al. (2020). More recently, large-scale vision models including ViT Dosovitskiy et al. (2021), MAE He et al. (2022), and CLIP Radford et al. (2021) have been adapted for TSF Zhong et al. (2025); Chen et al. (2025); Yang et al. (2025b); Zeng et al. (2023b). For example, VisionTS Chen et al. (2025) applies a frozen MAE to rendered time series images, achieving competitive performance. Time-VLM Zhong et al. (2025) models time series as images and text, encoding these visual and textual views via ViLT to enhance forecasting. DMMV Shen et al. (2025a) leverages MAE's periodicity bias by decomposing series into seasonal and trend components within a multimodal fusion framework for temporal modeling. VisionTS++ Shen et al. (2025b) further introduces continual pre-training on large-scale time series, and proposes targeted strategies to mitigate cross-modal discrepancies, including data filtering, multivariate visual encoding, and quantile-based forecasting. Despite promising results, these methods establish LVMs as effective forecasters, yet they focus on input-side designs such as rendering and fusion, rather than cross-modal adaptation of the LVMs to the time-series modality.

2.2 Cross-modal Adaptation for Time Series Forecasting

Cross-modal adaptation reuses pre-trained models across modalities by aligning data distributions, enabling effective knowledge transfer Shen et al. (2023). Following its success in vision Verma et al. (2024); Lu et al. (2022) and audio Ghosal et al. (2023); Hassid et al. (2023), this paradigm has been extended to TSF along two routes. The language route reprograms continuous signals into

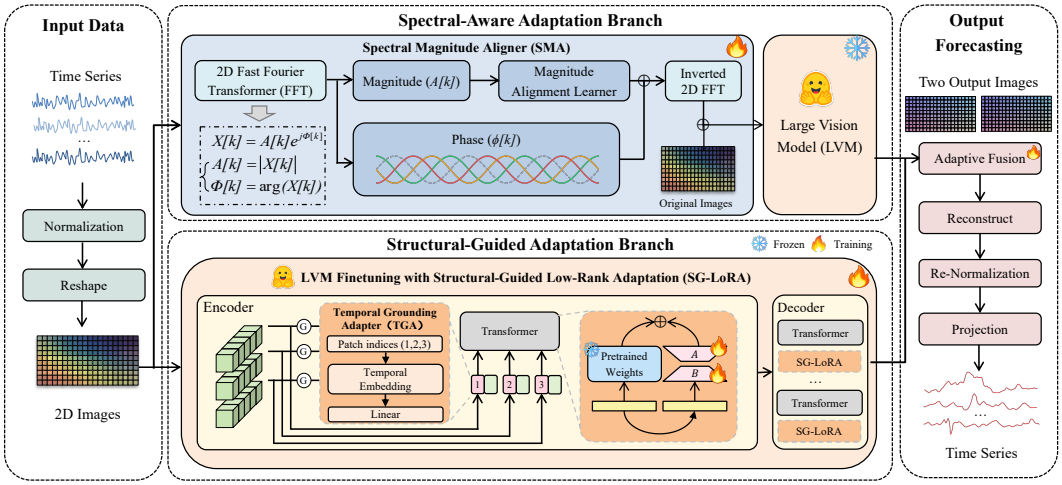


Figure 2: The Overall architecture of SSDA.

discrete tokens for LLM processing, exemplified by Time-LLM Jin et al. (2024), TEST Sun et al. (2024), and TimeCMA Liu et al. (2025); although effective, such tokenization is inherently lossy and disrupts numerical continuity. The vision route is a more natural alternative, since continuous signals align with image patches far better than with discrete tokens, prompting a growing line of LVM adaptations Chen et al. (2025); Zhong et al. (2025); Shen et al. (2025a). However, current LVM adaptations rely on constrained strategies such as freezing the backbone, attaching lightweight adapters, or performing costly continual pre-training Shen et al. (2025b). These strategies adjust what the model sees but not how it reads patches; neither the spectral statistics of the rendered input nor the structural ordering of patches is aligned with the LVM's pre-training distribution, leaving both modality gaps unaddressed.

3 Methodology

3.1 Model Structure

As shown in Figure 2, SSDA processes a normalized, image-rendered time series through two parallel branches that each target one identified gap. The **spectral branch** closes the spectral gap at the *data level*: a **Spectral Magnitude Aligner (SMA)** reshapes the frequency-domain statistics of the rendered image toward those of natural images before forwarding it to a frozen LVM. The **structural branch** closes the structural gap at the *model level*: **Structural-Guided LoRA (SG-LoRA)**, comprising a **Temporal Grounding Adapter (TGA)** and low-rank attention adapters, restores true temporal ordering within a trainable LVM. The two branch outputs are combined by an **Adaptive Fusion** module, then reconstructed and denormalized to produce the final forecast.

3.2 Spectral Magnitude Aligner

Spectral Magnitude Aligner (SMA) decomposes the input image into magnitude and phase components. The magnitude is selectively enhanced by a learnable frequency enhancer to shift the spectral statistics of the rendered time series toward those of natural images, while the phase is kept unchanged to preserve temporal structure, following recent evidence that amplitude components yield more stable optimization dynamics Jiang et al. (2026). The enhanced spectrum is recombined and passed through inverse FFT before being forwarded to the frozen MAE encoder.

Concretely, given an input 2D time series image $\mathbf{I} \in \mathbb{R}^{B\times 1\times H\times W}$, where B is the batch size and H, W are spatial dimensions, SMA first computes the 2D Real Fast Fourier Transform (RFFT):

$$\mathbf{F} = \mathcal{F}_{\text{RFFT}}(\mathbf{I}), \tag{1}$$

where $\mathbf{F} \in \mathbb{C}^{B\times 1\times H\times (W/2+1)}$ contains the complex frequency coefficients. The transform decomposes $\mathbf{F}$ into magnitude and phase components:

$$\mathbf{A} = |\mathbf{F}|,\ \varphi = \angle\mathbf{F}, \tag{2}$$

with $\mathbf{A}, \varphi \in \mathbb{R}^{B\times 1\times H\times (W/2+1)}$ representing the magnitude and phase spectra, respectively. The key innovation lies in selectively enhancing only the magnitude spectrum while preserving the phase information. A learnable convolutional network $\mathcal{E}_\theta$ processes the magnitude spectrum:

$$\mathbf{A}' = \mathcal{E}_\theta(\mathbf{A}), \tag{3}$$

where $\mathcal{E}_\theta$ consists of two convolutional layers:

$$\mathcal{E}_\theta = \text{Conv}_{3\times 3}(1, 16) \to \text{BN} \to \text{ReLU} \to \text{Dropout}(0.1) \to \text{Conv}_{3\times 3}(16, 1). \tag{4}$$

The enhanced frequency representation is reconstructed by combining the processed magnitude with the original phase:

$$\mathbf{F}' = \mathbf{A}' \odot e^{j\varphi}, \tag{5}$$

where $\odot$ denotes element-wise multiplication. The enhanced spatial image is obtained via inverse RFFT:

$$\mathbf{I}_{\text{enhanced}} = \mathcal{F}_{\text{IRFFT}}(\mathbf{F}'). \tag{6}$$

To prevent over-enhancement and maintain training stability, a residual blending is applied:

$$\mathbf{I}_{\text{sp}} = \mathbf{I} + \lambda(\mathbf{I}_{\text{enhanced}} - \mathbf{I}), \tag{7}$$

where λ is a blending coefficient. We set $\lambda = 0.05$, validated through ablation as the optimal trade-off: larger values risk over-enhancement and information loss, while smaller values yield negligible spectral correction.

The enhanced single-channel image is replicated across three channels to satisfy MAE's RGB input format:

$$\mathbf{I}_{\text{sp}}^{\text{3c}} = \text{repeat}(\mathbf{I}_{\text{sp}}, \text{channels} = 3). \tag{8}$$

The resulting $\mathbf{I}_{\text{sp}}^{\text{3c}} \in \mathbb{R}^{B\times 3\times H\times W}$ is forwarded to the frozen MAE encoder to produce y_{sp}.

3.3 Structural-Guided Low-Rank Adaptation

To close the structural gap between rendered time series images and pre-trained vision models, we propose **Structural-Guided Low-Rank Adaptation (SG-LoRA)**, a parameter-efficient fine-tuning strategy that explicitly injects temporal ordering into patch-level representations while preserving the spatial reasoning capacity of the pre-trained backbone.

3.3.1 Temporal Grounding Adapter

The core innovation of the structural branch is the **Temporal Grounding Adapter (TGA)**, which re-establishes the correspondence between spatial patch positions in the rendered image and their original temporal positions in the 1D series, thereby resolving the structural misalignment introduced by image reshaping.

Given the original rendered image $\mathbf{I} \in \mathbb{R}^{B\times 1\times H\times W}$, we first divide it into non-overlapping patches following the standard ViT preprocessing. Each patch is then linearly projected through a learnable embedding layer to obtain the patch embeddings $\mathbf{X} \in \mathbb{R}^{B\times L\times D}$, where $L = \frac{H}{P} \times \frac{W}{P}$ denotes the number of image patches with patch size P. Based on these patch embeddings, the TGA module then generates the original temporal indices:

$$\mathbf{T} = [0, 1, 2, \ldots, L-1] \in \mathbb{R}^{1\times L}, \tag{9}$$

where each index corresponds to the original temporal position in the time series. Subsequently, temporal positional embeddings are generated via 1D sinusoidal position encoding:

$$\mathbf{P}_{\text{temp}}[i, 2k] = \sin\left(\frac{i}{\omega_k}\right), \quad \mathbf{P}_{\text{temp}}[i, 2k+1] = \cos\left(\frac{i}{\omega_k}\right), \tag{10}$$

where $\omega_k = 10000^{2k/D}$ is the frequency base and $k \in \{0, \ldots, D/2-1\}$. This sinusoidal encoding captures multi-frequency temporal patterns, consistent with standard position encoding in Transformers. To enhance representational capacity and task adaptability, TGA introduces a learnable projection layer:

$$\mathbf{P}'_{\text{temp}} = \mathbf{W}_{\text{proj}}\mathbf{P}_{\text{temp}}, \tag{11}$$

where $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{D\times D}$ is a learnable linear projection matrix. Finally, temporal positional information is injected into the original patch embeddings via a learnable fusion gating mechanism:

$$g = \sigma(w_{\text{fusion}}), \quad \mathbf{X}_{\text{enhanced}} = \mathbf{X} + g \cdot \mathbf{P}'_{\text{temp}}, \tag{12}$$

where w_{fusion} is a learnable parameter, $\sigma(\cdot)$ is the sigmoid function, and $g \in [0, 1]$ adaptively controls the fusion strength. This design enables the model to dynamically balance original visual features with temporal structural information based on task requirements.

3.3.2 Low-Rank Adaptation

TGA is combined with Low-Rank Adaptation (LoRA) to form the complete SG-LoRA module. LoRA achieves parameter-efficient fine-tuning by imposing a low-rank decomposition on the query, key, and value projection matrices of the attention layers:

$$\mathbf{W}' = \mathbf{W} + \frac{\alpha_{\text{lora}}}{r}\mathbf{BA}, \tag{13}$$

where $\mathbf{W} \in \mathbb{R}^{D\times D}$ is the original weight matrix, $\mathbf{B} \in \mathbb{R}^{D\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times D}$ are the low-rank decomposition matrices, r is the rank, and α_{lora} is the scaling factor.

The output of the MAE after SG-LoRA fine-tuning is the prediction y_{st}.

3.4 Adaptive Fusion

The two branches address complementary aspects of the modality gap, but their relative importance varies across datasets and forecasting horizons. To adaptively balance their contributions, we introduce a learnable scalar $\beta \in \mathbb{R}$, initialized to 0.5 and constrained to a valid mixing range:

$$\beta = \text{clamp}\,(\beta,\ 0,\ 1)\,. \tag{14}$$

The final prediction is a convex combination of both branch outputs:

$$\hat{y} = \beta \cdot y_{\text{st}} + (1-\beta) \cdot y_{\text{sp}}. \tag{15}$$

where $\hat{y}$ is the final output of the SSDA. A larger β emphasizes temporal dependencies recovered by SG-LoRA, while a smaller β relies more on the visual priors of the frozen LVM. We empirically observe that β remains within $[0, 1]$ throughout training and find this simple parameterization to be as effective as a sigmoid-based gate while avoiding saturation in extreme cases.

4 Experiments

Datasets. We use seven public benchmarks commonly used in time-series forecasting: ETT (Electricity Transformer Temperature) Zhou et al. (2021), including ETTh1, ETTh2, ETTm1 and ETTm2; Weather Wu et al. (2021); Traffic Wu et al. (2021); and Electricity Trindade (2015). For the full-shot experiments, we use chronological splits of 60%/20%/20% (train/val/test) for the ETT datasets, and chronological splits of 70%/10%/20% for the remaining datasets. For the few-shot experiments, we use only 5% or 10% of the original training data as the training set, while the validation and test sets remain unchanged.

4.1 Full-shot Forecasting

We evaluate SSDA against a diverse set of baselines on seven real-world benchmarks under four forecasting horizons, with results summarized in Table 1. The following observations highlight the effectiveness of SSDA: **(1) Consistent state-of-the-art performance.** SSDA achieves **48 first-place results** out of all metric-horizon combinations. This demonstrates that SSDA delivers not only strong

average performance but also consistent superiority across diverse datasets and prediction horizons. **(2) Significant gains over the strongest LVM-based baseline.** Compared with DMMV, the second-best model, SSDA reduces MSE and MAE significantly. These improvements are consistent across all datasets, indicating that explicitly addressing the spectral and structural modality gaps leads to substantial and reliable performance gains beyond what ad hoc adaptation strategies can achieve. **(3) Clear advantage over LLM-based methods.** SSDA outperforms LLM-based forecasters (Time-LLM, GPT4TS and TimeCMA) by a wider margin, with the performance gap widening further on long-horizon settings. This suggests that visual representations are more transferable to time series than language representations, consistent with our spectral analysis showing that the modality gap between rendered time series images and natural images is substantially smaller than that between time series and text. **(4) Superiority of LVM-based approaches over classical methods.** Across all datasets, the top-performing models are uniformly vision-based (SSDA, DMMV, VisionTS), while traditional deep forecasters (PatchTST, TimesNet, DLinear) lag considerably behind, especially on datasets with strong periodic structure. This observation underscores the value of leveraging the rich visual priors accumulated during large-scale pre-training on natural images.

Table 1: Full TSF results with forecasting horizons $H \in \{96, 192, 336, 720\}$. Red indicates the best performance, and Blue indicates the second best. "OOM" denotes the out-of-memory failure.

Models		SSDA		DMMV		Time-VLM		VisionTS		Time-LLM		TimeCMA		GPT4TS		PatchTST		TimesNet		FEDformer		DLinear	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.343	0.372	0.354	0.389	0.361	0.386	0.355	0.386	0.362	0.392	0.373	0.391	0.376	0.397	0.370	0.399	0.384	0.402	0.376	0.419	0.375	0.399
	192	0.387	0.402	0.393	0.405	0.397	0.415	0.395	0.407	0.398	0.418	0.427	0.421	0.416	0.418	0.413	0.421	0.436	0.429	0.420	0.448	0.405	0.416
	336	0.409	0.430	0.387	0.413	0.420	0.421	0.419	0.421	0.430	0.427	0.458	0.448	0.442	0.433	0.422	0.436	0.491	0.469	0.459	0.465	0.439	0.443
	720	0.408	0.430	0.445	0.450	0.441	0.458	0.458	0.460	0.442	0.457	0.449	0.460	0.477	0.456	0.447	0.466	0.521	0.500	0.506	0.507	0.472	0.490
	Avg	0.387	0.406	0.395	0.414	0.405	0.420	0.407	0.419	0.408	0.423	0.423	0.431	0.465	0.455	0.413	0.430	0.458	0.450	0.440	0.460	0.422	0.437
ETTh2	96	0.267	0.328	0.294	0.349	0.267	0.335	0.288	0.334	0.286	0.346	0.286	0.336	0.285	0.342	0.274	0.336	0.340	0.374	0.358	0.397	0.289	0.353
	192	0.324	0.368	0.339	0.395	0.326	0.373	0.349	0.380	0.329	0.375	0.363	0.387	0.354	0.389	0.339	0.379	0.402	0.414	0.429	0.439	0.383	0.418
	336	0.347	0.392	0.322	0.384	0.357	0.406	0.364	0.398	0.368	0.409	0.406	0.421	0.373	0.407	0.329	0.380	0.452	0.452	0.496	0.487	0.448	0.465
	720	0.370	0.415	0.392	0.425	0.412	0.449	0.403	0.431	0.405	0.434	0.417	0.438	0.406	0.441	0.379	0.422	0.462	0.468	0.463	0.474	0.605	0.551
	Avg	0.327	0.376	0.337	0.388	0.341	0.391	0.351	0.386	0.334	0.383	0.372	0.397	0.381	0.412	0.330	0.379	0.414	0.427	0.437	0.449	0.431	0.446
ETTm1	96	0.283	0.328	0.279	0.329	0.304	0.346	0.284	0.332	0.291	0.341	0.312	0.351	0.292	0.346	0.290	0.342	0.338	0.375	0.379	0.419	0.299	0.343
	192	0.316	0.353	0.317	0.357	0.332	0.366	0.327	0.362	0.341	0.369	0.361	0.378	0.332	0.372	0.332	0.369	0.374	0.387	0.426	0.441	0.335	0.365
	336	0.340	0.370	0.351	0.381	0.364	0.383	0.354	0.382	0.359	0.379	0.392	0.401	0.366	0.394	0.366	0.392	0.410	0.411	0.445	0.459	0.369	0.386
	720	0.388	0.400	0.411	0.415	0.364	0.410	0.411	0.415	0.433	0.419	0.453	0.438	0.417	0.421	0.416	0.420	0.478	0.450	0.543	0.490	0.425	0.421
	Avg	0.332	0.363	0.340	0.371	0.351	0.376	0.344	0.373	0.356	0.377	0.380	0.392	0.388	0.403	0.351	0.380	0.400	0.406	0.448	0.452	0.357	0.378
ETTm2	96	0.162	0.249	0.172	0.260	0.160	0.252	0.174	0.262	0.161	0.253	0.173	0.258	0.173	0.262	0.165	0.255	0.187	0.267	0.203	0.287	0.167	0.269
	192	0.217	0.290	0.227	0.298	0.220	0.291	0.228	0.297	0.219	0.293	0.238	0.301	0.229	0.301	0.220	0.292	0.249	0.309	0.269	0.328	0.224	0.303
	336	0.268	0.323	0.272	0.327	0.270	0.325	0.281	0.337	0.271	0.329	0.297	0.338	0.286	0.341	0.274	0.329	0.321	0.351	0.325	0.366	0.281	0.342
	720	0.341	0.375	0.351	0.381	0.348	0.378	0.384	0.410	0.352	0.379	0.393	0.394	0.378	0.401	0.362	0.385	0.408	0.403	0.421	0.415	0.397	0.421
	Avg	0.249	0.311	0.256	0.317	0.250	0.312	0.267	0.327	0.251	0.313	0.275	0.323	0.284	0.339	0.255	0.315	0.291	0.333	0.305	0.349	0.267	0.333
Weather	96	0.139	0.193	0.143	0.195	0.148	0.200	0.146	0.194	0.147	0.201	0.167	0.211	0.162	0.212	0.149	0.198	0.172	0.220	0.217	0.296	0.176	0.237
	192	0.185	0.236	0.187	0.242	0.193	0.240	0.194	0.238	0.189	0.234	0.212	0.253	0.204	0.248	0.194	0.241	0.219	0.261	0.276	0.336	0.220	0.282
	336	0.230	0.272	0.237	0.273	0.243	0.281	0.243	0.275	0.262	0.279	0.270	0.292	0.254	0.286	0.245	0.282	0.280	0.306	0.339	0.380	0.265	0.319
	720	0.314	0.329	0.302	0.315	0.312	0.332	0.318	0.328	0.304	0.316	0.350	0.348	0.326	0.337	0.314	0.334	0.365	0.359	0.403	0.428	0.333	0.362
	Avg	0.217	0.258	0.217	0.256	0.224	0.263	0.225	0.258	0.225	0.257	0.250	0.276	0.237	0.270	0.225	0.264	0.259	0.287	0.309	0.360	0.248	0.300
Electricity	96	0.126	0.219	0.126	0.213	0.142	0.245	0.127	0.217	0.131	0.224	0.143	0.238	0.139	0.238	0.129	0.222	0.168	0.272	0.193	0.308	0.140	0.237
	192	0.145	0.238	0.145	0.237	0.157	0.260	0.148	0.237	0.152	0.241	0.161	0.259	0.153	0.251	0.157	0.240	0.184	0.289	0.201	0.315	0.153	0.249
	336	0.160	0.255	0.162	0.254	0.174	0.276	0.163	0.253	0.160	0.248	0.169	0.261	0.169	0.266	0.163	0.259	0.198	0.300	0.214	0.329	0.169	0.267
	720	0.194	0.288	0.197	0.286	0.214	0.308	0.199	0.293	0.192	0.298	0.219	0.315	0.206	0.297	0.197	0.290	0.220	0.320	0.246	0.355	0.203	0.301
	Avg	0.156	0.250	0.158	0.248	0.172	0.272	0.159	0.250	0.158	0.252	0.174	0.269	0.167	0.263	0.161	0.252	0.192	0.295	0.214	0.327	0.166	0.263
Traffic	96	0.345	0.243	0.344	0.237	0.393	0.290	0.346	0.234	0.362	0.248	OOM	OOM	0.388	0.282	0.360	0.249	0.593	0.321	0.587	0.366	0.410	0.282
	192	0.359	0.247	0.363	0.249	0.405	0.296	0.376	0.245	0.374	0.247	OOM	OOM	0.407	0.290	0.379	0.256	0.617	0.336	0.604	0.373	0.423	0.287
	336	0.395	0.267	0.387	0.256	0.420	0.305	0.389	0.252	0.385	0.271	OOM	OOM	0.412	0.294	0.392	0.264	0.629	0.336	0.621	0.383	0.436	0.296
	720	0.427	0.282	0.433	0.284	0.459	0.323	0.432	0.293	0.430	0.288	OOM	OOM	0.450	0.312	0.432	0.286	0.640	0.350	0.626	0.382	0.466	0.315
	Avg	0.381	0.261	0.382	0.257	0.419	0.304	0.386	0.256	0.388	0.264	OOM	OOM	0.414	0.294	0.390	0.263	0.620	0.336	0.610	0.376	0.433	0.295
1st count		**48**		16		3		6		1		0		0		1		0		0		0	

4.2 Few-shot Forecasting

Table 2 reports few-shot results using 10% and 5% training data. SSDA achieves **9** and **8** first-place results under the two ratios, respectively, consistently outperforming competing methods across the majority of datasets. In contrast, traditional deep models degrade sharply under data scarcity, while pre-trained model-based methods remain relatively stable, underscoring the practical value of transferable LVM representations in low-data regimes. Note that DMMV is excluded from the few-shot comparison as its official implementation does not support the few-shot training protocol used in this evaluation. Full per-horizon results are provided in Appendix D.1.

4.3 Ablation Study

Table 3 reports ablation results on four ETT datasets, and full results are in Appendix D.2. We study three ablation variants: (1) **W/O Spectral branch**: removes SMA, retaining only SG-LoRA; (2) **W/O Structural branch**: removes SG-LoRA, retaining only SMA; (3) **W/O Spectral branch & TGA**: further removes TGA from the structural branch, reducing it to vanilla LoRA fine-tuning.

Table 2: Few-shot forecasting results on 10% and 5% training data.

Ratio	Dataset	SSDA		Time-VLM		VisionTS		Time-LLM		GPT4TS		PatchTST		TimesNet		FEDformer		Informer		DLinear	
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
10%	ETTh1	0.396	0.410	0.431	0.442	0.409	0.412	0.556	0.522	0.590	0.525	0.633	0.542	0.869	0.628	0.639	0.561	1.199	0.809	0.691	0.600
	ETTh2	0.324	0.373	0.361	0.405	0.353	0.389	0.370	0.394	0.397	0.421	0.415	0.431	0.479	0.465	0.466	0.475	3.872	1.513	0.605	0.538
	ETTm1	0.360	0.369	0.360	0.382	0.595	0.508	0.404	0.427	0.464	0.441	0.501	0.466	0.677	0.537	0.722	0.605	1.192	0.821	0.411	0.429
	ETTm2	0.270	0.320	0.263	0.323	0.292	0.342	0.277	0.323	0.293	0.335	0.296	0.343	0.320	0.353	0.463	0.488	3.370	1.440	0.316	0.368
	Weather	0.230	0.271	0.245	0.282	0.247	0.285	0.234	0.273	0.238	0.275	0.242	0.279	0.279	0.301	0.284	0.324	0.597	0.495	0.241	0.283
	Electricity	0.199	0.290	0.198	0.291	0.180	0.265	0.175	0.270	0.176	0.269	0.180	0.273	0.323	0.392	0.346	0.427	1.195	0.891	0.180	0.280
	Traffic	0.468	0.310	0.484	0.357	0.483	0.344	0.429	0.306	0.440	0.310	0.430	0.305	0.951	0.535	0.663	0.425	1.534	0.811	0.447	0.313
1st count		**9**		1		1		3		0		0		0		0		0		0	
5%	ETTh1	0.390	0.401	0.442	0.453	0.409	0.403	0.627	0.543	0.681	0.560	0.694	0.569	0.925	0.647	0.658	0.562	1.225	0.817	0.750	0.611
	ETTh2	0.311	0.358	0.354	0.402	0.348	0.376	0.382	0.418	0.400	0.433	0.827	0.615	0.439	0.448	0.463	0.454	3.922	1.653	0.694	0.577
	ETTm1	0.363	0.369	0.364	0.385	0.574	0.498	0.425	0.434	0.472	0.450	0.526	0.476	0.717	0.561	0.730	0.592	1.163	0.791	0.400	0.417
	ETTm2	0.281	0.326	0.262	0.323	0.308	0.350	0.274	0.323	0.308	0.346	0.314	0.352	0.344	0.372	0.381	0.404	3.658	1.489	0.399	0.426
	Weather	0.244	0.275	0.246	0.284	0.255	0.296	0.260	0.309	0.263	0.301	0.269	0.303	0.298	0.318	0.309	0.353	0.584	0.527	0.263	0.308
	Electricity	0.191	0.276	0.218	0.315	0.183	0.269	0.179	0.268	0.178	0.273	0.181	0.277	0.402	0.453	0.266	0.353	1.281	0.929	0.176	0.275
	Traffic	0.493	0.330	0.558	0.410	0.715	0.469	0.423	0.298	0.434	0.305	0.418	0.296	0.867	0.493	0.676	0.423	1.591	0.832	0.450	0.317
1st count		**8**		2		0		1		0		2		0		0		0		1	

Both branches are necessary. Removing either branch consistently degrades performance across all datasets, confirming that the spectral and structural gaps are complementary and must be addressed jointly. The structural branch contributes more since removing it (Variant 1) causes larger drops than removing the spectral branch (Variant 2), particularly on the minute-level datasets (ETTm1: -9.04%, ETTm2: -15.70% in MSE).

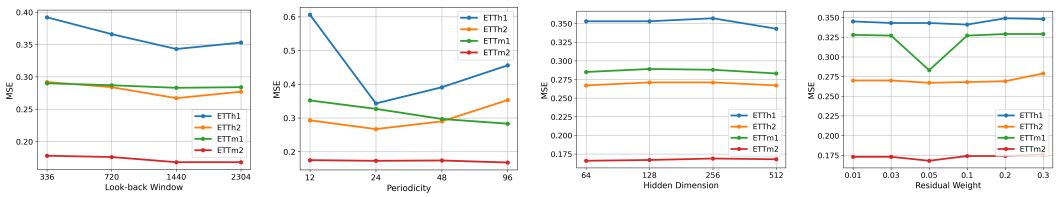
Vanilla LoRA alone is insufficient. Variant 3 further isolates the role of TGA by reducing SG-LoRA to plain LoRA, and causes the largest degradation among all variants (-14.01% MSE / -5.86% MAE on average). This confirms that naive LoRA fine-tuning, without explicit temporal structure injection through TGA, fails to bridge the gap between visual pretraining and time series forecasting, and that TGA is the key driver enabling LoRA to transfer effectively.

Table 3: Ablation study with relative change (%). All degradations are highlighted in red.

Dataset	ETTh1		ETTh2		ETTm1		ETTm2		Avg	
Method(↓), Metric(→)	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
SSDA (full)	**0.387**	**0.406**	**0.327**	**0.376**	**0.332**	**0.363**	**0.249**	**0.311**	**0.324**	**0.364**
(1) W/O Spectral branch	0.403	0.417	0.332	0.378	0.362	0.374	0.288	0.331	0.346	0.375
Drop	−4.13%	−2.71%	−1.53%	−0.53%	−9.04%	−3.03%	−15.70%	−6.43%	−7.60%	−3.18%
(2) W/O Structural branch	0.391	0.408	0.332	0.378	0.342	0.367	0.260	0.318	0.331	0.368
Drop	−1.03%	−0.49%	−1.53%	−0.53%	−3.01%	−1.10%	−4.42%	−2.25%	−2.50%	−1.09%
(3) W/O Spectral branch & TGA	0.462	0.438	0.396	0.410	0.353	0.369	0.272	0.323	0.371	0.385
(Only LoRA) Drop	−19.38%	−7.88%	−21.10%	−9.04%	−6.33%	−1.65%	−9.24%	−4.86%	−14.01%	−5.86%

4.4 Hyper-parameter Analysis

We analyze four key hyper-parameters in Figure 3. **(a) Look-back window:** performance improves up to 1440 and saturates beyond, indicating that longer history helps but redundancy eventually sets in. **(b) Periodicity:** the optimum is 24 on ETTh and 96 on ETTm, matching the $4\times$ sampling-rate gap between the two and confirming that periodicity should align with the daily cycle of the data. **(c) Hidden dimension:** performance stays nearly flat across $d_{\text{model}} \in \{64, \ldots, 512\}$, showing the



(a) Look-back window (b) Periodicity (c) Hidden dimension (d) Residual weight

Figure 3: Sensitivity analysis of key hyper-parameters.

method is not capacity-bound. **(d) Residual weight λ:** the best result is achieved at λ=0.05; smaller values weaken SMA's enhancement, while larger values distort the original signal. Overall, SSDA is robust across a wide range of settings.

4.5 Validation of Gap Mitigation

SPECTRAL GAP MITIGATION. To verify that SMA effectively closes the spectral gap, we conduct a PSS analysis on both the original rendered images and the SMA-enhanced images. As shown in Table 4, the PSS values consistently increase after applying SMA across all datasets, indicating that SMA successfully shifts the frequency distribution of rendered time series images toward natural image statistics, thereby narrowing the spectral gap identified in Section 1.

Table 4: PSS comparison between original images and SMA-enhanced images.

	ETTh1	ETTh2	ETTm1	ETTm2
Original TS images	1.37	1.98	1.68	1.62
SMA enhanced TS images	**1.48**	**2.06**	**1.78**	**1.68**

STRUCTURAL GAP MITIGATION. Figure 4 presents a visualization comparison of the reconstructed images from both branches. The spectral branch produces relatively smooth structures that largely preserve the spurious spatial layout introduced by 2D reshaping, and therefore does not explicitly resolve the structural gap. In contrast, the structural branch yields reconstructed images with pronounced vertical patterns, which are consistent with the one-dimensional temporal ordering of the original series (where each column corresponds to a temporal slice). This observation indicates that SG-LoRA effectively restores temporal coherence that is disrupted during the image transformation process, while suppressing misleading spatial correlations. More examples and corresponding prediction results are provided in Appendix D.4.

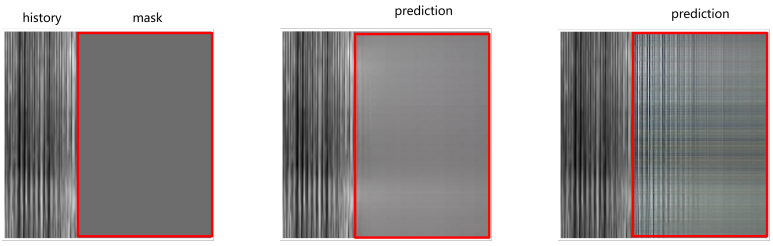


Figure 4: Visualization of structural gap mitigation on an ETTm1 sample. **Left:** original rendered image. **Middle:** spectral branch output, which preserves the spurious 2D spatial layout. **Right:** structural branch output, which exhibits pronounced vertical patterns aligned with the 1D temporal ordering, indicates that SG-LoRA bridges the structural gap.

5 Conclusion

In this paper, we revisit a largely unexamined premise behind LVM-based time series forecasting and identify two distinct facets of the modality gap between rendered and natural images: a *spectral* gap in data statistics and a *structural* gap induced by 2D folding. Guided by this diagnosis, we propose **SSDA**, a dual-branch network that addresses each gap at the level where it originates. The Spectral Magnitude Aligner (SMA) reshapes frequency-domain statistics at the data level via selective magnitude enhancement under phase preservation, while Structural-Guided LoRA (SG-LoRA) restores temporal ordering at the model level through position-aware encodings and low-rank attention updates. Experiments on seven real-world benchmarks show that SSDA achieves state-of-the-art performance in both full-shot and few-shot settings over 12 strongest baselines. We hope our findings could open new avenues for further cross-modality research.

REFERENCES

Zhongde An, Jinhong You, Jiyanglin Li, Yiming Tang, Wen Li, Heming Du, and Shouguo Du. Fredn: Spectral disentanglement for time series forecasting via learnable frequency decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19623–19631, 2026.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. Visionts: Visual masked autoencoders are free-lunch zero-shot time series forecasters. In *International Conference on Machine Learning*, pp. 8979–9007. PMLR, 2025.

Sully F Chen, Zhicheng Guo, Cheng Ding, Xiao Hu, and Cynthia Rudin. Sparse learned kernels for interpretable and efficient medical time series processing. *Nature machine intelligence*, 6(10): 1132–1144, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, pp. 1–21, 2021.

Jingru Fei, Kun Yi, Wei Fan, Qi Zhang, and Zhendong Niu. Amplifier: Bringing attention to neglected low-energy components in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 11645–11653, 2025.

Christoph Feichtenhofer, Haoqi Fan, and Yanghao Li Kaiming He. Masked autoencoders as spatiotemporal learners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 35946–35958, 2022.

D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America. A, Optics and Image Science*, 4(12):2379–2394, 1987. doi: 10.1364/josaa.4.002379.

Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, and Soujanya Poria. Text-to-audio generation using instruction guided latent diffusion model. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 3590–3598, 2023.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. In *International Conference on Machine Learning*, pp. 16115–16152. PMLR, 2024.

Michael Hassid, Tal Remez, Tu Anh Nguyen, Itai Gat, Alexis Conneau, Felix Kreuk, Jade Copet, Alexandre Defossez, Gabriel Synnaeve, Emmanuel Dupoux, et al. Textually pretrained speech language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 63483–63501, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Yue Hu, Jialiang Tang, Siwei Yu, Baosheng Yu, Jing Zhang, and Dacheng Tao. Timeapn: Adaptive amplitude-phase non-stationarity normalization for time series forecasting. *arXiv preprint arXiv:2603.17436*, 2026.

Xudong Jiang, Mingshan Loo, Hanchen Yang, Wengen Li, Mingrui Zhang, Yichao Zhang, Jihong Guan, and Shuigeng Zhou. Adamamba: Adaptive frequency-gated mamba for long-term time series forecasting, 2026. URL https://arxiv.org/abs/2604.23239.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Unb5CVPtae.

Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pp. 5583–5594. PMLR, 2021.

Xixi Li, Yanfei Kang, and Feng Li. Forecasting with time series imaging. *Expert Systems with Applications*, 160:113680, 2020. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2020.113680. URL https://www.sciencedirect.com/science/article/pii/S0957417420305042.

Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 6555–6565, 2024.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18780–18788, 2025.

Ming-Chih Lo, Ching Chang, and Wen-Chih Peng. Text2freq: Learning series patterns from text via frequency domain. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024. URL https://openreview.net/forum?id=Pi6sA1MSSr.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Frozen pretrained transformers as universal computation engines. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7628–7636, 2022.

Huu Hiep Nguyen, Minh Hoang Nguyen, Dung Nguyen, and Hung Le. Spectral text fusion: A frequency-aware approach to multimodal time-series forecasting. *arXiv preprint arXiv:2602.01588*, 2026.

Jingchao Ni, Ziming Zhao, ChengAo Shen, Hanghang Tong, Dongjin Song, Wei Cheng, Dongsheng Luo, and Haifeng Chen. Harnessing vision models for time series analysis: a survey. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 10612–10620, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, pp. 1–10, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Peisong Niu, Tian Zhou, Xue Wang, Liang Sun, and Rong Jin. Understanding the role of textual prompts in llm for time series forecasting: an adapter view. *arXiv preprint arXiv:2311.14782*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Daniel L Ruderman and William Bialek. Statistics of natural images: scaling in the woods. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, pp. 551–558, 1993.

ChengAo Shen, Wenchao Yu, Ziming Zhao, Dongjin Song, Wei Cheng, Haifeng Chen, and Jingchao Ni. Multi-modal view enhanced large vision models for long-term time series forecasting. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, pp. 1–28, 2025a.

Junhong Shen, Liam Li, Lucio M Dery, Corey Staten, Mikhail Khodak, Graham Neubig, and Ameet Talwalkar. Cross-modal fine-tuning: Align then refine. In *International Conference on Machine Learning*, pp. 31030–31056. PMLR, 2023.

Lefei Shen, Mouxiang Chen, Xu Liu, Han Fu, Xiaoxue Ren, Jianling Sun, Zhuo Li, and Chenghao Liu. Visionts++: Cross-modal time series foundation model with continual pre-trained vision backbones. *arXiv preprint arXiv:2508.04379*, 2025b.

K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, pp. 1–14. Computational and Biological Learning Society, 2015.

Chenxi Sun, Hongyan Li, Yaliang Li, and Shenda Hong. TEST: Text prototype aligned embedding to activate LLM's ability for time series. In *The Twelfth International Conference on Learning Representations*, pp. 1–21, 2024.

Artur Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015.

Gaurav Verma, Minje Choi, Kartik Sharma, Jamelle Watson-Daniels, Sejoon Oh, and Srijan Kumar. Cross-modal projection in multimodal llms doesn't really project visual attributes to textual space. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 657–664, 2024.

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 *The Eleventh International Conference on Learning Representations*, pp. 1–10, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Hanchen Yang, Jiannong Cao, Wengen Li, Yu Yang, Xiaoyi Li, Lingbai Kong, Yichao Zhang, Jihong Guan, and Shuigeng Zhou. Towards robust and interpretable spatial-temporal graph modeling for traffic prediction. *ACM Transactions on Knowledge Discovery from Data*, 19(9):1–20, 2025a.

Luoxiao Yang, Yun Wang, Xinqi Fan, Israel Cohen, jingdong chen, and Zijun Zhang. Vitime: Foundation model for time series forecasting powered by vision intelligence. *Transactions on Machine Learning Research*, 2025b. ISSN 2835-8856. URL https://openreview.net/forum?id=XInsJDBIkp.

Jiacheng You, Jingcheng Yang, Yuhang Xie, Zhongxuan Wu, Xiucheng Li, Feng Li, Pengjie Wang, Jian Xu, Bo Zheng, and Xinyang Chen. Loft-llm: Low-frequency time-series forecasting with large language models. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1809–1820, 2026.

Xinli Yu, Zheng Chen, and Yanbin Lu. Harnessing llms for temporal data-a study on explainable financial time series forecasting. In *Proceedings of the 2023 conference on empirical methods in natural language processing: industry track*, pp. 739–753, 2023.

Zhenxiang Zang, Yang Qiao, Shaozhen Yan, and Jie Lu. Reliability and validity of power spectrum slope (pss): A metric for measuring resting-state functional magnetic resonance imaging activity of single voxels. *Frontiers in Neuroscience*, 16:871609, 2022. ISSN 1662-4548, 1662-453X. doi: 10.3389/fnins.2022.871609. URL https://www.frontiersin.org/articles/10.3389/fnins.2022.871609/full.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):11121–11128, Jun. 2023a. doi: 10.1609/aaai.v37i9.26317. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Zhen Zeng, Rachneet Kaur, Suchetha Siddagangappa, Tucker Balch, and Manuela Veloso. From pixels to predictions: Spectrogram and vision transformer for better time series forecasting. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 82–90, 2023b.

Mingrui Zhang, Hanchen Yang, Wengen Li, Xudong Jiang, Jihong Guan, Yichao Zhang, and Shuigeng Zhou. Piformer: Towards subseasonal sst prediction with spatial-patched inverted transformer. *Expert Systems with Applications*, pp. 131618, 2026.

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 *International Conference on Machine Learning*, pp. 78478–78497. PMLR, 2025.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*, pp. 27268–27286. PMLR, 2022.

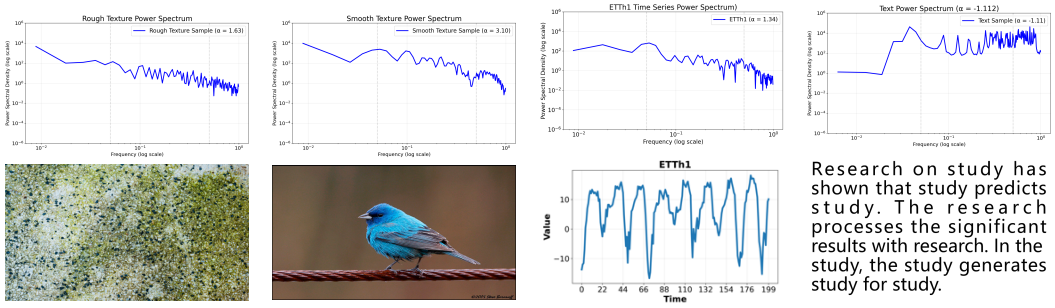
Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. One fits all: Power general time series analysis by pretrained lm. In *Advances in neural information processing systems*, volume 36, pp. 43322–43355, 2023.

A MODALITY GAP ANALYSIS

A.1 EXPERIMENTAL PRINCIPLE

FREQUENCY DISTRIBUTION AND ITS PHYSICAL SIGNIFICANCE Any signal in the spatial (or temporal) domain can be decomposed into a sum of sinusoidal components of different frequencies through Fourier transform. The frequency spectrum reveals how signal energy is distributed across different frequency components. Low frequencies correspond to slow spatial variations (e.g., overall brightness, large-scale structures), while high frequencies capture rapid changes (e.g., edges, textures, fine details).

Taking natural images as an example, as shown in Figure 5(a)-(b), the texture details in images (such as the rough surface of sand or the smooth surface of leaves) manifest as different frequency characteristics. Rough textures contain more high-frequency components, while smooth regions are dominated by low frequencies. This principle extends naturally to time series data (Figure 5(c)), where seasonal patterns and periodic trends correspond to low-frequency oscillations, while noise and sudden fluctuations occupy the high-frequency band. Interestingly, even text (Figure 5(d)), when encoded as character sequences and transformed into spatial representations, exhibits structured frequency patterns that reflect the statistical regularities of language.



(a) Fine gravel and sand($\alpha = 1.66$)

(b) Bird with blurred background($\alpha = 3.10$)

(c) Time series : ETTh1($\alpha = 1.34$)

(d) Text : Wikipedia corpus($\alpha = -1.11$)

Figure 5: Power spectrum analysis of different modalities. Each subfigure shows the power spectrum (top) and the original modality example (bottom). (a) Rough-texture image (fine gravel and sand): $\alpha = 1.66$, indicating high-frequency content from fine textures; (b) Smooth-texture image (bird with blurred background): $\alpha = 3.10$, with gentler decay reflecting large-scale homogeneity; (c) Time series (ETTh1): $\alpha = 1.34$; (d) Text (Wikipedia corpus): $\alpha = -1.11$.

POWER SPECTRUM SLOPE To quantitatively characterize these frequency-domain properties, we employ **Power Spectrum Slope (PSS)** analysis. PSS—defined as the slope of the power spectrum in log–log space—reveals distinct scaling behaviors across modalities. It quantifies the decay rate α in the characteristic power-law scaling $P(f) \propto f^{-\alpha}$, serving as a powerful indicator of the signal's statistical complexity and self-similarity.

In natural images, this manifests as the well-established $1/f^{\alpha}$ scaling (typically $\alpha \approx 2$) Field (1987); Ruderman & Bialek (1993), where deviations encode texture properties—steeper slopes correspond to smoother textures with fewer fine details, and shallower slopes indicate rougher, more granular textures. This property arises from the statistical structure of natural scenes and has been widely used in image processing, computer vision, and neuroscience Hu et al. (2026); Zang et al. (2022). As shown in Figure 5, our measurements ($\alpha = 1.66$ for rough-texture, $\alpha = 3.10$ for smooth-texture, $\alpha = 1.34$ for ETTh1 time series, and $\alpha = -1.11$ for Wikipedia text) reveal substantial variations in frequency distributions across modalities. The PSS thus serves as an effective metric for quantifying these modal disparities. Consequently, by systematically comparing the power-law exponents across these modalities, we can quantitatively measure the modality gap and provide empirical evidence for addressing such distributional shifts in multi-modal time series modeling frameworks.

A.2 DATASETS

TIME SERIES DATA We employ the Electricity Transformer Temperature (ETT) benchmark Zhou et al. (2021) with all four subsets (ETTh1, ETTh2, ETTm1, ETTm2). For each subset, we randomly sample 100 sequences, each with length $L = 1440$, from the full dataset. Each sequence is rendered into a 224×224 grayscale image via SSDA's rendering pipeline.

NATURAL IMAGES We randomly select 100 images from the ImageNet-100 dataset Deng et al. (2009), a curated subset of ImageNet containing 100 object categories. All images are preprocessed to a uniform resolution of 224×224 pixels to ensure consistency with the ViT architecture's input requirements. Each image is normalized to zero mean and unit standard deviation.

TEXT DATA We extract 100 text samples from the English Wikipedia corpus (snapshot dated 20220301.en). Each text sample is converted into a numerical representation by mapping each character to its ASCII code value (ranging from 32 to 126 for printable characters), normalized to the range $[0, 1]$. The resulting 1D sequence is then tiled or truncated to match the target size of $224 \times 224 = 50,176$ elements, and reshaped into a 2D matrix.

A.3 CALCULATION PROCESS

The power spectrum analysis pipeline consists of four sequential stages, each detailed below:

(1) DATA PREPROCESSING

- **Time series**: Loaded from ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2), standardized using training set statistics, then rendered into 224×224 images using the SSDA rendering method (periodic folding with bilinear interpolation).
- **Natural images**: Loaded from standard natural image datasets, resized to 224×224, normalized to zero mean and unit variance.
- **Text**: Converted to numerical sequence using ASCII encoding, reshaped into 2D image format, then processed identically to images.

(2) 2D DISCRETE FOURIER TRANSFORM For each preprocessed image $\mathbf{I}(x, y)$ of size $H \times W$, where $x \in [0, H-1]$ and $y \in [0, W-1]$ denote the spatial coordinates (row and column indices) respectively, we compute the 2D discrete Fourier transform:

$$\mathcal{F}\{\mathbf{I}(x,y)\} = \hat{\mathbf{I}}(u,v) = \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} \mathbf{I}(x,y) e^{-j2\pi\left(\frac{ux}{H} + \frac{vy}{W}\right)} \tag{16}$$

where $u \in [0, H-1]$ and $v \in [0, W-1]$ are the frequency coordinates corresponding to the vertical and horizontal spatial frequencies respectively, $j = \sqrt{-1}$ is the imaginary unit, and the exponential term represents the complex sinusoid basis function at frequency (u, v).

The power spectrum is then computed as the squared magnitude of the Fourier coefficients:

$$P(u,v) = |\hat{\mathbf{I}}(u,v)|^2 = \hat{\mathbf{I}}(u,v) \cdot \hat{\mathbf{I}}^*(u,v) \tag{17}$$

where $|\cdot|$ denotes the complex magnitude and * denotes complex conjugation. For efficiency, we apply the shift operation $\mathcal{F}_{shift}$ to center the zero frequency at the middle of the spectrum before computing the power.

(3) RADIAL AVERAGING To reduce the 2D frequency plane to a 1D power spectrum, we perform radial averaging. Given the shifted power spectrum $P_{shift}(u, v)$, we compute the radial distance for each frequency component:

$$r = \sqrt{(u - H/2)^2 + (v - W/2)^2} \tag{18}$$

where $(H/2, W/2)$ is the center coordinates. The radial power spectrum is then computed as:

$$P_{radial}(r_k) = \frac{1}{N_k} \sum_{(u,v):r\in \text{bin}(r_k)} P_{shift}(u,v) \tag{19}$$

where r_k denotes the k-th radial bin, N_k is the count of frequency components falling into bin k, and $\text{bin}(r_k)$ assigns each (u, v) pair to its corresponding radial bin based on $\lfloor r \rfloor$. The resulting 1D spectrum $P_{radial} \in \mathbb{R}^{R_+}$ represents the average power at each radial frequency $f_k = k/R_{max}$, where $R_{max} = \sqrt{(H/2)^2 + (W/2)^2}$ is the maximum radial frequency.

(4) POWER LAW FITTING The radial power spectrum is fitted to the power law model in log-log space:

$$P(f) \propto f^{-\alpha} \quad \Longleftrightarrow \quad \log P(f) = -\alpha \log f + C \tag{20}$$

where f denotes the normalized spatial frequency (in cycles per pixel, ranging from 0 to 0.5), α is the power law exponent (slope in log-log space) that characterizes the spectral decay rate, and C is a constant offset (intercept) representing the baseline power level.

The fitting is performed using ordinary least squares linear regression on the log-transformed data within the frequency range $f \in [0.05, 0.5]$. This range is selected to exclude: (a) very low frequencies where finite image size effects dominate, and (b) very high frequencies dominated by noise and discretization artifacts. Formally, we minimize:

$$\min_{\alpha,C} \sum_i \left[\log P(f_i) - (-\alpha \log f_i + C)\right]^2 \tag{21}$$

subject to the frequency mask $\mathcal{M} = \{i : 0.05 < f_i < 0.5\}$. The fitting procedure also returns the coefficient of determination R^2 to assess goodness-of-fit.

For statistical robustness, we compute the mean and standard deviation of α across all samples within each modality:

$$\bar{\alpha} = \frac{1}{N} \sum_{j=1}^{N} \alpha_j, \quad \sigma_\alpha = \sqrt{\frac{1}{N-1} \sum_{j=1}^{N} (\alpha_j - \bar{\alpha})^2} \tag{22}$$

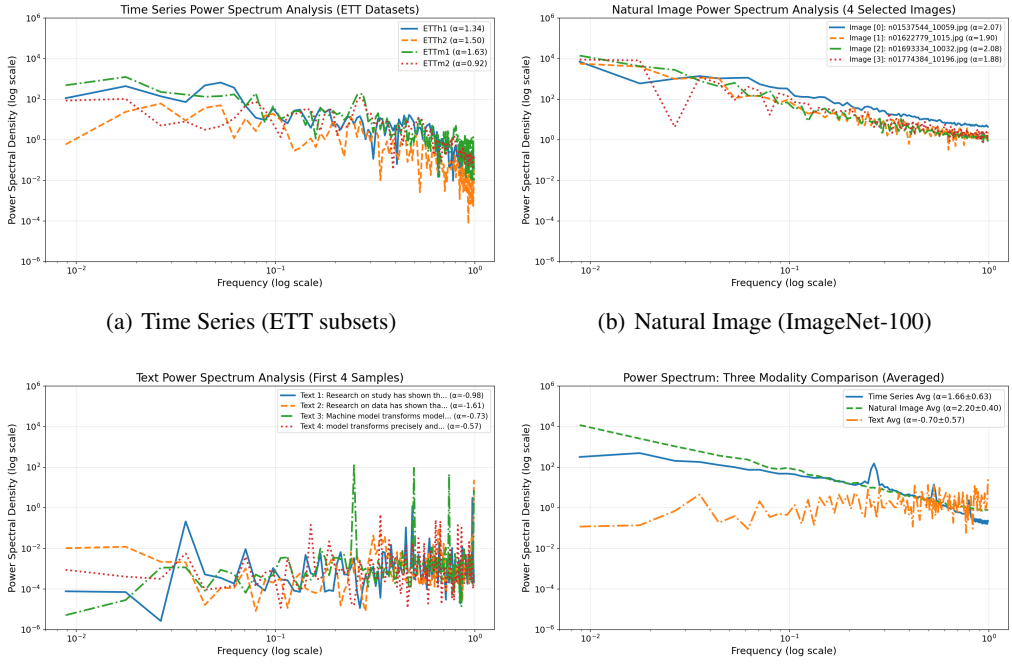
where N is the number of samples per modality, and the subscript j denotes the sample index.

A.4 EXPERIMENTAL RESULTS

To provide intuitive understanding of the modality gap, Figure 6 presents representative power spectral density curves for each modality. Table 5 summarizes the power law fitting results for all three modalities, including data source names, sample counts, and mean exponent values with standard deviations.

Table 5: Power law fitting results for different modalities. α denotes the fitted power law exponent, σ_α is the standard deviation.

Modality	**Source**	**Samples**	$\alpha \pm \sigma_\alpha$
Natural Image	ImageNet-100	100	2.19 ± 0.40
Time Series	ETTh1	100	1.37 ± 0.73
	ETTh2	100	1.98 ± 0.64
	ETTm1	100	1.57 ± 0.53
	ETTm2	100	1.62 ± 0.49
Average	**All subsets**	**400**	1.66 ± 0.63
Text (ASCII)	Wikipedia	100	-0.70 ± 0.57



(a) Time Series (ETT subsets)

(b) Natural Image (ImageNet-100)

(c) Text (Wikipedia)

(d) Three Modality Comparison (Averaged)

Figure 6: Power spectral slope analysis for different modalities. Subfigures (a)-(c) show individual sample curves for time series, natural images, and text respectively. Subfigure (d) displays the averaged power spectral slope.

The results reveal distinct spectral characteristics across modalities. Natural images exhibit the expected $1/f^2$ scaling ($\alpha \approx 2.19$), while time series rendered images show a significantly lower exponent ($\alpha = 1.66$), indicating a modality gap of $\Delta\alpha \approx 0.53$. This discrepancy arises from artificial periodicities introduced by the folding operation. Text modality shows the most distinct behavior with negative exponent ($\alpha = -0.70$), reflecting the high-frequency nature of ASCII-encoded character sequences.

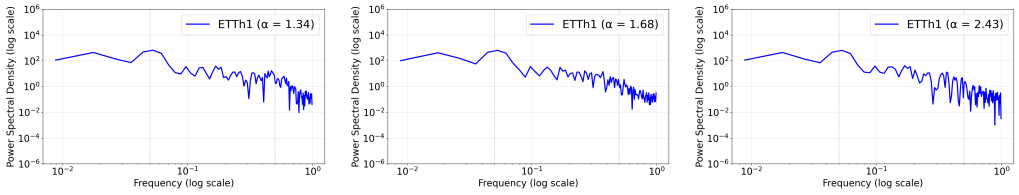
A.5 Discussion

As illustrated in Figure7, during our experiments, we observe that when converting one-dimensional time series into two-dimensional visual representations, the PSS is significantly influenced by the periodicity parameter. Specifically, the PSS α of time series images increases monotonically with periodicity. This is because periodicity determines how temporal information is spatially distributed in the image domain. Larger periodicity values compress more time steps into each row, this spatial compression amplifies high-frequency components in the visual representation, resulting in steeper spectral decay (larger α). Conversely, smaller periodicity preserves finer temporal structures, yielding gentler spectral slopes (smaller α).

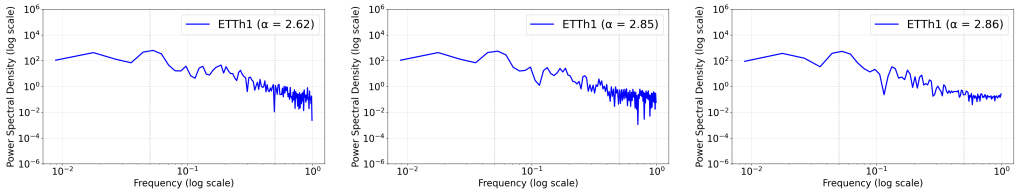
Table6 demonstrates that PSS does not necessarily correlate with prediction performance, thus we do not recommend using PSS as an evaluation metric for forecasting models. In this paper, PSS serves as an analytical tool to characterize the frequency distributions of time series images and natural

Table 6: Ablation study on periodicity.

Periodicity	24	28	30	32	48	64
α	1.34	1.68	2.43	2.62	2.85	2.86
MSE	0.343	0.455	0.477	0.451	0.391	0.444



(a) Periodicity=24 ($\alpha = 1.34$) (b) Periodicity=28 ($\alpha = 1.68$) (c) Periodicity=30 ($\alpha = 2.43$)



(d) Periodicity=32 ($\alpha = 2.62$) (e) Periodicity=48 ($\alpha = 2.85$) (f) Periodicity=64 ($\alpha = 2.86$)

Figure 7: Power spectral slope analysis for time series data rendered with different periodicity values. Subfigures (a)-(f) show the power spectral density (PSD) of ETTh1 data when rendered to image representation with periodicity values of 24, 28, 30, 32, 48, and 64 respectively.

images, enabling frequency-domain analysis and alignment of the two modalities. Our objective is to enhance the visual representations of time series data by exploiting the underlying patterns that emerge across different modalities Shen et al. (2025a). Subsequently, we feed the time series images with stronger visual features into a frozen MAE pretrained on natural images, fully leveraging the knowledge learned from natural image representations, which provides more robust visual-view characteristics and patterns.

In summary, PSS provides a means to quantify the frequency-domain discrepancies between modalities, facilitating the distinction and alignment of time series and natural image distributions. However, it is crucial to avoid excessive modification of the original data during frequency redistribution, as this may introduce unnecessary noise or lead to information loss. Rather than viewing PSS as a performance metric, we advocate treating it as a novel analytical perspective for characterizing modality gaps. This approach offers a novel research direction for future work on multimodal learning and modality fusion.

B Implementation Details

B.1 Data Processing Details

To bridge the gap between time series data and vision model inputs, SSDA implements a bidirectional transformation pipeline that converts temporal sequences into 2D image representations and recovers temporal predictions from vision model outputs. This reshape-reconstruct mechanism preserves temporal structure while enabling the use of spatial reasoning capabilities in vision models.

B.1.1 Reshape

The reshape process transforms time series data $\mathbf{X} \in \mathbb{R}^{B \times T \times N}$ into masked image inputs suitable for MAE processing. The transformation involves several coordinated steps that convert temporal sequences into spatial representations while maintaining temporal relationships.

As illustrated in the Algorithm 1, the process begins by rearranging the time series to group variables together, then applies left padding for periodicity alignment. The padded sequences are converted into 2D spatial representations where height corresponds to the number of periods and width corresponds to the periodicity parameter. These representations are resized to match vision model requirements,

Algorithm 1: Time Series to Image Reshaping

Data: Time series data $\mathbf{X} \in \mathbb{R}^{B \times T \times N}$, batch size B, sequence length T, number of variables N, periodicity p, padding length p_l, target image dimensions $H_{\text{target}} \times W_{\text{target}}$.
Result: Masked image input $\mathbf{I}_{\text{input}} \in \mathbb{R}^{(B \cdot N) \times 3 \times H \times (W + W_{\text{mask}})}$.
$\mathbf{X}_{\text{enc}} \leftarrow \text{rearrange}(\mathbf{X}, \text{'b t n} \rightarrow \text{b n t'})$;
$\mathbf{X}_{\text{pad}} \leftarrow \text{pad}(\mathbf{X}_{\text{enc}}, (p_l, 0), \text{mode} = \text{'replicate'})$;
$\mathbf{X}_{2d} \leftarrow \text{rearrange}(\mathbf{X}_{\text{pad}}, \text{'b n (p f)} \rightarrow \text{(b n) 1 f p'})$;
$\mathbf{X}_{\text{resize}} \leftarrow \text{Resize}(\mathbf{X}_{2d}, (H_{\text{target}}, W_{\text{target}}))$;
if *spectral branch* **then**
  $\mathbf{X}_{3c} \leftarrow \text{Enhance}(\mathbf{X}_{\text{resize}})$;
end
else
  $\mathbf{X}_{3c} \leftarrow \text{repeat}(\mathbf{X}_{\text{resize}}, \text{channels} = 3)$;
end
$\mathbf{I}_{\text{input}} \leftarrow \text{concat}(\mathbf{X}_{3c}, \mathbf{M}_{\text{masked}}, \text{dim} = -1)$;

expanded to three channels, and concatenated with a masked region representing the prediction window.

B.1.2 RECONSTRUCT

The reconstruction process reverses the reshaping operation to recover temporal predictions from MAE outputs $\mathbf{Z} \in \mathbb{R}^{(B \cdot N) \times L \times D}$.

As illustrated in the Algorithm 2, starting with the latent representations from the MAE decoder, patch tokens are converted back to spatial images through unpatchify operations. The RGB reconstructions are converted to grayscale, resized to restore temporal dimensions, flattened back to temporal sequences, and finally the prediction window is extracted from the reconstructed output.

Algorithm 2: Image to Time Series Reconstruction

Data: MAE decoder output $\mathbf{Z} \in \mathbb{R}^{(B \cdot N) \times L \times D}$, batch size B, number of variables N, patch sequence length L, embedding dimension D, context length T_{input}, prediction length T_{pred}.
Result: Temporal predictions $\mathbf{Y}_{\text{pred}} \in \mathbb{R}^{B \times T_{\text{pred}} \times N}$.
$\mathbf{I}_{\text{recon}} \leftarrow \text{unpatchify}(\mathbf{Z})$;
$\mathbf{I}_{\text{gray}} \leftarrow \frac{1}{3} \sum c = 1^3 \mathbf{I}_{\text{recon}}[:, :, c, :, :]$;
$\mathbf{I}_{\text{temp}} \leftarrow \text{Resize}(\mathbf{I}_{\text{gray}}, (p, f \cdot p))$;
$\mathbf{Y} \leftarrow \text{rearrange}(\mathbf{I}_{\text{temp}}, \text{'(b n) 1 f p} \rightarrow \text{b (p f) n'})$;
$\mathbf{Y}_{\text{pred}} \leftarrow \mathbf{Y}[:, T_{\text{input}} : T_{\text{input}} + T_{\text{pred}}, :]$;

B.2 TRAINING CONFIGURATION

B.2.1 MODEL ARCHITECTURE DETAILS

We summarize the default model architecture parameters in Table 7. Image representations are set to a size of 224×224, which conforms to the input requirements of MAE. The model backbone follows a standard Transformer-based design, where the hidden dimension d_model is set to 512 and the feed-forward dimension d_ff is 2048 to ensure sufficient model capacity. The multi-head attention module uses 8 attention heads, while the encoder and decoder are composed of 2 and 1 layers, respectively, balancing performance and computational efficiency. A dropout rate of 0.1 is applied to mitigate overfitting during training. In addition, we employ 5 workers for data loading to accelerate the training pipeline.

Table 7: Default Model Architecture Parameters

Parameter	Default Value	Description
image_size	224	Size of reshaped image representation
d_model	512	Dimension of hidden embeddings
d_ff	2048	Dimension of feed-forward network
n_heads	8	Number of attention heads
e_layers	2	Number of encoder layers
d_layers	1	Number of decoder layers
dropout	0.1	Dropout rate
num_workers	5	Number of data loader workers

B.2.2 Training Hyperparameters

We summarize the default training parameters in Table 8. We use the Adam optimizer with a learning rate of 2×10^{-6}, a batch size of 32, and train for 10 epochs, with early stopping ($patience = 3$) and MSE as the loss function. The input sequence length (seq_len) is set according to dataset characteristics to capture long-range dependencies, while the prediction length ($pred_len$) is evaluated over multiple horizons $(96, 192, 336, 720)$. The output dimension (c_out) is dataset-specific. For the visual representation, periodicity controls the reshaping of time series into 2D image representations (e.g., 24 for hourly data and 96 for 15-minute data), and bilinear interpolation is used for resizing. The normalization and alignment constants are both set to 0.4 to stabilize representation transformation. We adopt LoRA for parameter-efficient fine-tuning with rank $r = 4$, scaling factor 16, and dropout 0.1, and introduce a residual weight of 0.05 for frequency enhancement. The vision backbone is initialized from a pre-trained MAE model. Unless otherwise specified, all parameters are kept consistent across datasets for fair comparison.

B.3 Running Environment

All experiments are conducted on a workstation equipped with 8 NVIDIA GeForce RTX 4090 GPUs (24GB memory each), running CUDA 12.1 and cuDNN 8.9. The model is implemented in Python 3.10 using PyTorch 2.2.2, with key dependencies including NumPy 1.23.5, pandas 1.5.3, and torchvision 0.17.2. More training details can be found in Sections B.2.1 and B.2.2.

C Benchmark and Baseline

C.1 Benchmark

Following prior work Chen et al. (2025); Zhong et al. (2025); Shen et al. (2025a), we evaluate our method on seven widely used long-term time series forecasting (LTSF) benchmarks, covering diverse sampling frequencies, variate dimensions, periodic patterns, and real-world scenarios.

The benchmarks include four Electricity Transformer Temperature (ETT) datasets (ETTh1, ETTh2, ETTm1, ETTm2), which record oil temperature from electric transformers with hourly and 15-minute sampling frequencies Zhou et al. (2021). In addition, the Weather dataset contains 10-minute resolution meteorological measurements Wu et al. (2021). The Traffic dataset collects hourly road occupancy rates from freeway sensors Wu et al. (2021), and the Electricity dataset provides hourly electricity consumption data Trindade (2015).

C.2 Baselines

Baselines We compare our method with a diverse set of representative baselines spanning vision-based, large-model-based, and classical time series forecasting approaches, ensuring a comprehensive evaluation.

Table 8: Default Training Parameters

Parameter	Default Value	Description
batch_size	32	Training batch size
learning_rate	2e-6	Initial learning rate
training_epochs	10	Number of training epochs
patience	3	Early stopping patience
loss	MSE	Mean square error loss
seq_len	1440 (ETTh1/ETTh2/ETTm1/ETTm2) 2304 (Electricity/Traffic) 2360 (Weather)	Input sequence length
pred_len	96/192/336/720	Prediction length
c_out	7 (ETTh1/ETTh2/ETTm1/ETTm2) 21 (Weather) 321 (Electricity) 862 (Traffic)	Output dimension (dataset-specific)
periodicity	24 (ETTh1/ETTh2/Electricity/Traffic) 96 (ETTm1/ETTm2) 144 (Weather)	Periodicity for time-series to image conversion
interpolation	bilinear	Image resizing method
norm_const	0.4	Normalization coefficient
align_const	0.4	Alignment scaling factor
LoRA_rank (r)	4	Rank of low-rank adaptation
LoRA_alpha	16	Scaling factor of LoRA
LoRA_dropout	0.1	Dropout in LoRA layers
residual_weight	0.05	Weight of frequency enhancement residual
vision_backbone	mae_base	Vision model architecture

1. **DMMV** Shen et al. (2025a) introduces a dual-modality modeling framework that jointly leverages visual representations and temporal dynamics for time series forecasting, demonstrating the effectiveness of multimodal fusion.

2. **Time-VLM** Zhong et al. (2025) adapts vision-language models to time series tasks by encoding temporal data into visual formats and leveraging pre-trained multimodal representations for forecasting.

3. **VisionTS** Chen et al. (2025) converts time series into images and directly applies large vision models (LVMs), showcasing the potential of transferring visual knowledge to time series forecasting.

4. **Time-LLM** Jin et al. (2024) reformulates time series forecasting as a language modeling problem, enabling large language models (LLMs) to process temporal data through prompt-based learning.

5. **GPT4TS** Zhou et al. (2023) extends the idea of LLM-based forecasting by further adapting GPT-style architectures to capture temporal dependencies via sequence modeling and instruction tuning.

6. **TimeCMA** Liu et al. (2025) proposes a cross-modality alignment mechanism to retrieve disentangled time series embeddings from LLM-based prompts via channel-wise similarity, enhancing robustness in multivariate forecasting.

7. **PatchTST** Nie et al. (2023) adopts a patch-based Transformer architecture, where time series are segmented into patches to improve long-range dependency modeling and computational efficiency.

8. **TimesNet** Wu et al. (2023) proposes a temporal 2D-variation modeling paradigm by transforming time series into multiple periodic representations, effectively capturing complex temporal patterns.

9. **FEDformer** Zhou et al. (2022) introduces frequency-enhanced decomposition with attention mechanisms, combining Fourier analysis with Transformer architectures to better model long-term dependencies.

10. **Informer** Zhou et al. (2021) is a Transformer-based model designed for long sequence forecasting, utilizing ProbSparse self-attention to reduce computational complexity while preserving performance.

11. **DLinear** Zeng et al. (2023a) is a lightweight linear model that decomposes time series into trend and seasonal components, serving as a strong and efficient baseline despite its simplicity.

D Full Experimental Results

D.1 Few-Shot Forecasting

Table 9: Full few-shot learning results on 10% training data with forecasting horizons $H \in \{96, 192, 336, 720\}$. Lower values indicate better performance. Red indicates the best performance, and Blue indicates the second best.

Models		SSDA		Time-VLM		VisionTS		Time-LLM		GPT4TS		PatchTST		TimesNet		FEDformer		Informer		DLinear	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.359	0.376	0.391	0.404	0.365	0.376	0.448	0.460	0.458	0.456	0.516	0.485	0.861	0.628	0.512	0.499	1.179	0.792	0.492	0.495
	192	0.389	0.403	0.420	0.431	0.407	0.404	0.484	0.483	0.570	0.516	0.598	0.524	0.797	0.593	0.624	0.555	1.199	0.806	0.565	0.538
	336	0.417	0.420	0.439	0.448	0.438	0.426	0.589	0.540	0.608	0.535	0.657	0.550	0.941	0.648	0.691	0.574	1.202	0.811	0.721	0.622
	720	0.418	0.442	0.476	0.484	0.426	0.440	0.700	0.604	0.725	0.591	0.762	0.610	0.877	0.641	0.728	0.614	1.217	0.825	0.986	0.743
	Avg	0.396	0.410	0.431	0.442	0.409	0.412	0.556	0.522	0.590	0.525	0.633	0.542	0.869	0.628	0.639	0.561	1.199	0.809	0.691	0.600
ETTh2	96	0.263	0.325	0.284	0.347	0.294	0.340	0.275	0.326	0.331	0.374	0.353	0.389	0.378	0.409	0.382	0.416	3.837	1.508	0.357	0.411
	192	0.319	0.367	0.349	0.398	0.355	0.383	0.374	0.373	0.402	0.411	0.403	0.414	0.490	0.467	0.478	0.474	3.856	1.513	0.569	0.519
	336	0.346	0.386	0.370	0.412	0.374	0.405	0.406	0.429	0.406	0.433	0.426	0.441	0.537	0.494	0.504	0.501	3.952	1.526	0.671	0.572
	720	0.368	0.412	0.441	0.466	0.390	0.427	0.427	0.449	0.449	0.464	0.477	0.480	0.510	0.491	0.499	0.509	3.842	1.503	0.824	0.648
	Avg	0.324	0.373	0.361	0.405	0.353	0.389	0.370	0.394	0.397	0.421	0.415	0.431	0.479	0.465	0.466	0.475	3.872	1.513	0.605	0.538
ETTm1	96	0.323	0.341	0.310	0.354	0.444	0.428	0.346	0.388	0.390	0.404	0.410	0.419	0.583	0.501	0.578	0.518	1.162	0.785	0.352	0.392
	192	0.340	0.355	0.340	0.370	0.502	0.457	0.373	0.416	0.429	0.423	0.437	0.434	0.630	0.528	0.617	0.546	1.172	0.793	0.382	0.412
	336	0.364	0.371	0.369	0.387	0.665	0.547	0.413	0.426	0.469	0.439	0.476	0.454	0.725	0.568	0.998	0.775	1.227	0.908	0.419	0.434
	720	0.414	0.409	0.423	0.417	0.767	0.601	0.485	0.476	0.569	0.498	0.681	0.556	0.769	0.549	0.693	0.579	1.207	0.797	0.490	0.477
	Avg	0.360	0.369	0.360	0.382	0.595	0.508	0.404	0.427	0.464	0.441	0.501	0.466	0.677	0.537	0.722	0.605	1.192	0.821	0.411	0.429
ETTm2	96	0.209	0.276	0.169	0.260	0.192	0.274	0.177	0.261	0.188	0.269	0.191	0.274	0.212	0.285	0.291	0.399	3.203	1.407	0.213	0.303
	192	0.242	0.300	0.222	0.296	0.243	0.308	0.241	0.314	0.251	0.309	0.252	0.317	0.270	0.323	0.307	0.379	3.112	1.387	0.278	0.345
	336	0.283	0.328	0.278	0.335	0.299	0.347	0.274	0.327	0.307	0.346	0.306	0.353	0.323	0.353	0.543	0.559	3.255	1.421	0.338	0.385
	720	0.346	0.375	0.381	0.401	0.433	0.439	0.417	0.390	0.426	0.417	0.433	0.427	0.474	0.449	0.712	0.614	3.909	1.543	0.436	0.440
	Avg	0.270	0.320	0.263	0.323	0.292	0.342	0.277	0.323	0.293	0.335	0.296	0.343	0.320	0.353	0.463	0.488	3.370	1.440	0.316	0.368
Weather	96	0.153	0.208	0.174	0.228	0.167	0.215	0.161	0.210	0.163	0.215	0.165	0.215	0.184	0.230	0.188	0.253	0.374	0.401	0.171	0.224
	192	0.199	0.251	0.217	0.262	0.210	0.262	0.204	0.248	0.210	0.254	0.210	0.257	0.245	0.283	0.250	0.304	0.552	0.478	0.215	0.263
	336	0.262	0.299	0.263	0.296	0.267	0.306	0.261	0.302	0.256	0.292	0.259	0.297	0.305	0.321	0.312	0.346	0.724	0.541	0.258	0.299
	720	0.306	0.327	0.326	0.340	0.344	0.356	0.309	0.332	0.321	0.339	0.332	0.346	0.381	0.371	0.387	0.393	0.739	0.558	0.320	0.346
	Avg	0.230	0.271	0.245	0.282	0.247	0.285	0.234	0.273	0.238	0.275	0.242	0.279	0.279	0.301	0.284	0.324	0.597	0.495	0.241	0.283
Electricity	96	0.137	0.228	0.160	0.269	0.135	0.224	0.139	0.241	0.139	0.237	0.140	0.238	0.299	0.373	0.231	0.323	1.259	0.919	0.150	0.253
	192	0.163	0.252	0.174	0.279	0.154	0.243	0.151	0.248	0.156	0.252	0.160	0.255	0.305	0.379	0.261	0.356	1.160	0.873	0.164	0.264
	336	0.185	0.273	0.190	0.294	0.178	0.267	0.169	0.270	0.175	0.270	0.180	0.276	0.319	0.391	0.360	0.445	1.157	0.872	0.181	0.282
	720	0.241	0.319	0.229	0.323	0.253	0.326	0.240	0.322	0.233	0.317	0.241	0.323	0.369	0.426	0.530	0.585	1.203	0.898	0.223	0.321
	Avg	0.199	0.268	0.198	0.291	0.180	0.265	0.175	0.270	0.176	0.269	0.180	0.273	0.323	0.392	0.346	0.427	1.195	0.891	0.180	0.280
Traffic	96	0.404	0.277	0.465	0.349	0.515	0.367	0.418	0.291	0.414	0.297	0.403	0.289	0.719	0.416	0.639	0.400	1.557	0.821	0.419	0.298
	192	0.469	0.315	0.468	0.350	0.464	0.343	0.414	0.296	0.426	0.301	0.415	0.296	0.748	0.428	0.637	0.416	1.454	0.765	0.434	0.305
	336	0.453	0.301	0.483	0.356	0.451	0.318	0.421	0.311	0.434	0.303	0.426	0.304	0.853	0.471	0.655	0.427	1.521	0.812	0.449	0.313
	720	0.544	0.346	0.520	0.373	0.502	0.347	0.462	0.327	0.487	0.337	0.474	0.331	1.485	0.825	0.722	0.456	1.605	0.846	0.484	0.336
	Avg	0.468	0.310	0.484	0.357	0.483	0.344	0.429	0.306	0.440	0.310	0.430	0.305	0.951	0.535	0.663	0.425	1.534	0.811	0.447	0.313
1st count		**35**		6		6		18		3		2		0		0		0		1	

D.2 Ablation Study

D.3 Efficiency Analysis

We conduct an efficiency analysis to evaluate the trade-off among forecasting performance, model complexity, and memory consumption. Specifically, we compare different models from three perspectives: (1) prediction accuracy measured by MSE, (2) model size quantified by the number of parameters, and (3) runtime memory usage. The results are visualized in a bubble chart, where the x-axis denotes the parameter scale (in log scale), the y-axis represents MSE, and the bubble size encodes memory consumption.

From the results, several important observations can be drawn. First, **SSDA achieves the best predictive performance** with the lowest MSE (0.343), while maintaining a moderate parameter scale (1.5M). Compared to lightweight models such as VisionTS and DMMV-S, SSDA significantly improves accuracy with only a moderate increase in memory usage, demonstrating a favorable efficiency-performance balance.

Table 10: Full few-shot learning results on 5% training data. "-" denotes that 5% of the time series is insufficient to form a valid training set.

Models		SSDA		Time-VLM		VisionTS		Time-LLM		GPT4TS		PatchTST		TimesNet		FEDformer		Informer		DLinear	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.359	0.376	0.417	0.435	0.369	0.377	0.483	0.464	0.543	0.506	0.557	0.519	0.892	0.625	0.593	0.529	1.225	0.812	0.547	0.503
	192	0.396	0.406	0.450	0.458	0.416	0.408	0.629	0.540	0.748	0.580	0.711	0.570	0.940	0.665	0.652	0.563	1.249	0.828	0.720	0.604
	336	0.415	0.421	0.460	0.465	0.442	0.424	0.768	0.626	0.754	0.595	0.816	0.619	0.945	0.653	0.731	0.594	1.202	0.811	0.984	0.727
	720	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Avg	0.390	0.401	0.442	0.453	0.409	0.403	0.627	0.543	0.681	0.560	0.694	0.569	0.925	0.647	0.658	0.562	1.225	0.817	0.750	0.611
ETTh2	96	0.265	0.325	0.302	0.365	0.300	0.337	0.336	0.397	0.376	0.421	0.401	0.421	0.409	0.420	0.390	0.424	3.837	1.508	0.442	0.456
	192	0.319	0.363	0.361	0.406	0.365	0.388	0.406	0.425	0.418	0.441	0.452	0.455	0.483	0.464	0.457	0.465	3.975	1.933	0.617	0.542
	336	0.349	0.385	0.398	0.434	0.379	0.402	0.405	0.432	0.408	0.439	0.464	0.469	0.499	0.479	0.477	0.483	3.956	1.520	1.424	0.849
	720	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Avg	0.311	0.358	0.354	0.402	0.348	0.376	0.382	0.418	0.401	0.433	0.439	0.448	0.463	0.454	0.441	0.457	3.923	1.654	0.694	0.577
ETTm1	96	0.330	0.342	0.314	0.357	0.444	0.433	0.316	0.377	0.386	0.405	0.399	0.414	0.606	0.518	0.628	0.544	1.130	0.775	0.332	0.374
	192	0.347	0.358	0.343	0.373	0.497	0.458	0.450	0.464	0.440	0.438	0.441	0.436	0.681	0.539	0.666	0.566	1.150	0.788	0.358	0.390
	336	0.369	0.373	0.373	0.391	0.653	0.530	0.450	0.424	0.485	0.459	0.499	0.467	0.786	0.597	0.807	0.628	1.198	0.809	0.402	0.416
	720	0.404	0.403	0.425	0.420	0.702	0.572	0.483	0.471	0.577	0.499	0.767	0.587	0.796	0.593	0.822	0.633	1.175	0.794	0.511	0.489
	Avg	0.363	0.369	0.364	0.385	0.574	0.498	0.425	0.434	0.472	0.450	0.526	0.476	0.717	0.561	0.730	0.592	1.163	0.791	0.400	0.417
ETTm2	96	0.207	0.272	0.169	0.260	0.196	0.277	0.174	0.261	0.199	0.280	0.206	0.288	0.220	0.299	0.229	0.320	3.599	1.478	0.236	0.326
	192	0.282	0.327	0.224	0.298	0.255	0.316	0.215	0.287	0.256	0.316	0.264	0.324	0.311	0.361	0.394	0.361	3.578	1.475	0.306	0.373
	336	0.285	0.329	0.282	0.338	0.321	0.364	0.273	0.330	0.318	0.353	0.334	0.367	0.338	0.366	0.378	0.427	3.561	1.473	0.380	0.423
	720	0.350	0.377	0.375	0.397	0.459	0.444	0.433	0.412	0.460	0.436	0.454	0.432	0.509	0.465	0.523	0.510	3.896	1.533	0.674	0.583
	Avg	0.281	0.326	0.262	0.323	0.308	0.350	0.274	0.323	0.308	0.346	0.314	0.352	0.344	0.372	0.381	0.404	3.658	1.489	0.399	0.426
Weather	96	0.175	0.224	0.176	0.231	0.174	0.228	0.172	0.263	0.175	0.230	0.171	0.224	0.207	0.253	0.229	0.309	0.497	0.497	0.184	0.242
	192	0.211	0.254	0.216	0.263	0.225	0.275	0.224	0.271	0.227	0.276	0.230	0.277	0.272	0.307	0.265	0.317	0.620	0.545	0.228	0.283
	336	0.262	0.289	0.264	0.298	0.284	0.323	0.282	0.321	0.286	0.322	0.294	0.326	0.313	0.328	0.353	0.392	0.649	0.547	0.279	0.322
	720	0.326	0.332	0.327	0.342	0.338	0.357	0.366	0.381	0.366	0.379	0.384	0.387	0.400	0.385	0.391	0.394	0.570	0.522	0.364	0.388
	Avg	0.244	0.275	0.246	0.284	0.255	0.296	0.260	0.309	0.263	0.301	0.269	0.303	0.298	0.318	0.309	0.353	0.584	0.527	0.263	0.308
Electricity	96	0.149	0.240	0.185	0.296	0.141	0.230	0.147	0.242	0.143	0.241	0.145	0.244	0.315	0.389	0.235	0.322	1.265	0.919	0.150	0.251
	192	0.166	0.256	0.194	0.302	0.160	0.249	0.158	0.241	0.159	0.255	0.163	0.260	0.318	0.396	0.247	0.341	1.298	0.939	0.163	0.263
	336	0.199	0.283	0.210	0.315	0.189	0.277	0.178	0.277	0.179	0.274	0.183	0.281	0.340	0.415	0.267	0.356	1.302	0.942	0.175	0.278
	720	0.248	0.324	0.251	0.346	0.243	0.320	0.224	0.312	0.233	0.323	0.233	0.323	0.635	0.613	0.318	0.394	1.259	0.919	0.219	0.311
	Avg	0.191	0.276	0.218	0.315	0.183	0.269	0.179	0.268	0.178	0.273	0.181	0.277	0.402	0.453	0.266	0.353	1.281	0.929	0.176	0.275
Traffic	96	0.469	0.315	0.550	0.408	1.278	0.796	0.414	0.291	0.419	0.298	0.404	0.286	0.854	0.492	0.670	0.421	1.557	0.821	0.427	0.304
	192	0.482	0.320	0.552	0.408	0.423	0.298	0.419	0.291	0.434	0.305	0.412	0.294	0.894	0.517	0.653	0.405	1.596	0.834	0.447	0.315
	336	0.528	0.354	0.572	0.414	0.444	0.312	0.437	0.314	0.449	0.313	0.439	0.310	0.853	0.471	0.707	0.445	1.621	0.841	0.478	0.333
	720	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	Avg	0.493	0.330	0.558	0.410	0.715	0.469	0.423	0.298	0.434	0.305	0.418	0.296	0.867	0.493	0.676	0.423	1.591	0.832	0.450	0.317
1st count		**29**		12		2		9		1		8		0		0		0		4	

Table 11: Full resutls of the ablation study.

Models		SSDA		w/o Structural Branch		w/o Spectral Branch		VisionTS+LoRA		VisionTS	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.343	0.372	0.373	0.386	0.344	0.372	0.473	0.428	0.355	0.386
	192	0.387	0.402	0.398	0.408	0.395	0.406	0.473	0.436	0.395	0.407
	336	0.409	0.420	0.418	0.424	0.405	0.416	0.464	0.439	0.419	0.421
	720	0.408	0.430	0.423	0.448	0.420	0.439	0.436	0.448	0.458	0.460
	Avg	0.387	0.406	0.403	0.417	0.391	0.408	0.462	0.438	0.407	0.419
ETTh2	96	0.267	0.328	0.279	0.336	0.274	0.329	0.402	0.399	0.288	0.334
	192	0.324	0.368	0.326	0.369	0.331	0.373	0.396	0.406	0.349	0.380
	336	0.347	0.392	0.351	0.393	0.350	0.393	0.397	0.415	0.364	0.398
	720	0.370	0.415	0.372	0.414	0.372	0.418	0.390	0.421	0.403	0.431
	Avg	0.327	0.376	0.332	0.378	0.332	0.378	0.396	0.410	0.351	0.386
ETTm1	96	0.283	0.328	0.327	0.347	0.302	0.338	0.319	0.342	0.284	0.332
	192	0.316	0.353	0.343	0.358	0.325	0.354	0.337	0.357	0.327	0.362
	336	0.340	0.370	0.368	0.382	0.349	0.373	0.355	0.373	0.354	0.382
	720	0.388	0.400	0.411	0.409	0.390	0.401	0.401	0.403	0.411	0.415
	Avg	0.332	0.363	0.362	0.374	0.342	0.367	0.353	0.369	0.344	0.373
ETTm2	96	0.168	0.256	0.218	0.281	0.179	0.264	0.206	0.276	0.174	0.262
	192	0.217	0.290	0.253	0.307	0.229	0.296	0.244	0.305	0.228	0.297
	336	0.268	0.323	0.292	0.334	0.277	0.330	0.284	0.331	0.281	0.337
	720	0.341	0.375	0.390	0.400	0.355	0.380	0.354	0.378	0.384	0.410
	Avg	0.249	0.311	0.288	0.331	0.260	0.318	0.272	0.323	0.267	0.327

Second, although models such as TimeLLM and Time-VLM employ substantially larger parameter sizes (up to 76M), they do not yield corresponding performance gains. In fact, their MSE values are consistently higher than those of smaller or medium-scale models. This suggests that **simply scaling up model parameters does not necessarily translate into better forecasting performance** in the time series domain.

Overall, SSDA lies on a more favorable region of the performance-efficiency trade-off curve, achieving a better balance among accuracy, parameter efficiency, and memory usage. These results demonstrate that our design not only improves forecasting performance but also maintains strong efficiency characteristics, making it more suitable for practical deployment scenarios.

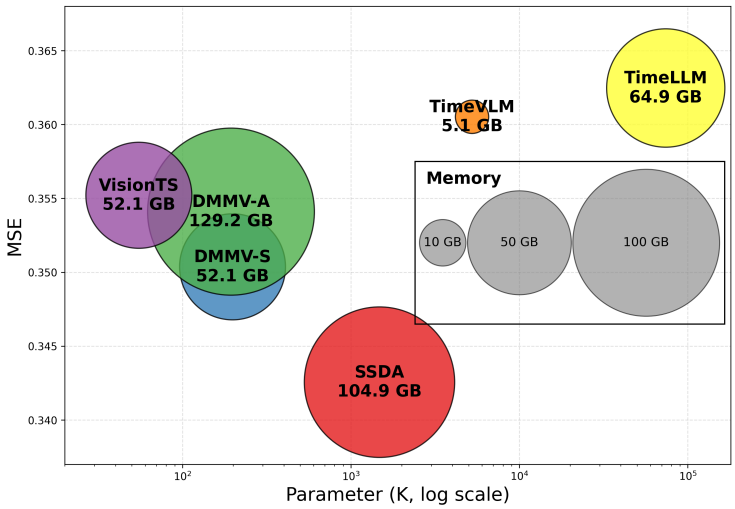


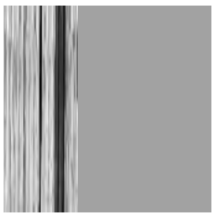
Figure 8: The bubble chart of performance vs. efficiency.

D.4 VISUALIZATION

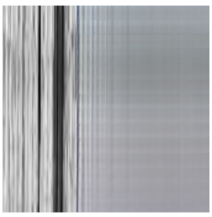
In this section, we present a comprehensive visualization analysis of SSDA from both branch-level and model-level perspectives. Specifically, we visualize the input and reconstructed images of both the spectral branch and the structural branch, enabling a direct comparison of their respective reconstruction behaviors and prediction characteristics. Furthermore, we compare the forecasting results of SSDA with those of DMMV by visualizing their predicted sequences alongside the ground truth. To provide a quantitative assessment, we also report the corresponding MSE and MAE values on selected samples.

The results consistently demonstrate that SSDA achieves superior prediction accuracy compared to DMMV. From the visualizations, SSDA produces predictions that are more closely aligned with the ground truth, exhibiting better trend fidelity and reduced noise. This improvement can be attributed to the complementary design of the two branches: the spectral branch enhances frequency-domain consistency, while the structural branch strengthens temporal dependency modeling. Together, these components effectively mitigate modality discrepancies, leading to more accurate and robust forecasting performance.

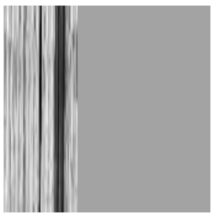
Beyond this overall improvement, an important and consistent phenomenon can be observed in the reconstructed images of the structural branch: clear vertical patterns persist across different samples. These vertical structures indicate that the model effectively captures and reinforces temporal consistency along the time axis, where each column corresponds to a temporal slice of the multivariate sequence, while horizontal patterns indicate underlying periodicity in the data. The emergence and stability of such patterns indicate that the structural branch suppresses spurious spatial correlations introduced during the image transformation process, while preserving meaningful temporal dependencies.



(a) Structural branch input image



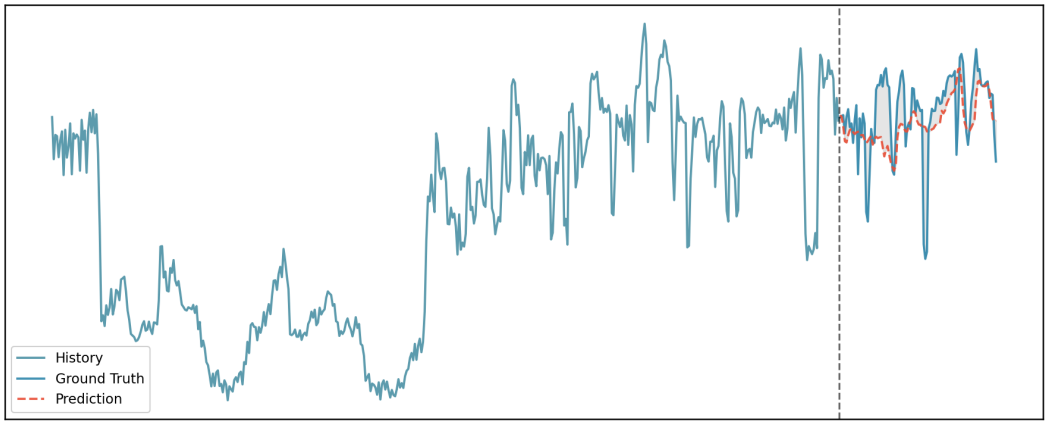
(b) Structural branch reconstructed image



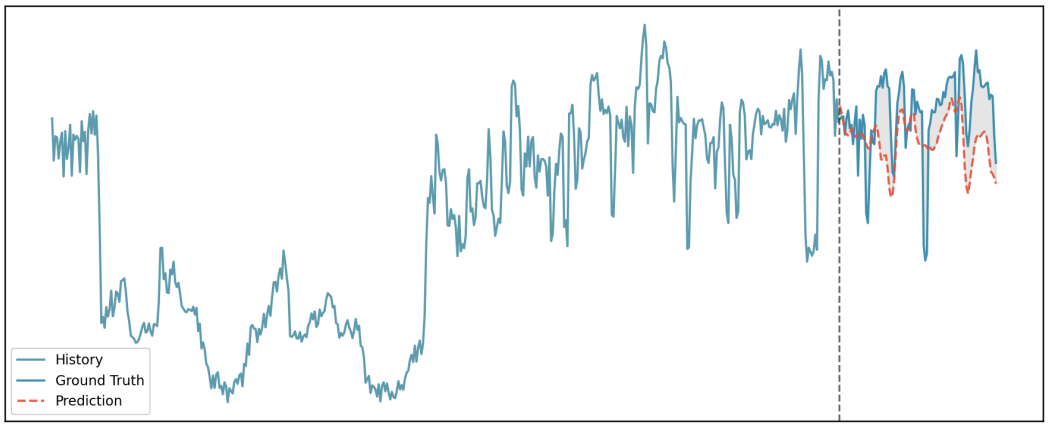
(c) Spectral branch input image



(d) Spectral branch reconstructed image

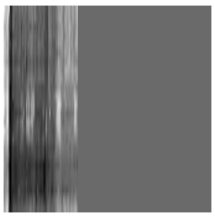


(e) SSDA (MSE=0.155,MAE=0.308).

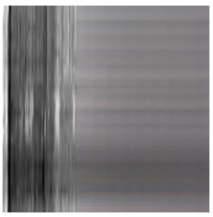


(f) DMMV (MSE=0.201,MAE=0.350).

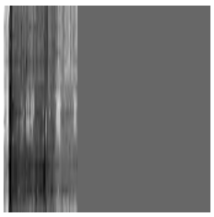
Figure 9: Forecasting visualization on a sample from ETTm2. (a) Input image of structural branch. (b) Output image of structural branch. (c) Input image of spectral branch(enhanced by Spectral Magnitude Aligner). (d) Output image of spectral branch. (e-f) Comparison of predictions and ground truths by SSDA and DMMV.



(a) Structural branch input image



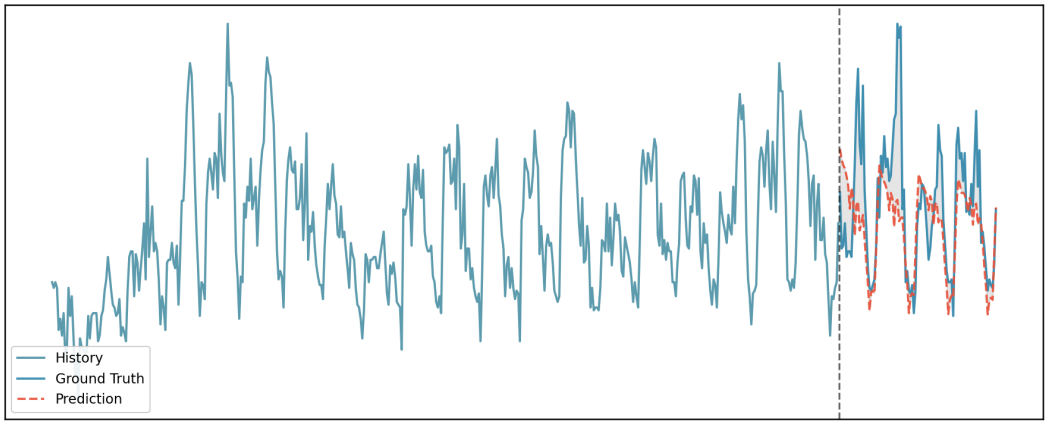
(b) Structural branch reconstructed image



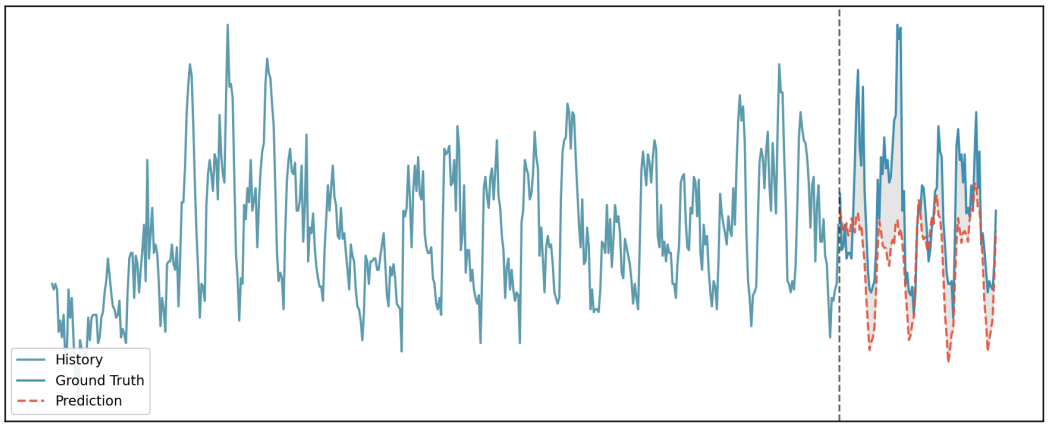
(c) Spectral branch input image



(d) Spectral branch reconstructed image

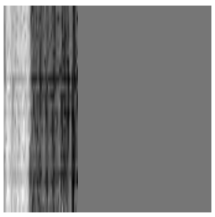


(e) SSDA (MSE=0.718,MAE=0.678).

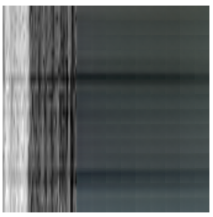


(f) DMMV (MSE=0.765,MAE=0.694).

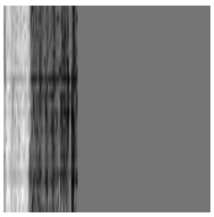
Figure 10: Forecasting visualization on a sample from ETTh1. (a) Input image of structural branch. (b) Output image of structural branch. (c) Input image of spectral branch(enhanced by Spectral Magnitude Aligner). (d) Output image of spectral branch. (e-f) Comparison of predictions and ground truths by SSDA and DMMV.



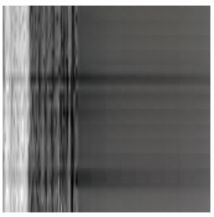
(a) Structural branch input image



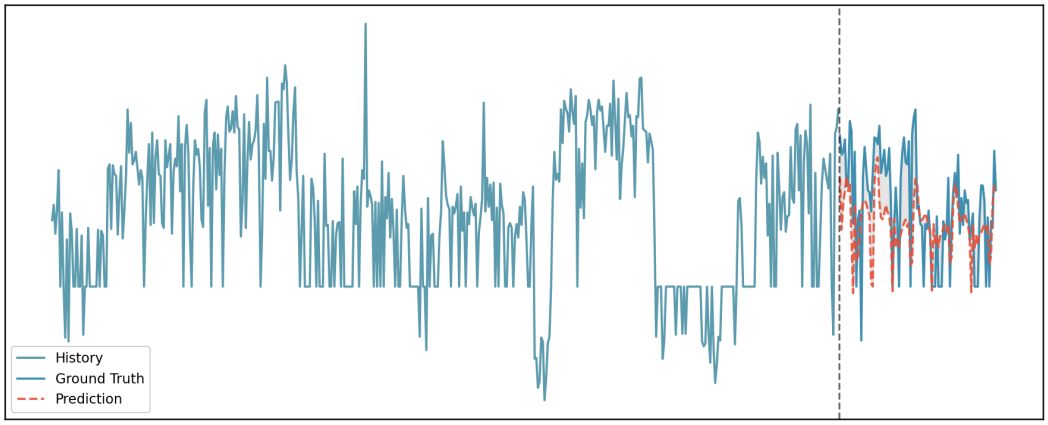
(b) Structural branch reconstructed image



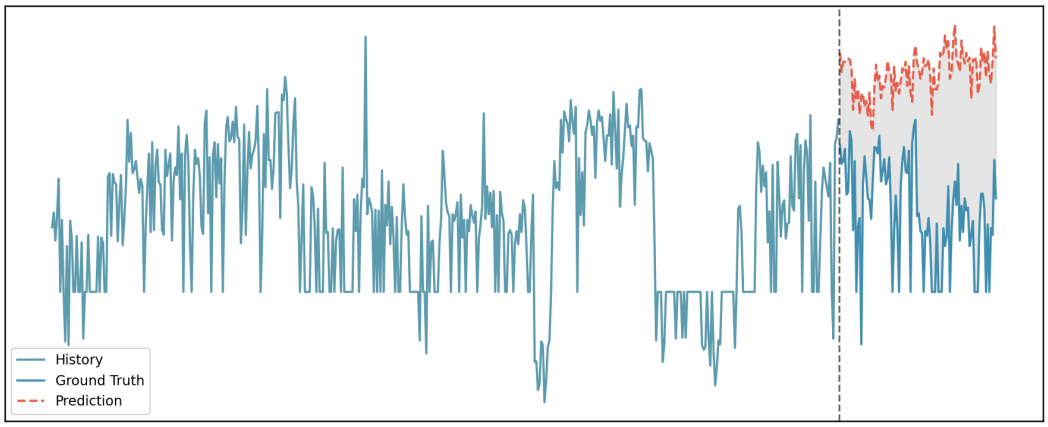
(c) Spectral branch input image



(d) Spectral branch reconstructed image



(e) SSDA (MSE=0.439,MAE=0.522).



(f) DMMV (MSE=2.242,MAE=1.370).

Figure 11: Forecasting visualization on a sample from ETTh2. (a) Input image of structural branch. (b) Output image of structural branch. (c) Input image of spectral branch(enhanced by Spectral Magnitude Aligner). (d) Output image of spectral branch. (e-f) Comparison of predictions and ground truths by SSDA and DMMV.