VISIONTS: Visual Masked Autoencoders Are Free-Lunch Zero-Shot Time Series Forecasters

Mouxiang Chen [1] **Lefei Shen** [1] **Zhuo Li** [2] **Xiaoyun Joy Wang** [2] **Jianling Sun** [1] **Chenghao Liu** [3]

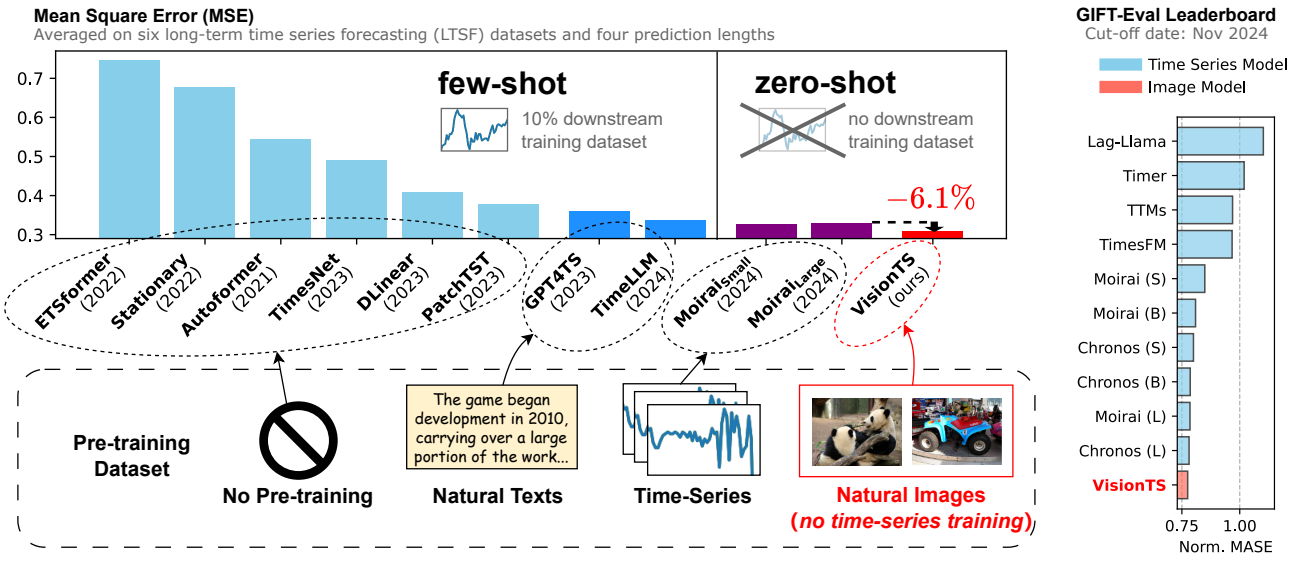


Figure 1. Long-term forecasting (left) and GIFT-Eval (right) performance comparison. Our VISIONTS, *without any training on time series data*, outperforms the pure time series foundation models in the zero-shot setting.

Abstract

Foundation models have emerged as a promising approach in time series forecasting (TSF). Existing approaches either repurpose large language models (LLMs) or build large-scale time series datasets to develop TSF foundation models for universal forecasting. However, these methods face challenges due to the severe cross-domain gap or in-domain heterogeneity. This paper explores a new road to building a TSF foundation model from rich, high-quality natural images. Our key insight is that a visual masked autoencoder, pre-trained on the ImageNet dataset, can naturally be a numeric series forecaster. By reformulating TSF as an image reconstruction task, we bridge the gap between image pre-training and TSF downstream tasks. Surprisingly, without further adaptation in the time series domain, the proposed VISIONTS could achieve better zero-shot forecast performance than existing TSF foundation models. With fine-tuning for one epoch, VISIONTS could further improve the forecasting and achieve state-of-the-art performance in most cases. Extensive experiments reveal *intrinsic similarities* between images and real-world time series, suggesting that visual models may offer a "free lunch" for TSF and highlight the potential for future cross-modality research. Our code is publicly available at https://github.com/Keytoyze/VisionTS.

[1]Zhejiang University [2]State Street Technology (Zhejiang) Ltd [3]Salesforce Research Asia. Correspondence to: Chenghao Liu <chenghao.liu@salesforce.com>, Zhuo Li <lizhuo@zju.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

1. Introduction

Foundation models (Bommasani et al., 2021) have revolutionized natural language processing (NLP) and computer

vision (CV) in recent years (Brown et al., 2020; He et al., 2022). By pretraining on large-scale data, they have shown remarkable few-shot and even zero-shot performance across various downstream tasks. This has motivated an emergent paradigm shift in time series forecasting (TSF), moving from a traditional one-model-per-dataset framework to *universal forecasting* with a single pre-trained model (Woo et al., 2024; Goswami et al., 2024). A TSF foundation model can greatly reduce the need for downstream data and demonstrate strong forecasting performance on diverse domains, such as energy consumption planning, weather forecasting, and traffic flow.

We have recently witnessed two roads to building a TSF foundation model for universal forecasting. The *first* tries to repurpose large language models (LLMs) that have been pre-trained on text data for TSF tasks (*i.e.*, **text-based**) (Zhou et al., 2023; Jin et al., 2024), based on the observation that LLMs and TSF models share a similar left-to-right forecasting paradigm. However, due to the significant gap between these two modalities, the effectiveness of such transferability between language and time series has recently been questioned by Tan et al. (2024).

The *second* road focuses on constructing large-scale time-series datasets collected from diverse domains to train a TSF foundation model from scratch (*i.e.*, time series-based or **TS-based**) (Woo et al., 2024; Das et al., 2024). Nevertheless, unlike images or language with unified formats, time series data is highly heterogeneous in length, frequency, number of variates, domains, and semantics, limiting the transferability between pre-training and downstream domains. Until recently, constructing a high-quality dataset remains challenging and is still in the early exploration stage.

In this paper, we investigate a *third* road that is less explored yet promising: building TSF foundation models with pre-trained *visual* models. Our key idea is that pixel variations in a natural image can be interpreted as temporal sequences, which share many intrinsic similarities with time series: ❶ **Similar modalities**: Unlike discrete texts, both images and time series are continuous; ❷ **Similar origin**: Both time series and images are observations of real-world physical systems, whereas languages are products of human cognitive processes; ❸ **Similar information density**: Languages are human-generated signals with high semantic density, while images and time series are natural signals with heavy redundancy (He et al., 2022); and ❹ **Similar features**: As shown in Section 1, images often display many features of real-world time series, which are rarely found in language data. Based on these findings, images could be a promising modality for transferring to TSF. We are motivated to answer the question: *Can a visual model pre-trained on images be a free-lunch foundation model for time series forecasting?*

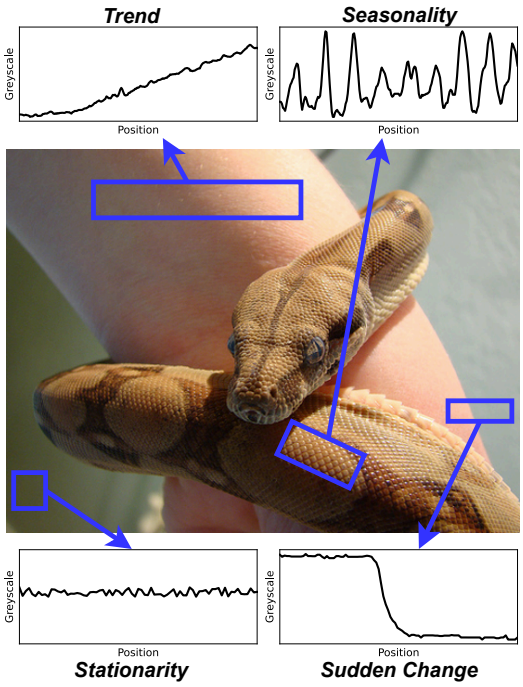


Figure 2. An image of the ImageNet dataset (Deng et al., 2009), in which the pixel arrays can display many well-known features of real-world time series, such as trend, seasonality, and stationarity (Qiu et al., 2024). By self-supervised pre-training on ImageNet, it is reasonable that a visual model could understand these features and exhibit a level of time series forecasting ability.

We focus on visual masked autoencoder (MAE)[1], a popular CV foundation model (He et al., 2022) by self-supervised pre-training on ImageNet (Deng et al., 2009). As an image reconstruction and completion model, MAE can naturally be a *numeric series forecaster*. Inspired by the well-known prompt technique in NLP (Schick & Schütze, 2021), we propose a simple method to reformulate TSF as a patch-level image reconstruction task to bridge the gap between pre-training and downstream tasks. Specifically, we transform 1D time-series data into 2D matrices via segmentation. Then, we render the matrices into images and align the forecasting window with masked image patches. This allows us to make a zero-shot forecast without further adaptation.

We evaluate our proposed VISIONTS on large-scale benchmarks, including 8 long-term TSF (Zhou et al., 2021), 29 Monash (Godahewa et al., 2021), and 23 GIFT-Eval (Aksu et al., 2024) datasets, spanning diverse domains, frequencies, and multivariates. To the best of our knowledge, **the scale of our evaluation benchmark is the largest among existing TSF foundation models**. As demonstrated in Fig. 1, *without* further adaptation on time series, a vanilla MAE can

[1]We use fonts to distinguish MAE (Masked Autoencoder) and MAE (Mean Absolute Error) in this paper.

surprisingly achieve a comparable performance or even outperform the strong zero-shot TSF foundation models. By fine-tuning MAE in each downstream dataset for a single epoch, VISIONTS can lead to SOTA performance in most long-term TSF benchmarks.

To further understand and explain the transferability, we use an MAE encoder to visualize both modalities, showing a level of similarity between time series and natural image representations. Additionally, **we observe considerable heterogeneity within time-series data across domains, and images can serve as a *bridge* to connect these isolated time-series representations**. This could further explain why VISIONTS performs better than some cross-domain TSF models. Our findings suggest that time series and natural images may be two sides of a coin, and visual models can be a *free lunch* for time series forecasting. We hope our findings inspire future cross-modal research on CV and TSF.

Our contributions are summarized as follows:

- We explore a road to building a TSF foundation model from natural images, conceptually different from the existing text-based and TS-based pre-training methods.
- We introduce VISIONTS, a novel TSF foundation model based on a visual MAE. To bridge the gap between the two modalities, we reformulate the TSF task into an image reconstruction task.
- Comprehensive evaluations of VISIONTS on large-scale benchmarks across multiple domains demonstrate its significant forecasting performance, surpassing few-shot text-based TSF foundation models and achieving comparable or superior results to zero-shot TS-based models.

2. Preliminaries

Time Series Forecasting (TSF) For a multivariate time series with M variables, let $\boldsymbol{x}_t \in \mathbb{R}^M$ represent the value at t-th time step. Given a historical sequence (*i.e.*, look-back window) $\boldsymbol{X}_{t-L:t} = [\boldsymbol{x}_{t-L}, \cdots, \boldsymbol{x}_{t-1}] \in \mathbb{R}^{L\times M}$ with context length L, the TSF task is to predict future values (*i.e.*, forecast horizon) with prediction length H: $\hat{\boldsymbol{X}}_{t:t+H} = [\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t+H-1}] \in \mathbb{R}^{H\times M}$.

Patch-Level Image Reconstruction To obtain high-quality visual representation for downstream CV tasks, He et al. (2022) proposed masked autoencoder (MAE) to pre-train a Vision Transformer (ViT) (Dosovitskiy et al., 2021) using a patch-level image reconstruction task on ImageNet. Specifically, for an image of size $W \times W$ (where W represents both the width and height, as ImageNet images are square), the image is evenly divided into $N \times N$ patches, each with a width and height of $S = W/N$. During pre-training, some random patches are masked, while the remaining *visible patches* are fed into the ViT with their position encodings. MAE are trained to reconstruct the masked pixel values from these visible patches.

3. Methodology

As noted in the Introduction, time series and images share intrinsic *similarities*, suggesting the transfer potential of pre-trained visual models (particularly MAE in this paper) for TSF. To reformulate TSF tasks into MAE's pre-training task, our high-level idea is straightforward: map the look-back/forecasting windows to visible/masked patches, respectively. This idea is supported by the prompt tuning (Schick & Schütze, 2021) in NLP, where the predictions for `[mask]` token in pre-trained language models, *e.g.*, BERT (Devlin et al., 2019), are directly used for downstream tasks. By unifying the forms of the two tasks, we bridge the gap between the two modalities without further training.

However, implementing this idea poses a challenge: the dimension of time-series data (1D) is different from images (2D). Moreover, the size of images in the pre-training dataset is fixed at 224×224, while the lengths of time series data can vary dynamically. In the following, we describe the details of VISIONTS to address this challenge. Our architecture is depicted in Fig. 3.

Segmentation Given a univariate input $X \in \mathbb{R}^L$, the first goal is to transform it into a 2D matrix. We propose to segment it into $\lfloor L/P \rfloor$ subsequences of length P, where P is the periodicity. Notably, when the time series lacks clear periodicity, we can set $P = 1$ directly, which is also effective in our experiments (Appendix B.6). In practice, P can be determined using statistical methods like Fast Fourier Transform (Wu et al., 2023; Chen et al., 2024) or domain knowledge like sampling frequency (Godahewa et al., 2021; Alexandrov et al., 2020). In this paper, we select P based on the sampling frequency, elaborated in Appendix A.2.

After that, these subsequences are then stacked into a 2D matrix, denoted by $\boldsymbol{I}_{\text{raw}} \in \mathbb{R}^{P\times\lfloor L/P \rfloor}$. This encoding strategy is proven to be efficient by recent work like TimesNet (Wu et al., 2023) and SparseTSF (Lin et al., 2024), as it allows for the simultaneous capture of both variations within the same period (*i.e.*, intra-period) and across periods with the same phase (*i.e.*, inter-period). Moreover, it ensures that each element in $\boldsymbol{I}_{\text{raw}}$ and its neighbors align with the *spatial locality* property of images (Krizhevsky et al., 2012), where nearby pixels tend to be similar due to the inherent cohesiveness of objects in the real world. Therefore, this further narrows the gap between time series and images.

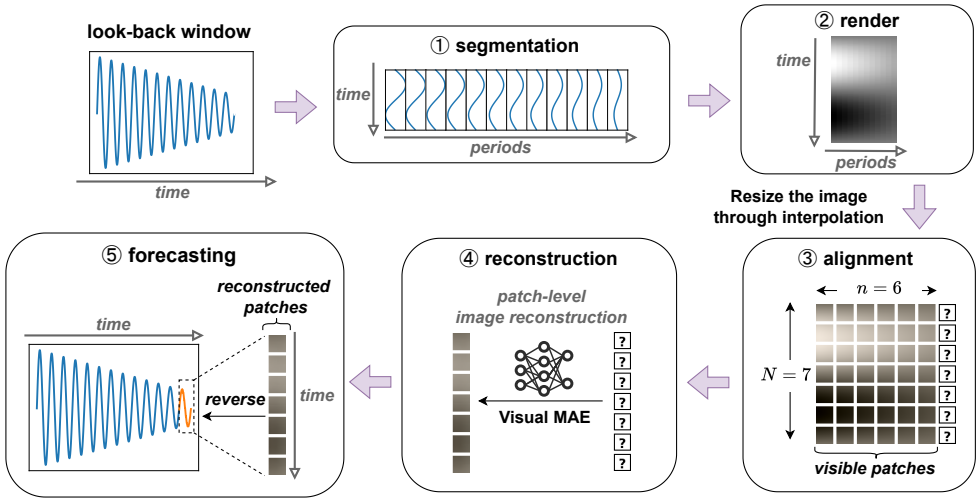


Figure 3. VISIONTS architecture. The input is first segmented by period, rendered into a grayscale image, and then aligned with the visible patches on the left through resampling. MAE is used to predict the masked patches on the right, and the reconstructed image is then reversed to forecasting.

Normalization MAE standardizes each image based on the mean and standard deviation computed on ImageNet. Therefore, we apply instance normalization to $\boldsymbol{I}_{\text{raw}}$, which is also a standard practice in current TSF (Kim et al., 2022). Notably, we observed that normalizing $\boldsymbol{I}_{\text{raw}}$ to a standard deviation of r, where r is a hyperparameter less than 1, yields superior performance. One explanation is that the magnitude of inputs/outputs during MAE pretraining is constrained by the limited range of color values. Therefore, reducing the magnitude of $\boldsymbol{I}_{\text{raw}}$ prevents exceeding these limits. However, an excessively low r can result in values that are difficult to distinguish. We found that a moderate value (0.4) of r performs well across most scenarios (See Appendix B.9 for more details). Let $\boldsymbol{I}_{\text{norm}}$ denote the normalized matrix, which is computed as follows:

$$\boldsymbol{I}_{\text{norm}} = r \cdot \frac{\boldsymbol{I}_{\text{raw}} - \text{Mean}(\boldsymbol{I}_{\text{raw}})}{\text{Standard-Deviation}(\boldsymbol{I}_{\text{raw}})}.$$

Rendering Since each image has three channels, we simply render $\boldsymbol{I}_{\text{norm}}$ as a grayscale image $\boldsymbol{I}_{\text{grey}} \in \mathbb{R}^{P \times \lfloor L/P \rfloor \times 3}$, where all three channels are identical to $\boldsymbol{I}_{\text{norm}}$. This choice is purely result-driven: In our early experiments, we added a convolutional layer with three output channels to convert the grayscale image into a color image and then fine-tuned it to find the optimal color transformation, which, however, did not significantly improve the performance.

Alignment Our goal is to predict the columns on the right of $\boldsymbol{I}_{\text{grey}}$ to forecast the future sequence. A straightforward approach is to treat $\boldsymbol{I}_{\text{grey}}$ as the visible left portion and the predicted columns as the masked right portion. However, since the image size during pre-training may not match the size of $\boldsymbol{I}_{\text{grey}}$, we propose to resize $\boldsymbol{I}_{\text{grey}}$ to align with the pre-training data. Formally, let the total number of 2D patches used in pre-training be $N \times N$ and the size of each patch be $S \times S$. We set the number of visible patches to $N \times n$ and the masked patches to $N \times (N - n)$, where $n = \lfloor N \cdot L/(L+H) \rfloor$ is determined by the ratio of context length L to prediction length H. We resample the image $\boldsymbol{I}_{\text{grey}}$ to adjust the size from the original dimensions $(P, \lfloor L/P \rfloor)$ to $(N \cdot S, n \cdot S)$, making it more compatible with MAE. We select *bilinear interpolation* for the resampling process.

Moreover, we found that reducing the width of the visible portion can further improve performance. One possible explanation is that MAE uses a large masked ratio during pre-training, with only 25% of patches visible. Reducing the image width may align the masked ratio more closely with pre-training. Therefore, we propose multiplying n by a hyperparameter $c \in [0, 1]$. Similar to r, we found that setting $c = 0.4$ performs well in our experiments (See Appendix B.9). This can be formulated as $n = \lfloor c \cdot N \cdot L/(L+H) \rfloor$.

Reconstruction and Forecasting After obtaining the MAE-reconstructed image, we simply reverse the previous steps for forecasting. Specifically, we resize the entire image back to the original time series segmentations through the same bilinear interpolation, and average the three channels to obtain a single-channel image. After de-normalizing and flattening, the forecasting window can be extracted.

Discussion on Multivariate Forecasting In addition to the temporal interactions, multivariate time series data sometimes show interactions between variables. While pre-

Table 1. Zero-shot or few-shot results on the long-term TSF benchmark. Results are averaged across prediction lengths {96, 192, 336, 720}, with full results in Table 9 (Appendix B.2). **Bold**: the best result.

Pretrain		Zero-Shot				Few-Shot (10% In-distribution Downstream Dataset)						
		Images	*Time series*			*Text*		*No Pretrain*				
Method		VISIONTS	MOIRAI$_S$	MOIRAI$_B$	MOIRAI$_L$	TimeLLM	GPT4TS	DLinear	PatchTST	TimesNet	Autoformer	Informer
ETTh1	MSE	**0.390**	0.400	0.434	0.510	0.556	0.590	0.691	0.633	0.869	0.702	1.199
	MAE	**0.414**	0.424	0.439	0.469	0.522	0.525	0.600	0.542	0.628	0.596	0.809
ETTh2	MSE	**0.333**	0.341	0.346	0.354	0.370	0.397	0.605	0.415	0.479	0.488	3.872
	MAE	**0.375**	0.379	0.382	0.377	0.394	0.421	0.538	0.431	0.465	0.499	1.513
ETTm1	MSE	**0.374**	0.448	0.382	0.390	0.404	0.464	0.411	0.501	0.677	0.802	1.192
	MAE	**0.372**	0.410	0.388	0.389	0.427	0.441	0.429	0.466	0.537	0.628	0.821
ETTm2	MSE	0.282	0.300	**0.272**	0.276	0.277	0.293	0.316	0.296	0.320	1.342	3.370
	MAE	0.321	0.341	0.321	**0.320**	0.323	0.335	0.368	0.343	0.353	0.930	1.440
Electricity	MSE	0.207	0.233	0.188	0.188	**0.175**	0.176	0.180	0.180	0.323	0.431	1.195
	MAE	0.294	0.320	0.274	0.273	0.270	**0.269**	0.280	0.273	0.392	0.478	0.891
Weather	MSE	0.269	0.242	0.238	0.260	**0.234**	0.238	0.241	0.242	0.279	0.300	0.597
	MAE	0.292	0.267	**0.261**	0.275	0.273	0.275	0.283	0.279	0.301	0.342	0.495
Average	MSE	**0.309**	0.327	0.310	0.329	0.336	0.360	0.407	0.378	0.491	0.678	1.904
	MAE	0.345	0.357	**0.344**	0.350	0.368	0.378	0.416	0.389	0.446	0.579	0.995
1st count		**7**	0	3	1	2	1	0	0	0	0	0

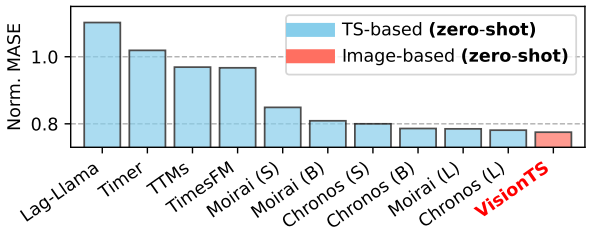


Figure 4. Performance on the GIFT-Eval Leaderboard (cut-off at VISIONTS's release).

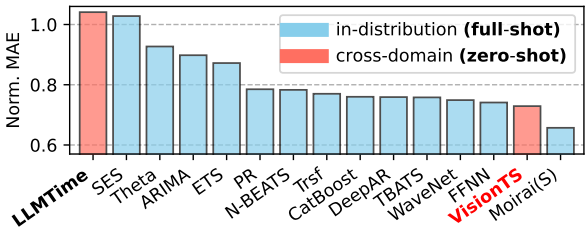


Figure 5. Aggregated results on the Monash TSF Benchmark, with full results in Table 15 (Appendix B.6).

trained vision models effectively capture temporal interactions based on the intrinsic similarities between images and time series, they struggle to capture inter-variable interactions due to a limited number of image channels, especially without further training. Fortunately, recent work shows that channel independence — forecasting each variable separately — can be effective and is widely used in recent deep forecasting models (Nie et al., 2022; Han et al., 2024; Jin et al., 2024; Zhou et al., 2023; Lin et al., 2024). Following these works, we adopt channel independence in our paper while leaving the exploration of capturing inter-variable interactions to future work.

4. Experiments

We follow the standard evaluation protocol proposed by Woo et al. (2024) to test our VISIONTS on 35 widely-used TSF benchmarks, and additionally evaluate it on the GIFT-Eval (Aksu et al., 2024) which is the *largest* TSF benchmark for zero-shot foundation models. We use `MAE` (Base) as our backbone by default. Baseline and benchmark details are elaborated in Appendix A.1.

4.1. Zero-Shot Time Series Forecasting

Setups We first evaluate VISIONTS's **zero-shot** TSF performance without fine-tuning on time-series modalities. To prevent data leakage, we selected six widely-used datasets from the long-term TSF benchmark that are not included in MOIRAI's pre-training set for evaluation. Since most baselines cannot perform zero-shot forecasting, we report their **few-shot** results by fine-tuning on the 10% of the individual target datasets. We also evaluate the Monash benchmark and GIFT-Eval benchmark. Notably, the Monash benchmark is more challenging for VISIONTS since they were used in MOIRAI's pre-training but not for VISIONTS. We set the hyperparameters to $r = c = 0.4$. Following common practice (Zhou et al., 2023; Woo et al., 2024), we conduct hyperparameter tuning on validation sets to determine the optimal context length L, detailed in Appendix B.1.

Table 2. Average MSE of different MAE variants, with full results in Table 17 (Appendix B.7).

	Base 112M	Large 330M	Huge 657M
ETTh1	0.390	**0.378**	0.391
ETTh2	**0.333**	0.340	0.339
ETTm1	**0.374**	0.379	0.383
ETTm2	**0.282**	0.286	0.284
Electricity	0.207	0.209	**0.202**
Weather	**0.269**	0.272	0.292
Avg.	**0.309**	0.311	0.315

Table 3. Computational cost in terms of seconds for forecasting a batch of 32 time series data.

Context Length	1k				1k	2k	3k	4k
Prediction Length	1k	2k	3k	4k	1k			
PatchTST	0.01	0.01	0.01	0.01	0.01	0.02	0.03	0.04
DeepAR	0.26	0.32	0.37	0.43	0.26	4.06	6.10	8.17
GPT4TS	0.01	0.01	0.01	0.02	0.01	0.03	0.04	0.06
MOIRAI$_{\text{Base}}$	0.03	0.04	0.04	0.05	0.03	0.04	0.05	0.06
TimesFM	0.08	0.14	0.20	0.27	0.07	0.13	0.20	0.25
LLMTime (8B)	> 200				> 200			
VISIONTS ($c = 0.4$)	0.04	0.03	0.03	0.03	0.04	0.04	0.05	0.05

Results on Long-Term TSF Benchmark Table 1 shows that VISIONTS surprisingly achieves the best forecasting performance in most cases (7 out of 14). Specifically, VISIONTS demonstrates a relative average MSE reduction of approximately 6% compared to MOIRAI$_{\text{Small}}$ and MOIRAI$_{\text{Large}}$, and performs comparably to MOIRAI$_{\text{Base}}$. When compared to the various few-shot baselines, VISIONTS shows a relative average MSE reduction ranging from 8% to 84%. Given that all baselines except for VISIONTS are trained on the time-series domain, this result is particularly encouraging. It suggests that **the transferability from images to time-series is stronger than from text to time-series, and even comparable to the in-domain transferability between time-series**. We also include a comparison with two TSF foundation models, TimesFM (Das et al., 2024) and LLMTime (Gruver et al., 2023), in Appendix B.3, as well as traditional algorithms (ETS, ARIMA, and Seasonal Naïve) in Appendix B.4. Results show that VISIONTS still outperforms all of these baselines.

Results on GIFT-Eval Benchmarks Fig. 4 shows the comparison of VISIONTS with six previously published TSF foundation models on the GIFT-Eval TSF Leaderboard[2], where VISIONTS surprisingly ranked first in terms of normalized MASE. It should be noted that although some concurrent works (after the release of VISIONTS) in the current leaderboard outperform VISIONTS, there may be data leakage issues for these works. In contrast, visual MAE was trained on ImageNet, long before the release of GIFT-Eval leaderboard, which can ensure no leakage.

Results on Monash Benchmark Fig. 5 shows the results aggregated from 29 Monash datasets, showing that VISIONTS in the zero-shot setting surpasses all models *individually* trained on each dataset and significantly outperforms the other cross-domain baseline (*i.e.*, LLMTime). It achieves second place among all baselines, just behind MOIRAI that pre-trained on *all* the training datasets. This promising result highlights VisionTS's strong zero-shot forecasting ability and effective cross-modality transferability.

[2]https://huggingface.co/spaces/Salesforce/GIFT-Eval

4.2. Further Analysis of VISIONTS

Backbone Analysis In Table 2 (full results in Appendix B.7), we observe that the overall performance of three MAE variants (112M, 330M, and 657M) outperforms MOIRAI$_{\text{Small}}$ and MOIRAI$_{\text{Large}}$. Particularly, larger models show a slight decrease in performance. This may be due to **larger visual models overfitting image-specific features, reducing their transferability**. A similar phenomenon was reported in MOIRAI, where larger models were found to degrade performance. We leave the exploration of scaling laws in image-based TSF foundation models for the future. Additionally, to explore the potential with other vision models, we also test LaMa (Suvorov et al., 2022), a visual inpainting model. Results in Appendix B.7 demonstrate that VISIONTS with LaMa performs similarly to MOIRAI in the zero-shot setting. This suggests that the performance is driven by the inherent similarity between images and time series, not solely by the MAE model.

Computational Cost We evaluate the computation cost of different baselines on an NVIDIA A800 GPU. Results are averaged on 90 runs. Table 3 shows the results between various TSF foundation models, showing that VISIONTS are comparable to MOIRAI$_{\text{Base}}$ and GPT4TS and faster than TimesFM, which is an auto-regressive model. While computation time increases with context length for all the other Transformer-based baselines, VISIONTS remains nearly constant. This is because VISIONTS encodes input sequences into an image with constant size, ensuring $O(1)$ efficiency. In contrast, Transformer-based methods operate at $O(L^2)$ relative to context length L.

Hyperparameter Analysis Appendix B.9 illustrates the impact of three hyperparameters. For context length L, as shown in Fig. 6, performance typically improves with increasing L, particularly on high-frequency datasets like

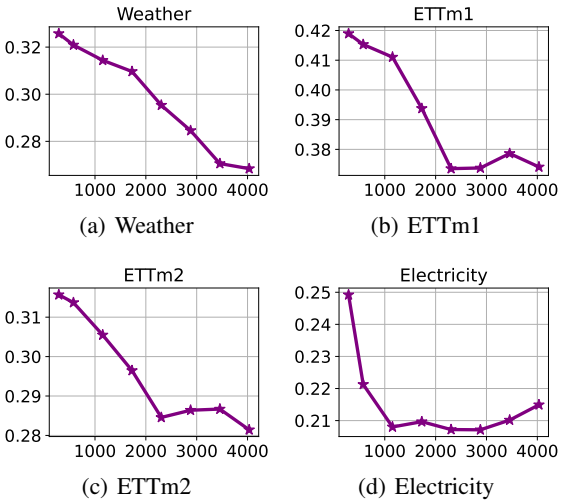


Figure 6. MSE (Y-axis) performance of different context lengths L (X-axis), averaged on four prediction lengths.

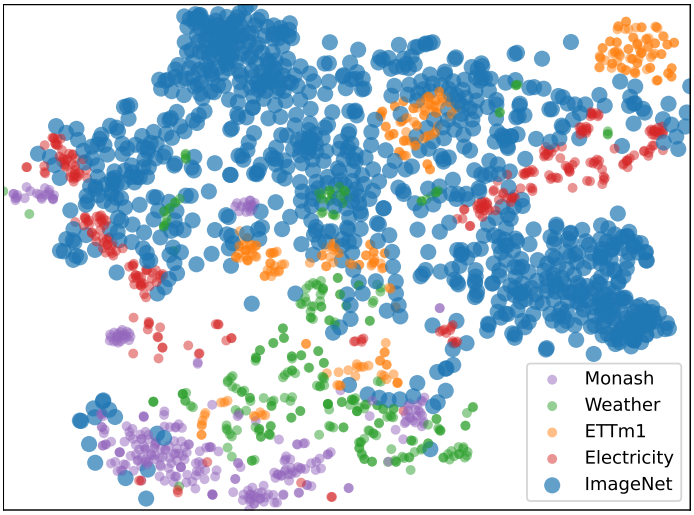


Figure 7. Modality visualization of the images (ImageNet) and time series (Monash, Weather, Electricity, and ETTm1) based on the MAE encoder.

Table 4. Aggregated full-shot forecasting performance on eight long-term TSF benchmarks (ETTh1, ETTh2, ETTm1, ETTm2, Illness, Weather, Traffic, and Electricity). VISIONTS is fine-tuned only a single epoch on each dataset except for Illness. Due to the space limit, we report the 1st count for each baseline, with full results in Table 21 (Appendix C.2).

Pretrain	***Images***	***Text***		***No Pretrain***							
Method	VISIONTS	Time-LLM	GPT4TS	Dlinear	PatchTST	TimesNet	FEDformer	Autoformer	Stationary	ETSformer	Informer
1st count	**46**	4	12	0	19	0	0	0	0	0	0

Weather (10-minute frequency) and ETTm1/ETTm2 (15-minute frequency). This aligns with other TSF foundation models like MOIRAI. As for the normalization constant r and alignment constant c, when both of them are around 0.4, performance is generally well across most benchmarks.

Modality Analysis: Where does the zero-shot forecastability come from? We further examine the gap between time series and images to explain the transferability of zero-shot forecasting. We sampled 1,000 images from ImageNet-1k and 300 samples from each time series dataset. We fed them into the MAE, maintaining a consistent image mask across all data. Fig. 7 visualizes the MAE encoder outputs of these data, which are flattened and reduced to 2-dimension by t-SNE. Notably, some time series, such as ETTm1 and Electricity, fall within the ImageNet distribution. **It suggests a relatively small gap between images and some time series (*e.g.*, Electricity and ETTm1), which could explain the good transferability**. Additionally, while ImageNet displays a concentrated distribution, time series are generally more scattered. For instance, ETTm1 clusters in the upper right, whereas Monash is found in the lower left, with a significant gap. **This indicates strong heterogeneity within time series data and suggests that images may serve as a bridge to connect isolated time series modality**.

Ablation Study We conduct experiments to validate our choices in the Alignment step, detailed in Appendix B.8. First, we test three different interpolation strategies, which shows that **Bilinear interpolation performs best**. Second, we apply horizontal and vertical flips on the image to examine whether the assumed left-to-right, top-to-bottom order is efficient. Results show that these changes do not significantly affect performance, suggesting that **image reconstruction is isotropic and not influenced by certain orientation**.

Qualitative Analysis: When does VISIONTS perform well, and when does it not? In Appendix D, we visualize the zero-shot forecasting of VISIONTS alongside the input and reconstruction images, highlighting both *successful* cases (where VISIONTS outperforms MOIRAI) and *failures* (where MOIRAI prevails). When the input exhibits strong regularity (Fig. 11), VISIONTS effectively forecasts both the periodicity (via segmentation) and trends (via MAE's capabilities). In contrast, MOIRAI, akin to seasonal naïve methods, struggles to capture inter-period trends. For less-structured input (Figs. 12 to 14), MOIRAI adopts a conservative approach with lower volatility to minimize errors, while VISIONTS takes a more aggressive stance. This strategy occasionally yields more accurate trend predictions (Figs. 12

and 13) but may also result in greater MAE (Fig. 14).

4.3. Full-Shot Long-Term Time Series Forecasting

Setups We evaluate the full-shot capability of each baseline trained on individual long-term TSF benchmarks. In addition to the six datasets used for zero-shot forecasting, we also include the popular Traffic and Illness datasets. As self-attention and feed-forward layers contain rich knowledge that can be transferred to TSF, we choose to **fine-tune only the layer normalization (LN) layers while freezing the other parameters**, which is also adopted by Zhou et al. (2023). Training details are elaborated in Appendix C.1.

Main Results Table 4 summarizes the full-shot results, with the full results and standard deviations detailed in Appendix C.2. It shows that VISIONTS outperforms other baselines in most cases (46 out of 80), surpassing the non-pretrained PatchTST and the language-pretrained GPT4TS. Remarkably, except for Illness with the least data, VISIONTS demands **only a single epoch of fine-tuning**. This suggests that even minimal fine-tuning enables VisionTS to adapt to time series effectively. Compared with Table 1, fine-tuning provides limited benefits for ETTh1 and ETTh2 but significantly improves other datasets. We attribute this to the smaller data scale of ETTh1 and ETTh2.

Ablation Study Tan et al. (2024) proposed several ablation variants for text-based foundation models, including **w/o LLM** (removing the LLM), **LLM2Attn/LLM2Trsf** (replacing the LLM with a single self-attention/Transformer layer), and **RandLLM** (randomly initializing the LLM). They found no significant performance differences and concluded that textual knowledge is unnecessary for TSF. We conducted similar ablations to assess the role of the vision model (VM), including **w/o VM**, **VM2Attn**, **VM2Trsf**, and **RandVM**. Appendix C.3 shows that these variants lead to worse performance, indicating that visual knowledge is beneficial for TSF.

Analysis: Fine-tuning strategies As stated before, we fine-tune only the layer normalization (LN). We also tested fine-tuning the bias, MLP, or attention layers, in addition to full fine-tuning and freezing. All hyperparameters were kept constant. Note that freezing differs from the previous zero-shot experiment, where a longer context length was used. Appendix C.3 show that fine-tuning LN is the best. Modifying MLP or attention layers results in significant performance drops, suggesting that valuable knowledge resides in these components.

5. Related Work

Depending on the pre-training data, TSF foundation models can be categorized into Text-based and TS-based models. We first review related works and then introduce recent research on image-based time series analysis.

Text-based TSF Foundation Models Large Language Models (LLMs) pre-trained on large amounts of text data are being applied to TSF tasks. For example, Zhou et al. (2023) fine-tuned a pre-trained GPT (Radford et al., 2019) on each time-series downstream task, such as forecasting, classification, imputation, and anomaly detection. Based on Llama (Touvron et al., 2023), Jin et al. (2024) froze the pre-trained LLM and reprogrammed the time series to align with the language modality. Bian et al. (2024) adopted a two-stage approach by continually pre-training GPT (Radford et al., 2019) on the time-series domain. Nevertheless, the TSF performance of LLMs has recently been questioned by Tan et al. (2024), which designed several ablation studies to show that textual knowledge is unnecessary for forecasting. In this paper, we attribute it to the large modality gap. Some recent approaches focus on directly transforming the time series into natural texts for LLMs, allowing for zero-shot forecasting. For example, PromptCast (Xue & Salim, 2023) used pre-defined templates to describe numerical time series data, while LLMTime (Gruver et al., 2023) directly separated time steps using commas and separates digits using spaces to construct the text input. However, due to the efficiency issue of the autoregressive decoding strategy and the expensive inference cost of large language models, their practical use is limited.

Time Series-Based TSF Foundation Models Self-supervised pre-training a TSF model on the same dataset used for downstream TSF tasks is a well-explored topic (Ma et al., 2023; Zhang et al., 2024), such as denoising autoencoders (Zerveas et al., 2021) or contrastive learning (Woo et al., 2022a; Yue et al., 2022). They follow a similar paradigm to the masked autoencoder (MAE) in computer vision, which is a well-studied topic in other machine learning fields, such as BERT (Devlin et al., 2019), CBraMod (Wang et al., 2025), and HuBERT (Hsu et al., 2021). However, these methods rarely examine the cross-dataset generalization capabilities. Recently, research has shifted towards training universal foundation models, by collecting large-scale time series datasets from diverse domains (Ansari et al., 2024; Goswami et al., 2024; Liu et al., 2024; Das et al., 2024; Dong et al., 2024; Feng et al., 2024) or generating numerous synthetic time series data (Fu et al., 2024; Yang et al., 2024). As a representative method, Woo et al. (2024) collected 27 billion observations across nine domains and trained TSF foundation models of various scales, achieving strong zero-shot performance. However, given the severe

heterogeneity, constructing high-quality large datasets poses significant challenges for building these foundation models.

Image-Based Time-Series Analysis Previous research has investigated encoding time series data into images and used convolutional neural networks (CNNs) trained from scratch for classification (Wang & Oates, 2015a;b; Hatami et al., 2018) or forecasting (Li et al., 2020; Sood et al., 2021; Semenoglou et al., 2023). Recent researchers explored using pre-trained models for these imaging time series. Li et al. (2024) used a pre-trained vision transformer (ViT) for classification. Wimmer & Rekabsaz (2023) and Zhang et al. (2023) employed vision-language multimodal pre-trained models to extract predictive features and generate text descriptions. Yang et al. (2024) generated synthetic time series data to pre-train a vision model for the TSF task. However, these studies did not deeply examine the transferability from natural images to TSF. Despite early efforts by Zhou et al. (2023) to fine-tune a BEiT (Bao et al., 2022) trained on images for time series forecasting, it still falls short of the leading text-based and TS-based TSF foundation models. To the best of our knowledge, we are the first to show that an image-based foundation model, without further time-series adaptation, can match or even surpass other types of TSF foundation models.

6. Conclusion

In this paper, we explore a novel approach to building a time series forecasting (TSF) foundation model using natural images, offering a new perspective distinct from the traditional text-based and TS-based methods. By leveraging the intrinsic similarities between images and time series, we introduced VISIONTS, an MAE-based TSF foundation model that reformulates the TSF task as an image reconstruction problem. Our extensive evaluations demonstrate that VISIONTS achieves outstanding forecasting performance in zero-shot and full-shot settings, being a free lunch for a TSF foundation model. We hope our findings could open new avenues for further cross-modality research.

Limitations and Future Directions. (1) As a preliminary study, we employed MAE and LaMa. Utilizing more advanced models like diffusion models (Rombach et al., 2022; Peebles & Xie, 2023) presents a promising research direction. (2) Due to limitations in the visual model, VISIONTS cannot capture multivariate interactions and perform distribution forecasting. Future modifications to the model structure may empower it with more time series capabilities.

Acknowledgments

Research work mentioned in this paper is supported by State Street Zhejiang University Technology Center. We would also like to thank reviewers for their valuable comments.

Impact Statement

This paper presents work whose goal is to advance the field of time series forecasting. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Aksu, T., Woo, G., Liu, J., Liu, X., Liu, C., Savarese, S., Xiong, C., and Sahoo, D. Gift-eval: A benchmark for general time series forecasting model evaluation, 2024. URL https://arxiv.org/abs/2410.10393.

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., Stella, L., Türkmen, A. C., and Wang, Y. GluonTS: Probabilistic and Neural Time Series Modeling in Python. *Journal of Machine Learning Research*, 21(116):1–6, 2020. URL http://jmlr.org/papers/v21/19-820.html.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Bao, H., Dong, L., Piao, S., and Wei, F. BEit: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=p-BhZSz59o4.

Bian, Y., Ju, X., Li, J., Xu, Z., Cheng, D., and Xu, Q. Multi-patch prediction: Adapting llms for time series representation learning. *arXiv preprint arXiv:2402.04852*, 2024.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Chen, M., Shen, L., Fu, H., Li, Z., Sun, J., and Liu, C. Calibration of time-series forecasting: Detecting and adapting context-driven distribution shift. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 341–352, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671926. URL https://doi.org/10.1145/3637528.3671926.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. 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.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Dong, J., Wu, H., Wang, Y., Qiu, Y.-Z., Zhang, L., Wang, J., and Long, M. Timesiam: A pre-training framework for siamese time-series modeling. In *Forty-first International Conference on Machine Learning*, 2024.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Ekambaram, V., Jati, A., Dayama, P., Mukherjee, S., Nguyen, N., Gifford, W. M., Reddy, C., and Kalagnanam, J. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *Advances in Neural Information Processing Systems*, 37:74147–74181, 2024.

Feng, C., Huang, L., and Krompass, D. Only the curve shape matters: Training foundation models for zero-shot multivariate time series forecasting through next curve shape prediction. *arXiv preprint arXiv:2402.07570*, 2024.

Fu, F., Chen, J., Zhang, J., Yang, C., Ma, L., and Yang, Y. Are synthetic time-series data really not as good as real data?, 2024. URL https://arxiv.org/abs/2402.00607.

Godahewa, R. W., Bergmeir, C., Webb, G. I., Hyndman, R., and Montero-Manso, P. Monash time series forecasting archive. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wEc1mgAjU-.

Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., and Dubrawski, A. Moment: A family of open time-series foundation models. In *Forty-first International Conference on Machine Learning*, 2024.

Gruver, N., Finzi, M., Qiu, S., and Wilson, A. G. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2023.

Han, L., Ye, H.-J., and Zhan, D.-C. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge & Data Engineering*, (01): 1–14, 2024.

Hatami, N., Gavet, Y., and Debayle, J. Classification of time-series images using deep convolutional neural networks. In *Tenth international conference on machine vision (ICMV 2017)*, volume 10696, pp. 242–249. SPIE, 2018.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J. Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y.-F., Pan, S., et al. Time-llm: Time series forecasting by reprogramming large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Li, X., Kang, Y., and Li, F. Forecasting with time series imaging. *Expert Systems with Applications*, 160:113680, 2020.

Li, Z., Li, S., and Yan, X. Time series as images: Vision transformer for irregularly sampled time series. *Advances in Neural Information Processing Systems*, 36, 2024.

Lin, S., Lin, W., Wu, W., Chen, H., and Yang, J. Sparsetsf: Modeling long-term time series forecasting with 1k parameters. In *Forty-first International Conference on Machine Learning*, 2024.

Liu, P., Guo, H., Dai, T., Li, N., Bao, J., Ren, X., Jiang, Y., and Xia, S.-T. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18915–18923, 2025.

Liu, Y., Wu, H., Wang, J., and Long, M. Non-stationary transformers: Exploring the stationarity in time series forecasting, 2022.

Liu, Y., Zhang, H., Li, C., Huang, X., Wang, J., and Long, M. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning*, 2024.

Ma, Q., Liu, Z., Zheng, Z., Huang, Z., Zhu, S., Yu, Z., and Kwok, J. T. A survey on time-series pre-trained models. *arXiv preprint arXiv:2305.10716*, 2023.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Qiu, X., Hu, J., Zhou, L., Wu, X., Du, J., Zhang, B., Guo, C., Zhou, A., Jensen, C. S., Sheng, Z., and Yang, B. TFB: towards comprehensive and fair benchmarking of time series forecasting methods. *Proc. VLDB Endow.*, 17(9): 2363–2377, 2024.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Schick, T. and Schütze, H. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 255–269, 2021.

Semenoglou, A.-A., Spiliotis, E., and Assimakopoulos, V. Image-based time series forecasting: A deep convolutional neural network approach. *Neural Networks*, 157: 39–53, 2023.

Shi, X., Wang, S., Nie, Y., Li, D., Ye, Z., Wen, Q., and Jin, M. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*, 2024.

Sood, S., Zeng, Z., Cohen, N., Balch, T., and Veloso, M. Visual time series forecasting: an image-driven approach. In *Proceedings of the Second ACM International Conference on AI in Finance*, pp. 1–9, 2021.

Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., and Lempitsky, V. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2149–2159, 2022.

Tan, M., Merrill, M. A., Gupta, V., Althoff, T., and Hartvigsen, T. Are language models actually useful for time series forecasting? *arXiv preprint arXiv:2406.16964*, 2024.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Wang, J., Zhao, S., Luo, Z., Zhou, Y., Jiang, H., Li, S., Li, T., and Pan, G. CBramod: A criss-cross brain foundation model for EEG decoding. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=NPNUHgHF2w.

Wang, Z. and Oates, T. Imaging time-series to improve classification and imputation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 3939–3945, 2015a.

Wang, Z. and Oates, T. Spatially encoding temporal correlations to classify temporal data using convolutional neural networks. *arXiv preprint arXiv:1509.07481*, 2015b.

Wimmer, C. and Rekabsaz, N. Leveraging vision-language models for granular market change prediction. *arXiv preprint arXiv:2301.10166*, 2023.

Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=PilZY3omXV2.

Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. C. H. Etsformer: Exponential smoothing transformers for time-series forecasting. *CoRR*, abs/2202.01381, 2022b. URL https://arxiv.org/abs/2202.01381.

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., and Sahoo, D. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024.

Wu, H., Xu, J., Wang, J., and Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Xue, H. and Salim, F. D. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2023.

Yang, L., Wang, Y., Fan, X., Cohen, I., Zhao, Y., and Zhang, Z. Vitime: A visual intelligence-based foundation model for time series forecasting. *arXiv preprint arXiv:2407.07311*, 2024.

Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., and Xu, B. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8980–8987, 2022.

Zaken, E. B., Goldberg, Y., and Ravfogel, S. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., and Eickhoff, C. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2114–2124, 2021.

Zhang, K., Wen, Q., Zhang, C., Cai, R., Jin, M., Liu, Y., Zhang, J. Y., Liang, Y., Pang, G., Song, D., et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Zhang, Y., Zhang, Y., Zheng, M., Chen, K., Gao, C., Ge, R., Teng, S., Jelloul, A., Rao, J., Guo, X., et al. Insight miner: A time series analysis dataset for cross-domain alignment with natural language. In *NeurIPS 2023 AI for Science Workshop*, 2023.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. 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.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pp. 27268–27286. PMLR, 2022.

Zhou, T., Niu, P., Sun, L., Jin, R., 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.

Appendix

A. Details of Experiments

A.1. Benchmark and baselines

Long-Term TSF Benchmark We evaluate our model on 8 widely used long-term TSF datasets (Zhou et al., 2021; Wu et al., 2021), including ETTh1, ETTh2, ETTm1, ETTm2, Electricity, Traffic, Illness, and Weather. Performance is assessed using Mean Squared Error (MSE) and Mean Absolute Error (MAE), with lower values indicating better forecasting accuracy.

Monash Benchmark Following Woo et al. (2024), we tested 29 Monash datasets (Godahewa et al., 2021) using GluonTS (Alexandrov et al., 2020), including M1 Monthly, M3 Monthly, M3 Other, M4 Monthly, M4 Weekly, M4 Daily, M4 Hourly, Tourism Quarterly, Tourism Monthly, CIF 2016, Australian Electricity Demand, Bitcoin, Pedestrian Counts, Vehicle Trips, KDD Cup, Weather, NN5 Daily, NN5 Weekly, Carparts, FRED-MD, Traffic Hourly, Traffic Weekly, Rideshare, Hospital, COVID Deaths, Temperature Rain, Sunspot, Saugeen River Flow, and US Births. Performance is assessed using MAE.

GIFT-Eval Benchmark Aksu et al. (2024) introduces the General Time Series Forecasting Model Evaluation, GIFT-Eval, encompasses 23 datasets over 144,000 time series and 177 million data points, spanning seven domains, 10 frequencies, multivariate inputs, and prediction lengths ranging from short to long-term forecasts. We use a constant context length 2,000 for VISIONTS and we report the point forecast performance using MAPE.

Baselines We select representative baselines for comparison, including **TS-based** and **Text-based** foundation models, and **other popular TSF baselines** covering both Transformer-based, MLP-based and CNN-based architectures. The baseline models selected for comparison are briefly described below:

1. **MOIRAI** (Woo et al., 2024) is a TSF foundation model trained on the Large-scale Open Time Series Archive (LOTSA), with over 27B observations across nine domains. It has three variants: **small**, **base**, and **large**.
2. **TimesFM** (Das et al., 2024) is a decoder-style TSF foundation model, using a large time-series corpus comprising both real-world and synthetic datasets.
3. **Time-LLM** (Jin et al., 2024) is a text-based TSF foundation model built on Llama, which reprograms time series data to align with the language modality, keeping the LLM frozen.
4. **GPT4TS** (Zhou et al., 2023) (OneFitsAll) is another text-based model based on GPT, fine-tuned for forecasting tasks.
5. **LLMTime** (Gruver et al., 2023) encodes time series data to a text sequence, supporting zero-shot forecasting.
6. **DLinear** (Zeng et al., 2023) proposes a linear forecasting model, enhanced by seasonal-trend decomposition or normalization.
7. **PatchTST** (Nie et al., 2022) uses Transformer encoders with patching and channel independence techniques for improved predictions.
8. **TimesNet** (Wu et al., 2023) applies convolution kernels along the time dimension, using temporal decomposition and periodical segmentation to capture temporal patterns.
9. **FEDformer** (Zhou et al., 2022) employs a sparse frequency domain representation, using frequency-enhanced blocks for cross-time dependency.
10. **Autoformer** (Wu et al., 2021) uses series decomposition blocks and Auto-Correlation to capture cross-time dependency.
11. **Stationary** (Liu et al., 2022) introduces stationarization and de-stationary attention mechanisms.
12. **ETSFormer** (Woo et al., 2022b) leverages exponential smoothing principles, including exponential smoothing and frequency attention mechanisms.
13. **Informer** (Zhou et al., 2021) proposes ProbSparse self-attention and distillation operations.

Table 5. Periodicity (P) search range for the sampling frequency. x denotes the number of sampling frequencies. For example, for data with a sampling frequency of 2 minutes (2T), we have $x = 2$, and the possible search range of P is $\{^{1440}/_{x}, ^{10080}/_{x}, 1\} = \{720, 5040, 1\}$.

Sampling Frequency	Possible Seasonalities	Possible P
Second (S)	1 hour	$\{^{3600}/_{x}, 1\}$
Minute (T)	1 day or 1 week	$\{^{1440}/_{x}, ^{10080}/_{x}, 1\}$
Hour (H)	1 day or 1 week	$\{^{24}/_{x}, ^{168}/_{x}, 1\}$
Day (D)	1 week, 1 month, or 1 year	$\{^{7}/_{x}, ^{30}/_{x}, ^{365}/_{x}, 1\}$
Week (W)	1 year or 1 month	$\{^{52}/_{x}, ^{4}/_{x}, 1\}$
Month (M)	1 year, 6 months, or 3 months	$\{^{12}/_{x}, ^{6}/_{x}, ^{3}/_{x}, 1\}$
Business Day (B)	1 week	$\{^{5}/_{x}, 1\}$
Quarter (Q)	1 year or 6 months	$\{^{4}/_{x}, ^{2}/_{x}, 1\}$
Others	-	$\{1\}$

Table 6. Final P used for each dataset in our experiment.

	Frequency	P	Datasets			
Long-Term TSF	H	24	ETTh1	ETTh2	Electricity	Traffic
	W	52	Illness			
	15T	96	ETTm1	ETTm2		
	10T	144	Weather			
Monash	D	1	M4 Daily	COVID Deaths		
	W	1	NN5 Weekly			
	M	1	FRED-MD			
	Q	1	M3 Other			
	M	3	M3 Monthly	M4 Monthly	CIF 2016 (6)	
	W	4	M4 Weekly	Traffic Weekly		
	Q	4	Tourism Quarterly			
	M	6	CIF 2016 (12)	Car Parts		
	D	7	Bitcoin	Vehicle Trips	Weather	NN5 Daily
	D	7	US Births	Saugeen Day	Temperature Rain	
	M	12	Tourism Monthly	Hospital	M1 Monthly	
	H	24	M4 Hourly	KDD cup	Pedestrian Counts	
	H	24	Traffic Hourly	Rideshare		
	D	30	Sunspot			
	0.5H	336	Aus. Elec. Demand			

Table 7. Comparison of setting $P = 1$ for VISIONTS.

	VISIONTS		$P = 1$	
	MSE	MAE	MSE	MAE
ETTh1	0.390	0.414	0.840	0.628
ETTh2	0.333	0.375	0.424	0.445
ETTm1	0.374	0.372	0.660	0.533
ETTm2	0.282	0.321	0.312	0.363
Average	0.344	0.370	0.559	0.492

For the long-term TSF benchmark, we include TS-based foundation model results from their original papers, Text-based model results from Tan et al. (2024), and other baseline results from Zhou et al. (2023). For the Monash and PF benchmark, we include results from Woo et al. (2024).

Environment All experiments are conducted using *Time-Series-Library* (https://github.com/thuml/Time-Series-Library) and GluonTS library (Alexandrov et al., 2020) on an NVIDIA A800 GPU.

A.2. Periodicity selection

We first determine a range of period lengths based on the sampling frequency of the data, shown in Table 5. This frequency-based strategy is also employed by Alexandrov et al. (2020) while we extend the search range for tuning. We select the optimal P from this range on the validation set. The final P used in our experiments are summarized in Table 6.

To demonstrate the influence of P and the effectiveness of our periodicity selection strategy, we set $P = 1$ and compare the results with the above strategy. Table 7 shows that such strategy (denoted as VISIONTS) significantly outperforms the naive strategy that sets $P = 1$.

B. Zero-Shot Forecasting

B.1. Hyperparameters

Table 8. Hyperparameters for VISIONTS used in our zero-shot forecasting (Long-term TSF).

	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Electricity
Normalization constant r	0.4	0.4	0.4	0.4	0.4	0.4
Alignment constant c	0.4	0.4	0.4	0.4	0.4	0.4
Context length L	2880	1728	2304	4032	4032	2880

We conduct hyperparameter tuning on validation sets to determine the optimal context length L. Final used hyperparameters are summarized in Table 8.

B.2. Full forecasting results of the long-term TSF benchmark

Table 9 shows the full results of zero-shot/few-shot long-term forecasting performance. VISIONTS achieves the best results in most cases (32 out of 62), outperforming MOIRAI$_{\text{Base}}$ (10 out of 62) and MOIRAI$_{\text{Large}}$ (8 out of 62).

B.3. Comparison of TimesFM and LLMTime

Due to the step-by-step output of the decoder architecture, the efficiency of TimesFM (Das et al., 2024) and LLMTime (Gruver et al., 2023) are relatively slower. Thus, Das et al. (2024) only reported results for the last test window of the original split. We compared VISIONTS with their results under the same setting, as shown in Table 10. VISIONTS outperforms TimesFM and LLMTime in terms of MAE, indicating that image-based TSF models are on par with or even better than TS-based and text-based models.

B.4. Comparison of traditional methods

In addition to deep learning models, we also compare traditional methods, including ARIMA, ETS, and two methods that require periodicity as our VISIONTS: Seasonal Naïve (repeating the last period) and Seasonal Avg (similar to Seasonal Naïve but repeating the average of all periods in the look-back window). Due to the high computational cost of ARIMA and ETS, we only compare them on the small-scale benchmarks, *i.e.*, four ETT datasets. Table 12 shows that VISIONTS also achieves the best performance.

B.5. Comparison of concurrent works

We compare our work with other concurrent TSF methods. Table 13 presents the comparison from Time-MoE (Shi et al., 2024) and TTM (Ekambaram et al., 2024), and Table 14 shows the comparison with CALF (Liu et al., 2025), which is the existing SOTA LLMs-based time series forecasting work. These findings highlight the promising potential of vision models in TSF scenarios.

B.6. Full forecasting results of the Monash TSF benchmark

Setup Table 6 lists the sampling frequency and the selected period P for each dataset. Datasets with $P = 1$ indicate no significant periodicity, where we use a context length of $L = 300$. For other datasets with $P > 1$, we select a longer context

Table 9. Full results of Table 1: Zero-shot or few-shot results on the long-term TSF benchmark. **Bold**: the best result.

		Zero-Shot								Few-Shot (10% Downstream Dataset)													
Pretrain		*Images*		*Time-series*						*Text*				*No Pretrain*									
Method		VISIONTIME		MOIRAI$_S$		MOIRAI$_B$		MOIRAI$_L$		TimeLLM		GPT4TS		DLinear		PatchTST		TimesNet		Autoformer		Informer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.353**	**0.383**	0.375	0.402	0.384	0.402	0.380	0.398	0.448	0.460	0.458	0.456	0.492	0.495	0.516	0.485	0.861	0.628	0.613	0.552	1.179	0.792
	192	**0.392**	**0.410**	0.399	0.419	0.425	0.429	0.440	0.434	0.484	0.483	0.570	0.516	0.565	0.538	0.598	0.524	0.797	0.593	0.722	0.598	1.199	0.806
	336	**0.407**	**0.423**	0.412	0.429	0.456	0.450	0.514	0.474	0.589	0.540	0.608	0.535	0.721	0.622	0.657	0.550	0.941	0.648	0.750	0.619	1.202	0.811
	720	**0.406**	**0.441**	0.413	0.444	0.470	0.473	0.705	0.568	0.700	0.604	0.725	0.591	0.986	0.743	0.762	0.610	0.877	0.641	0.721	0.616	1.217	0.825
	avg	**0.390**	**0.414**	0.400	0.424	0.434	0.439	0.510	0.469	0.556	0.522	0.590	0.525	0.691	0.600	0.633	0.542	0.869	0.628	0.702	0.596	1.199	0.809
ETTh2	96	**0.271**	0.328	0.281	0.334	0.277	0.327	0.287	**0.325**	0.275	0.326	0.331	0.374	0.357	0.411	0.353	0.389	0.378	0.409	0.413	0.451	3.837	1.508
	192	**0.328**	**0.367**	0.340	0.373	0.340	0.374	0.347	**0.367**	0.374	0.373	0.402	0.411	0.569	0.519	0.403	0.414	0.490	0.467	0.474	0.477	3.856	1.513
	336	**0.345**	**0.381**	0.362	0.393	0.371	0.401	0.377	0.393	0.406	0.429	0.406	0.433	0.671	0.572	0.426	0.441	0.537	0.494	0.547	0.543	3.952	1.526
	720	**0.388**	0.422	0.380	0.416	0.394	0.426	0.404	**0.421**	0.427	0.449	0.449	0.464	0.824	0.648	0.477	0.480	0.510	0.491	0.516	0.523	3.842	1.503
	avg	**0.333**	**0.375**	0.341	0.379	0.346	0.382	0.354	0.377	0.370	0.394	0.397	0.421	0.605	0.538	0.415	0.431	0.479	0.465	0.488	0.499	3.872	1.513
ETTm1	96	**0.341**	**0.347**	0.404	0.383	0.335	0.360	0.353	0.363	0.346	0.388	0.390	0.404	0.352	0.392	0.410	0.419	0.583	0.501	0.774	0.614	1.162	0.785
	192	**0.360**	**0.360**	0.435	0.402	0.366	0.379	0.376	0.380	0.373	0.416	0.429	0.423	0.382	0.412	0.437	0.434	0.630	0.528	0.754	0.592	1.172	0.793
	336	**0.377**	**0.374**	0.462	0.416	0.391	0.394	0.399	0.395	0.413	0.426	0.469	0.439	0.419	0.434	0.476	0.454	0.725	0.568	0.869	0.677	1.227	0.908
	720	**0.416**	**0.405**	0.490	0.437	0.434	0.419	0.432	0.417	0.485	0.476	0.569	0.498	0.490	0.477	0.681	0.556	0.769	0.549	0.810	0.630	1.207	0.797
	avg	**0.374**	**0.372**	0.448	0.410	0.382	0.388	0.390	0.389	0.404	0.427	0.464	0.441	0.411	0.429	0.501	0.466	0.677	0.537	0.802	0.628	1.192	0.821
ETTm2	96	0.228	0.282	0.205	0.282	0.195	0.269	**0.189**	**0.260**	0.177	0.261	0.188	0.269	0.213	0.303	0.191	0.274	0.212	0.285	0.352	0.454	3.203	1.407
	192	0.262	0.305	0.261	0.318	0.247	0.303	**0.247**	**0.300**	0.241	0.314	0.251	0.309	0.278	0.345	0.252	0.317	0.270	0.323	0.694	0.691	3.112	1.387
	336	0.293	**0.328**	0.319	0.355	**0.291**	0.333	0.295	0.334	0.274	0.327	0.307	0.346	0.338	0.385	0.306	0.353	0.323	0.353	2.408	1.407	3.255	1.421
	720	**0.343**	**0.370**	0.415	0.410	0.355	0.377	0.372	0.386	0.417	0.390	0.426	0.417	0.436	0.440	0.433	0.427	0.474	0.449	1.913	1.166	3.909	1.543
	avg	0.282	0.321	0.300	0.341	**0.272**	0.321	0.276	**0.320**	0.277	0.323	0.293	0.335	0.316	0.368	0.296	0.343	0.320	0.353	1.342	0.930	3.370	1.440
Electricity	96	0.177	0.266	0.205	0.299	0.158	0.248	0.152	0.242	**0.139**	0.241	**0.139**	**0.237**	0.150	0.253	0.140	0.238	0.299	0.373	0.261	0.348	1.259	0.919
	192	0.188	0.277	0.220	0.310	0.174	0.263	0.171	0.259	**0.151**	**0.248**	0.156	0.252	0.164	0.264	0.160	0.255	0.305	0.379	0.338	0.406	1.160	0.873
	336	0.207	0.296	0.236	0.323	0.191	0.278	0.192	0.278	**0.169**	**0.270**	0.175	**0.270**	0.181	0.282	0.180	0.276	0.319	0.391	0.410	0.474	1.157	0.872
	720	0.256	0.337	0.270	0.347	0.229	**0.307**	0.236	0.313	0.240	0.322	**0.233**	0.317	0.223	0.321	0.241	0.323	0.369	0.426	0.715	0.685	1.203	0.898
	avg	0.207	0.294	0.233	0.320	0.188	0.274	0.188	0.273	**0.175**	0.270	0.176	**0.269**	0.180	0.280	0.180	0.273	0.323	0.392	0.431	0.478	1.195	0.891
Weather	96	0.220	0.257	0.173	0.212	0.167	**0.203**	0.177	0.208	**0.161**	0.210	0.163	0.215	0.171	0.224	0.165	0.215	0.184	0.230	0.221	0.297	0.374	0.401
	192	0.244	0.275	0.216	0.250	0.209	**0.241**	0.219	0.249	**0.204**	0.248	0.210	0.254	0.215	0.263	0.210	0.257	0.245	0.283	0.270	0.322	0.552	0.478
	336	0.280	0.299	0.260	0.282	**0.256**	**0.276**	0.277	0.292	0.261	0.302	**0.256**	0.292	0.258	0.299	0.259	0.297	0.305	0.321	0.320	0.351	0.724	0.541
	720	0.330	0.337	0.320	0.322	0.321	**0.323**	0.365	0.350	**0.309**	0.332	0.321	0.339	0.320	0.346	0.332	0.346	0.381	0.371	0.390	0.396	0.739	0.558
	avg	0.269	0.292	0.242	0.267	0.238	**0.261**	0.260	0.275	**0.234**	0.273	0.238	0.275	0.241	0.283	0.242	0.279	0.279	0.301	0.300	0.342	0.597	0.495
Average		**0.309**	0.345	0.327	0.357	0.310	**0.344**	0.329	0.350	0.336	0.368	0.360	0.378	0.407	0.416	0.378	0.389	0.491	0.446	0.678	0.579	1.904	0.995
1st count		**32**		0		10		8		10		6		0		0		0		0		0	

length of $L = 1000$. All datasets were tested with the hyperparameters $r = c = 0.4$ as we had done for the long-term TSF benchmark.

Results Table 15 presents VISIONTS 's MAE test results, with the normalized MAE calculated by dividing each dataset's MAE by the naive forecast's MAE and aggregated using the geometric mean across datasets. We include the result of each baseline from Woo et al. (2024). Particularly, we find that VISIONTS outperforms MOIRAI on some datasets with $P = 1$ (*e.g.*, FRED-MD and NN5 Weekly), showing that VISIONTS can still work effectively without significant periodicity.

B.7. Impact of backbones

Table 17 compares zero-shot forecasting performance of three MAE variants (112M, 330M, and 657M), showing that the three variants are similar, but larger models show a slight decrease. Particularly, the smallest model excels in ETTh2, ETTm1, ETTm2, and Weather, while the largest model excels in Electricity. Additionally, Table 16 compares VISIONTS with another visual backbone, LaMa.

B.8. Impact of the different image encoding strategies

Table 18 summarizes the impact of interpolation strategies and image orientations in the Alignment step. It shows that the smoother Bilinear and Bicubic interpolation perform similarly, both significantly better than the rougher Nearest Neighbor. This suggests that smooth resizing effectively handles time series interpolation. Moreover, image orientation has little impact on performance.

Table 10. MAE results of TimesFM and LLMTime for zero-shot forecasting, on the last test window of the original test split.

Method		VISIONTS	TimesFM	LLMTime
ETTh1	96	**0.35**	0.45	0.42
	192	**0.45**	0.53	0.50
ETTh2	96	**0.24**	0.35	0.33
	192	**0.60**	0.62	0.70
ETTm1	96	**0.12**	0.19	0.37
	192	**0.23**	0.26	0.71
ETTm2	96	**0.19**	0.24	0.29
	192	**0.24**	0.27	0.31
Average		**0.30**	0.36	0.45

Table 11. Comparison of traditional forecasting baselines in the zero-shot setting.

Method		VISIONTS		ETS		ARIMA		Seasonal Naïve		Seasonal Avg	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.353**	**0.383**	1.289	0.710	0.900	0.719	0.512	0.433	0.589	0.585
	192	**0.392**	**0.410**	1.319	0.730	0.906	0.724	0.581	0.469	0.598	0.590
	336	**0.407**	**0.423**	1.324	0.742	0.908	0.731	0.650	0.501	0.610	0.597
	720	**0.406**	**0.441**	1.329	0.751	0.932	0.753	0.655	0.514	0.656	0.624
	avg	**0.390**	**0.414**	1.315	0.733	0.912	0.732	0.600	0.479	0.613	0.599
ETTh2	96	**0.271**	**0.328**	0.399	0.408	0.488	0.508	0.391	0.380	0.457	0.494
	192	**0.328**	**0.367**	0.500	0.459	0.497	0.514	0.482	0.429	0.466	0.500
	336	**0.345**	**0.381**	0.562	0.498	0.507	0.522	0.532	0.466	0.476	0.509
	720	**0.388**	**0.422**	0.558	0.506	0.572	0.557	0.525	0.474	0.542	0.548
	avg	**0.333**	**0.375**	0.505	0.468	0.516	0.525	0.483	0.437	0.485	0.513
ETTm1	96	**0.341**	**0.347**	1.204	0.659	0.702	0.568	0.423	0.387	0.369	0.399
	192	**0.360**	**0.360**	1.251	0.685	0.704	0.570	0.463	0.406	0.374	0.402
	336	**0.377**	**0.374**	1.276	0.702	0.709	0.574	0.496	0.426	0.382	0.407
	720	0.416	**0.405**	1.311	0.724	0.713	0.580	0.574	0.464	**0.394**	0.416
	avg	**0.374**	**0.372**	1.261	0.693	0.707	0.573	0.489	0.421	0.380	0.406
ETTm2	96	**0.228**	**0.282**	0.257	0.324	0.397	0.434	0.263	0.301	0.365	0.411
	192	**0.262**	**0.305**	0.331	0.366	0.402	0.436	0.321	0.337	0.369	0.414
	336	**0.293**	**0.328**	0.402	0.406	0.407	0.439	0.376	0.370	0.375	0.418
	720	**0.343**	**0.370**	0.512	0.462	0.413	0.443	0.471	0.422	0.380	0.423
	avg	**0.282**	**0.321**	0.376	0.390	0.405	0.438	0.358	0.357	0.372	0.417
Average		**0.344**	**0.370**	0.864	0.571	0.635	0.567	0.482	0.424	0.463	0.484
1st count		**41**		0		0		0		1	

B.9. Hyperparameter analysis

Figs. 8 to 10 show the influence of three hyperparameters, r, c, and L. We report the MSE averaged on four prediction lengths {96, 192, 336, 720}.

Table 12. Comparison of traditional zero-shot forecasting baselines.

Method		VISIONTS		ETS		ARIMA		Seasonal Naïve		Seasonal Avg	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$ETTh1$	96	**0.353**	**0.383**	1.289	0.710	0.900	0.719	0.512	0.433	0.589	0.585
	192	**0.392**	**0.410**	1.319	0.730	0.906	0.724	0.581	0.469	0.598	0.590
	336	**0.407**	**0.423**	1.324	0.742	0.908	0.731	0.650	0.501	0.610	0.597
	720	**0.406**	**0.441**	1.329	0.751	0.932	0.753	0.655	0.514	0.656	0.624
	avg	**0.390**	**0.414**	1.315	0.733	0.912	0.732	0.600	0.479	0.613	0.599
$ETTh2$	96	**0.271**	**0.328**	0.399	0.408	0.488	0.508	0.391	0.380	0.457	0.494
	192	**0.328**	**0.367**	0.500	0.459	0.497	0.514	0.482	0.429	0.466	0.500
	336	**0.345**	**0.381**	0.562	0.498	0.507	0.522	0.532	0.466	0.476	0.509
	720	**0.388**	**0.422**	0.558	0.506	0.572	0.557	0.525	0.474	0.542	0.548
	avg	**0.333**	**0.375**	0.505	0.468	0.516	0.525	0.483	0.437	0.485	0.513
$ETTm1$	96	**0.341**	**0.347**	1.204	0.659	0.702	0.568	0.423	0.387	0.369	0.399
	192	**0.360**	**0.360**	1.251	0.685	0.704	0.570	0.463	0.406	0.374	0.402
	336	**0.377**	**0.374**	1.276	0.702	0.709	0.574	0.496	0.426	0.382	0.407
	720	0.416	**0.405**	1.311	0.724	0.713	0.580	0.574	0.464	**0.394**	0.416
	avg	**0.374**	**0.372**	1.261	0.693	0.707	0.573	0.489	0.421	0.380	0.406
$ETTm2$	96	**0.228**	**0.282**	0.257	0.324	0.397	0.434	0.263	0.301	0.365	0.411
	192	**0.262**	**0.305**	0.331	0.366	0.402	0.436	0.321	0.337	0.369	0.414
	336	**0.293**	**0.328**	0.402	0.406	0.407	0.439	0.376	0.370	0.375	0.418
	720	**0.343**	**0.370**	0.512	0.462	0.413	0.443	0.471	0.422	0.380	0.423
	avg	**0.282**	**0.321**	0.376	0.390	0.405	0.438	0.358	0.357	0.372	0.417
Average		**0.344**	**0.370**	0.864	0.571	0.635	0.567	0.482	0.424	0.463	0.484
1st count		**41**		0		0		0		1	

Table 13. Comparison of Time-MoE and TTM in the **zero-shot** setting. We report the base and large model results for Time-MoE, as the ultra model weights are not yet released. For TTM, we used the official HuggingFace model for replication. The following table summarizes the performance of various zero-shot foundation models.

		VISIONTS	Time-MoE (base)	Time-MoE (large)	TTM (v1)
ETTh1	MSE	**0.390**	0.400	0.394	0.398
	MAE	**0.414**	0.424	0.419	0.421
ETTh2	MSE	**0.333**	0.366	0.405	0.348
	MAE	**0.375**	0.404	0.415	0.393
ETTm1	MSE	**0.374**	0.394	0.376	0.520
	MAE	**0.372**	0.415	0.405	0.479
ETTm2	MSE	**0.282**	0.317	0.316	0.312
	MAE	**0.321**	0.365	0.361	0.348
Electricity	MSE	0.207	(data leakage)	(data leakage)	**0.201**
	MAE	0.294	(data leakage)	(data leakage)	**0.293**
Weather	MSE	0.269	0.265	0.270	**0.234**
	MAE	0.292	0.297	0.300	**0.266**
Average	MSE	**0.309**	-	-	0.335
	MAE	**0.345**	-	-	0.367
1st count		**10**	0	0	4

Table 14. Comparison of CALF in both zero-shot and full-shot settings.

		VISIONTS (zero-shot)	VISIONTS (full-shot)	CALF
ETTh1	MSE	**0.390**	0.395	0.432
	MAE	0.414	**0.409**	0.428
ETTh2	MSE	**0.333**	0.336	0.349
	MAE	**0.375**	0.382	0.382
ETTm1	MSE	0.374	**0.338**	0.395
	MAE	0.372	**0.367**	0.390
ETTm2	MSE	0.282	**0.261**	0.281
	MAE	0.321	**0.319**	0.321
Electricity	MSE	0.207	**0.156**	0.175
	MAE	0.294	**0.249**	0.265
Weather	MSE	0.269	**0.227**	0.250
	MAE	0.292	**0.262**	0.274
Average	MSE	0.309	**0.286**	0.314
	MAE	0.345	**0.331**	0.343

Table 15. Full results of Fig. 5: Forecasting results (MAE) on the Monash TSF benchmark. We reported the reproduction results of LLMTime based on the GPT3.5 API from Woo et al. (2024).

	VISIONTS	LLMTime	MOIRAI$_{\text{Small}}$	Naive	SES	Theta	TBATS	ETS	(DHR-)ARIMA	PR	CatBoost	FFNN	DeepAR	N-BEATS	WaveNet	Transformer
M1 Monthly	1987.69	2562.84	2082.26	2707.75	2259.04	2166.18	2237.5	1905.28	2080.13	2088.25	2052.32	2162.58	1860.81	**1820.37**	2184.42	2723.88
M3 Monthly	737.93	877.97	713.41	837.14	743.41	**623.71**	630.59	626.46	654.8	692.97	732	692.48	728.81	648.6	699.3	798.38
M3 Other	315.85	300.3	263.54	278.43	277.83	215.35	**189.42**	194.98	193.02	234.43	318.13	240.17	247.56	221.85	245.29	239.24
M4 Monthly	666.54	728.27	597.6	671.27	625.24	**563.58**	589.52	582.6	575.36	596.19	611.69	612.52	615.22	578.48	655.51	780.47
M4 Weekly	404.23	518.44	339.76	347.99	336.82	333.32	296.15	335.66	321.61	293.21	364.65	338.37	351.78	**277.73**	359.46	378.89
M4 Daily	215.63	266.52	189.1	180.83	178.27	178.86	**176.6**	193.26	179.67	181.92	231.36	177.91	299.79	190.44	189.47	201.08
M4 Hourly	288.37	576.06	268.04	1218.06	1218.06	1220.97	386.27	3358.1	1310.85	**257.39**	285.35	385.49	886.02	425.75	393.63	320.54
Tourism Quarterly	12931.88	16918.86	18352.44	15845.1	15014.19	**7656.49**	9972.42	8925.52	10475.47	9092.58	10267.97	8981.04	9511.37	8640.56	9137.12	9521.67
Tourism Monthly	2560.19	5608.61	3569.85	5636.83	5302.1	2069.96	2940.08	2004.51	2536.77	2187.28	2537.04	2022.21	**1871.69**	2003.02	2095.13	2146.98
CIF 2016	570907.24	599313.8	655888.58	578596.5	581875.97	714818.6	855578.4	642421.4	**469059**	563205.57	603551.3	1495923	3200418	679034.8	5998225	4057973
Aus. Elec. Demand	237.44	760.81	266.57	659.6	659.6	665.04	370.74	1282.99	1045.92	247.18	241.77	258.76	302.41	**213.83**	227.5	231.45
Bitcoin	2.33E+18	1.74E+18	1.76E+18	7.78E+17	5.33E+18	5.33E+18	9.9E+17	1.1E+18	3.62E+18	**6.66E+17**	1.93E+18	1.45E+18	1.95E+18	1.06E+18	2.46E+18	2.61E+18
Pedestrian Counts	52.01	97.77	54.88	170.88	170.87	170.94	222.38	216.5	635.16	44.18	**43.41**	46.41	44.78	66.84	46.46	47.29
Vehicle Trips	22.08	31.48	24.46	31.42	29.98	30.76	**21.21**	30.95	30.07	27.24	22.61	22.93	22	28.16	24.15	28.01
KDD cup	38.16	42.72	39.81	42.13	42.04	42.06	39.2	44.88	52.2	36.85	**34.82**	37.16	48.98	49.1	37.08	44.46
Weather	2.06	2.17	**1.96**	2.36	2.24	2.51	2.3	2.35	2.45	8.17	2.51	2.09	2.02	2.34	2.29	2.03
NN5 Daily	**3.51**	7.1	5.37	8.26	6.63	3.8	3.7	3.72	4.41	5.47	4.22	4.06	3.94	4.92	3.97	4.16
NN5 Weekly	14.67	15.76	15.07	16.71	15.66	15.3	14.98	15.7	15.38	14.94	15.29	15.02	14.69	**14.19**	19.34	20.34
Carparts	0.58	0.44	0.53	0.65	0.55	0.53	0.58	0.56	0.56	0.41	0.53	**0.39**	**0.39**	0.98	0.4	**0.39**
FRED-MD	**1893.67**	2804.64	2568.48	2825.67	2798.22	3492.84	1989.97	2041.42	2957.11	8921.94	2475.68	2339.57	4264.36	2557.8	2508.4	4666.04
Traffic Hourly	**0.01**	0.03	0.02	0.03	0.03	0.03	0.04	0.03	0.04	0.02	0.02	**0.01**	**0.01**	0.02	0.02	**0.01**
Traffic Weekly	1.14	1.15	1.17	1.19	1.12	1.13	1.17	1.14	1.22	1.13	1.17	1.15	1.18	**1.11**	1.2	1.42
Rideshare	5.92	6.28	**1.35**	6.29	6.29	7.62	6.45	6.29	3.37	6.3	6.07	6.59	6.28	5.55	2.75	6.29
Hospital	19.36	25.68	23	24.07	21.76	18.54	**17.43**	17.97	19.6	19.24	19.17	22.86	18.25	20.18	19.35	36.19
COVID Deaths	137.51	653.31	124.32	353.71	353.71	321.32	96.29	**85.59**	85.77	347.98	475.15	144.14	201.98	158.81	1049.48	408.66
Temperature Rain	6.37	6.37	5.3	9.39	8.18	8.22	7.14	8.21	7.19	6.13	6.76	5.56	5.37	7.28	5.81	**5.24**
Sunspot	2.81	5.07	**0.11**	3.93	4.93	4.93	2.57	4.93	2.57	3.83	2.27	7.97	0.77	14.47	0.17	0.13
Saugeen River Flow	30.22	34.84	24.07	21.5	21.5	21.49	22.26	30.69	22.38	25.24	**21.28**	22.98	23.51	27.92	22.17	28.06
US Births	519.94	1374.99	872.51	1152.67	1192.2	586.93	**399**	419.73	526.33	574.93	441.7	557.87	424.93	422	504.4	452.87
Normalized MAE	0.729	1.041	0.657	1.000	1.028	0.927	0.758	0.872	0.898	0.785	0.760	0.741	0.759	0.783	0.749	0.770
Rank	2	16	1	14	15	13	5	11	12	10	7	3	6	9	4	8

Table 16. Comparison of LaMa as the backbone. Results are averaged on four prediction lengths.

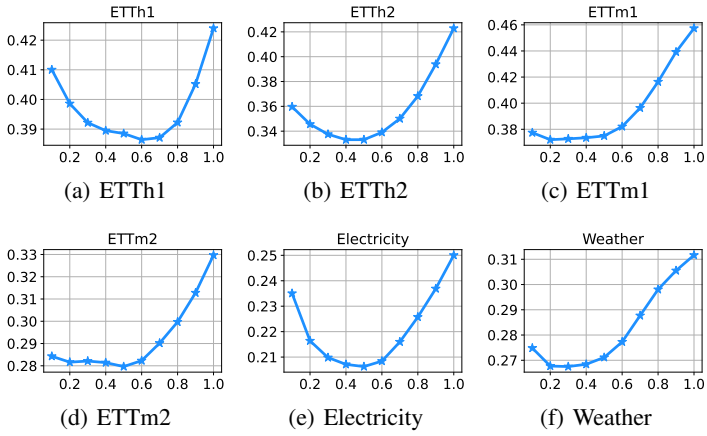
	MAE		LaMa		MOIRAI$_{\text{Small}}$		MOIRAI$_{\text{Large}}$	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.390	0.414	0.425	0.433	0.400	0.424	0.510	0.469
ETTh2	0.333	0.375	0.376	0.408	0.341	0.379	0.354	0.377
ETTm1	0.374	0.372	0.400	0.391	0.448	0.410	0.390	0.389
ETTm2	0.282	0.321	0.294	0.337	0.300	0.341	0.276	0.320
Average	0.344	0.370	0.374	0.392	0.372	0.388	0.382	0.388

Table 17. Full results of Table 2: zero-shot forecasting results of different MAE variants. **Bold**: best results among three variants. We also include the results from MOIRAI for reference.

Method		MAE (Base) 112M		MAE (Large) 330M		MAE (Huge) 657M		MOIRAI (Small) 14M		MOIRAI (Base) 91M		MOIRAI (Huge) 311M	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.353	0.383	**0.346**	**0.382**	0.362	0.384	*0.375*	*0.402*	*0.384*	*0.402*	*0.380*	*0.398*
	192	0.392	0.410	**0.379**	**0.406**	0.407	0.414	*0.399*	*0.419*	*0.425*	*0.429*	*0.440*	*0.434*
	336	0.407	0.423	**0.391**	**0.416**	0.399	0.419	*0.412*	*0.429*	*0.456*	*0.450*	*0.514*	*0.474*
	720	0.406	0.441	0.397	**0.433**	**0.395**	**0.433**	*0.413*	*0.444*	*0.470*	*0.473*	*0.705*	*0.568*
	avg	0.390	0.414	**0.378**	**0.409**	0.391	0.412	*0.400*	*0.424*	*0.434*	*0.439*	*0.510*	*0.469*
ETTh2	96	**0.271**	**0.328**	0.286	0.334	0.285	0.333	*0.281*	*0.334*	*0.277*	*0.327*	*0.287*	*0.325*
	192	**0.328**	**0.367**	0.346	0.375	0.337	0.369	*0.340*	*0.373*	*0.340*	*0.374*	*0.347*	*0.367*
	336	**0.345**	**0.381**	0.356	0.387	0.357	0.388	*0.362*	*0.393*	*0.371*	*0.401*	*0.377*	*0.393*
	720	0.388	0.422	**0.371**	**0.409**	0.379	0.412	*0.380*	*0.416*	*0.394*	*0.426*	*0.404*	*0.421*
	avg	**0.333**	**0.375**	0.340	0.377	0.339	**0.375**	*0.341*	*0.379*	*0.346*	*0.382*	*0.354*	*0.377*
ETTm1	96	**0.341**	**0.347**	0.344	0.349	0.352	0.351	*0.404*	*0.383*	*0.335*	*0.360*	*0.353*	*0.363*
	192	**0.360**	**0.360**	0.365	0.363	**0.360**	0.367	*0.435*	*0.402*	*0.366*	*0.379*	*0.376*	*0.380*
	336	**0.377**	**0.374**	0.381	0.376	0.381	0.383	*0.462*	*0.416*	*0.391*	*0.394*	*0.399*	*0.395*
	720	**0.416**	**0.405**	0.429	0.411	0.440	0.412	*0.490*	*0.437*	*0.434*	*0.419*	*0.432*	*0.417*
	avg	**0.374**	**0.372**	0.379	0.375	0.383	0.378	*0.448*	*0.410*	*0.382*	*0.388*	*0.390*	*0.389*
ETTm2	96	0.228	**0.282**	**0.225**	**0.282**	0.229	**0.282**	*0.205*	*0.282*	*0.195*	*0.269*	*0.189*	*0.260*
	192	**0.262**	**0.305**	**0.262**	**0.305**	0.265	0.306	*0.261*	*0.318*	*0.247*	*0.303*	*0.247*	*0.300*
	336	0.293	0.328	0.299	0.331	**0.286**	**0.324**	*0.319*	*0.355*	*0.291*	*0.333*	*0.295*	*0.334*
	720	**0.343**	**0.370**	0.358	0.377	0.355	0.374	*0.415*	*0.410*	*0.355*	*0.377*	*0.372*	*0.386*
	avg	**0.282**	**0.321**	0.286	0.324	0.284	0.322	*0.300*	*0.341*	*0.272*	*0.321*	*0.276*	*0.320*
Electricity	96	0.177	0.266	0.177	0.268	**0.170**	**0.259**	*0.205*	*0.299*	*0.158*	*0.248*	*0.152*	*0.242*
	192	0.188	0.277	0.192	0.283	**0.182**	**0.273**	*0.220*	*0.310*	*0.174*	*0.263*	*0.171*	*0.259*
	336	**0.207**	0.296	0.213	0.303	**0.207**	**0.295**	*0.236*	*0.323*	*0.191*	*0.278*	*0.192*	*0.278*
	720	0.256	0.337	0.256	0.337	**0.250**	**0.333**	*0.270*	*0.347*	*0.229*	*0.307*	*0.236*	*0.313*
	avg	0.207	0.294	0.209	0.298	**0.202**	**0.290**	*0.233*	*0.320*	*0.188*	*0.274*	*0.188*	*0.273*
Weather	96	**0.220**	**0.257**	0.222	**0.257**	0.235	0.265	*0.173*	*0.212*	*0.167*	*0.203*	*0.177*	*0.208*
	192	**0.244**	**0.275**	0.246	**0.275**	0.276	0.288	*0.216*	*0.250*	*0.209*	*0.241*	*0.219*	*0.249*
	336	**0.280**	**0.299**	0.283	0.301	0.304	0.309	*0.260*	*0.282*	*0.256*	*0.276*	*0.277*	*0.292*
	720	**0.330**	**0.337**	0.338	0.343	0.351	0.350	*0.320*	*0.322*	*0.321*	*0.323*	*0.365*	*0.350*
	avg	**0.269**	**0.292**	0.272	0.294	0.292	0.303	*0.242*	*0.267*	*0.238*	*0.261*	*0.260*	*0.275*
Average		**0.309**	**0.345**	0.311	0.346	0.315	0.347	*0.327*	*0.357*	*0.310*	*0.344*	*0.329*	*0.350*
1st count		**38**		17		17		-		-		-	

Table 18. Impact of resampling filters and image orientations.

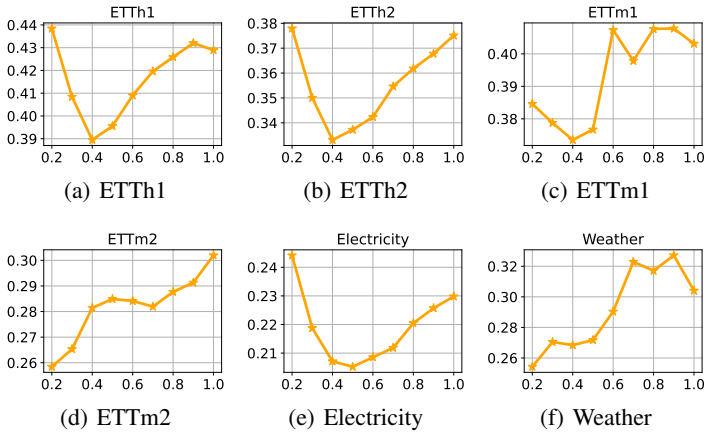
		Interpolation strategies in resampling								*Image orientation*					
Method		**Bilinear**		**Bicubic**		**Nearest Neighbor**		**Method**		**-**		**Horizontal flip**		**Vertical flip**	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	**Metric**		MSE	MAE	MSE	MAE	MSE	MAE
$ETTh1$	96	0.353	**0.383**	**0.351**	**0.383**	0.426	0.424	$ETTh1$	96	0.353	0.383	**0.348**	**0.379**	0.355	0.385
	192	**0.392**	0.410	**0.392**	**0.409**	0.450	0.443		192	0.392	0.410	**0.386**	**0.404**	0.394	0.411
	336	**0.407**	0.423	**0.407**	**0.422**	0.451	0.450		336	0.407	0.423	**0.401**	**0.416**	0.408	0.423
	720	0.406	0.441	**0.405**	**0.440**	0.454	0.470		720	0.406	0.441	**0.399**	**0.430**	0.406	0.442
	avg	0.390	**0.414**	**0.389**	**0.414**	0.445	0.446		avg	0.390	0.414	**0.384**	**0.407**	0.391	0.415
$ETTh2$	96	**0.271**	**0.328**	0.274	0.329	0.298	0.349	$ETTh2$	96	**0.271**	**0.328**	0.274	0.329	0.274	0.330
	192	**0.328**	**0.367**	0.330	**0.367**	0.343	0.380		192	**0.328**	**0.367**	0.331	0.370	0.330	**0.367**
	336	**0.345**	0.381	**0.345**	**0.380**	0.373	0.401		336	**0.345**	**0.381**	0.347	0.386	**0.345**	**0.381**
	720	0.388	0.422	**0.386**	**0.419**	0.404	0.431		720	0.388	0.422	**0.376**	**0.416**	0.388	0.422
	avg	**0.333**	0.375	0.334	**0.374**	0.354	0.390		avg	0.333	**0.375**	**0.332**	**0.375**	0.334	**0.375**
$ETTm1$	96	**0.341**	**0.347**	0.366	0.354	0.399	0.374	$ETTm1$	96	**0.341**	**0.347**	0.345	0.348	0.342	**0.347**
	192	**0.360**	**0.360**	0.383	0.367	0.397	0.376		192	**0.360**	**0.360**	0.364	0.362	**0.360**	**0.360**
	336	**0.377**	**0.374**	0.396	0.381	0.386	0.380		336	**0.377**	**0.374**	0.378	0.375	**0.377**	**0.374**
	720	**0.416**	**0.405**	0.429	0.409	0.417	0.409		720	**0.416**	**0.405**	0.419	0.408	0.417	**0.405**
	avg	**0.374**	**0.372**	0.393	0.378	0.400	0.384		avg	**0.374**	**0.372**	0.376	0.373	**0.374**	**0.372**
$ETTm2$	96	**0.228**	**0.282**	0.246	0.296	0.264	0.326	$ETTm2$	96	**0.228**	**0.282**	0.230	0.286	**0.228**	0.283
	192	**0.262**	**0.305**	0.273	0.313	0.273	0.328		192	**0.262**	**0.305**	0.264	0.308	**0.262**	**0.305**
	336	**0.293**	**0.328**	0.303	0.334	0.297	0.343		336	**0.293**	**0.328**	0.298	0.332	**0.293**	**0.328**
	720	0.343	0.370	0.343	0.370	**0.334**	**0.369**		720	**0.343**	0.370	0.350	0.373	**0.343**	**0.369**
	avg	**0.282**	**0.321**	0.291	0.328	0.292	0.341		avg	**0.282**	**0.321**	0.285	0.325	**0.282**	**0.321**
Average		**0.344**	**0.370**	0.352	0.373	0.373	0.391	**Average**		**0.344**	**0.370**	**0.344**	**0.370**	0.345	0.371
1st count		**30**		18		2		**1st count**		**28**		16		21	



(a) ETTh1 (b) ETTh2 (c) ETTm1

(d) ETTm2 (e) Electricity (f) Weather

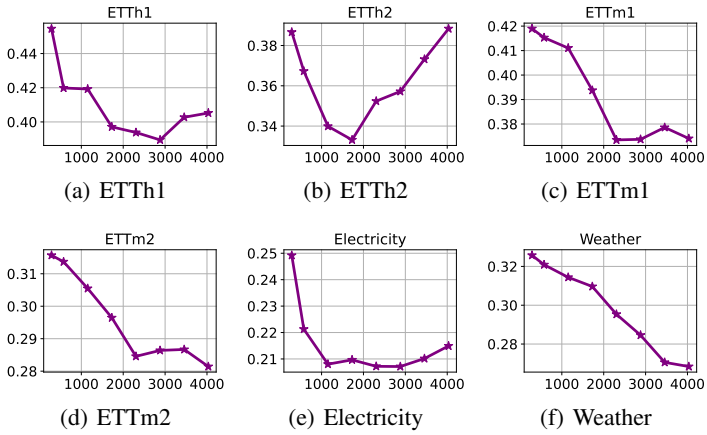
Figure 8. MSE (Y-axis) performance of different normalization constants r (X-axis).



(a) ETTh1 (b) ETTh2 (c) ETTm1

(d) ETTm2 (e) Electricity (f) Weather

Figure 9. MSE (Y-axis) performance of different alignment constants c (X-axis).



(a) ETTh1 (b) ETTh2 (c) ETTm1

(d) ETTm2 (e) Electricity (f) Weather

Figure 10. MSE (Y-axis) performance of different context lengths L (X-axis).

C. Full-Shot Forecasting

C.1. Training details

Table 19. Final hyperparameters for VISIONTS used in our full-shot forecasting.

	ETTh1	ETTh2	ETTm1	ETTm2	Illness	Weather	Traffic	Electricity
Normalization constant r	0.4	0.4	0.4	0.4	1.0	1.0	0.4	0.4
Alignment constant c	0.4	0.4	0.4	0.4	0.4	0.7	0.4	0.4
Context length L	1152	1152	2304	1152	104	576	1152	1152

Based on the principle of channel independence (Nie et al., 2022; Han et al., 2024), we treat the variables of each time series as individual data samples. We use an Adam optimizer with a learning rate 0.0001 and a batch size 256 to fine-tune `MAE`. All experiments are repeated three times. The training epoch is one for all the datasets except Illness, for which we train `MAE` for 100 epochs with an early stop due to the limited training dataset scale. We conduct tuning on validation sets for the three hyperparameters, r, c, and L. The final hyperparameters used are summarized in Table 19.

C.2. Full results and standard deviations

Table 20. Standard deviations of full-shot experiments.

Method		VISIONTS		Time-LLM		GPT4TS	
Metric		MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.347 ± 0.002**	**0.376 ± 0.000**	0.376 ± 0.003	0.402 ± 0.002	0.370 ± 0.003	0.389 ± 0.001
	192	**0.385 ± 0.001**	**0.400 ± 0.000**	0.407 ± 0.003	0.421 ± 0.002	0.412 ± 0.003	0.413 ± 0.001
	336	**0.407 ± 0.001**	**0.415 ± 0.001**	0.430 ± 0.004	0.438 ± 0.001	0.448 ± 0.003	0.431 ± 0.001
	720	**0.439 ± 0.001**	**0.443 ± 0.000**	0.457 ± 0.003	0.468 ± 0.001	0.441 ± 0.003	0.449 ± 0.001
ETTh2	96	**0.269 ± 0.003**	**0.328 ± 0.002**	0.286 ± 0.003	0.346 ± 0.002	0.280 ± 0.001	0.335 ± 0.001
	192	**0.332 ± 0.001**	**0.374 ± 0.001**	0.361 ± 0.003	0.391 ± 0.002	0.348 ± 0.002	0.380 ± 0.001
	336	**0.351 ± 0.002**	**0.395 ± 0.002**	0.390 ± 0.003	0.414 ± 0.002	0.380 ± 0.002	0.405 ± 0.001
	720	**0.390 ± 0.003**	**0.430 ± 0.002**	0.405 ± 0.003	0.434 ± 0.002	0.406 ± 0.002	0.436 ± 0.001
ETTm1	96	**0.281 ± 0.001**	**0.322 ± 0.001**	0.291 ± 0.001	0.341 ± 0.001	0.300 ± 0.001	0.340 ± 0.000
	192	**0.322 ± 0.006**	**0.353 ± 0.002**	0.341 ± 0.001	0.369 ± 0.001	0.343 ± 0.001	0.368 ± 0.000
	336	**0.356 ± 0.003**	**0.379 ± 0.002**	0.359 ± 0.002	0.379 ± 0.001	0.376 ± 0.001	0.386 ± 0.000
	720	**0.391 ± 0.001**	**0.413 ± 0.001**	0.433 ± 0.001	0.419 ± 0.001	0.431 ± 0.001	0.416 ± 0.000
ETTm2	96	0.169 ± 0.003	0.256 ± 0.002	**0.162 ± 0.001**	**0.248 ± 0.001**	0.163 ± 0.001	0.249 ± 0.001
	192	0.225 ± 0.003	0.294 ± 0.003	0.235 ± 0.002	0.304 ± 0.001	**0.222 ± 0.001**	**0.291 ± 0.000**
	336	0.278 ± 0.002	0.334 ± 0.001	0.280 ± 0.002	0.329 ± 0.001	**0.273 ± 0.001**	**0.327 ± 0.001**
	720	0.372 ± 0.002	0.392 ± 0.002	0.366 ± 0.002	0.382 ± 0.001	**0.357 ± 0.001**	**0.376 ± 0.001**
Weather	96	**0.142 ± 0.000**	0.192 ± 0.001	0.155 ± 0.001	0.199 ± 0.001	0.148 ± 0.001	**0.188 ± 0.000**
	192	**0.191 ± 0.000**	0.238 ± 0.000	0.223 ± 0.001	0.261 ± 0.001	0.192 ± 0.001	**0.230 ± 0.000**
	336	**0.246 ± 0.003**	0.282 ± 0.001	0.251 ± 0.001	0.279 ± 0.001	**0.246 ± 0.001**	**0.273 ± 0.000**
	720	0.328 ± 0.004	0.337 ± 0.001	0.345 ± 0.001	0.342 ± 0.001	**0.320 ± 0.001**	**0.328 ± 0.000**
Traffic	96	**0.344 ± 0.001**	**0.236 ± 0.000**	0.392 ± 0.001	0.267 ± 0.000	0.396 ± 0.001	0.264 ± 0.000
	192	**0.372 ± 0.001**	**0.249 ± 0.001**	0.409 ± 0.001	0.271 ± 0.000	0.412 ± 0.001	0.268 ± 0.000
	336	**0.383 ± 0.001**	**0.257 ± 0.001**	0.434 ± 0.001	0.296 ± 0.000	0.421 ± 0.001	0.273 ± 0.000
	720	**0.422 ± 0.001**	**0.280 ± 0.000**	0.451 ± 0.001	0.291 ± 0.000	0.455 ± 0.001	0.291 ± 0.000
Electricity	96	**0.126 ± 0.000**	**0.218 ± 0.000**	0.137 ± 0.000	0.233 ± 0.000	0.141 ± 0.000	0.239 ± 0.000
	192	**0.146 ± 0.001**	**0.239 ± 0.001**	0.152 ± 0.000	0.247 ± 0.000	0.158 ± 0.000	0.253 ± 0.000
	336	**0.161 ± 0.001**	**0.255 ± 0.001**	0.169 ± 0.000	0.267 ± 0.000	0.172 ± 0.000	0.266 ± 0.000
	720	**0.193 ± 0.000**	**0.286 ± 0.000**	0.200 ± 0.000	0.290 ± 0.000	0.207 ± 0.000	0.293 ± 0.000
1st count		**42**		2		12	

Table 21 shows the full results of the full-shot experiments. We also report the standard deviations of our full-shot experiments computed on three runs in Table 20, including the results of Time-LLM and GPT4TS from Tan et al. (2024) for reference.

C.3. Ablation study and fine-tuning strategy comparison

We compare the following ablation variants to verify the role of the visual model (VM), similar to Tan et al. (2024).

- **w/o VM** removes all the transformer blocks in encoders and decoders.

Table 21. Full results of Table 4: Full-shot forecasting performance on the long-term TSF benchmark. VISIONTS is fine-tuned only a single epoch on each dataset except for Illness.

Pretrain		*Images*		*Text*				*No Pretrain*															
Method		VISIONTS		Time-LLM		GPT4TS		DLinear		PatchTST		TimesNet		FEDformer		Autoformer		Stationary		ETSformer		Informer	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.347**	**0.376**	0.376	0.402	0.370	0.389	0.375	0.399	0.370	0.399	0.384	0.402	0.376	0.419	0.449	0.459	0.513	0.491	0.494	0.479	0.865	0.713
	192	**0.385**	**0.400**	0.407	0.421	0.412	0.413	0.405	0.416	0.413	0.421	0.436	0.429	0.420	0.448	0.500	0.482	0.534	0.504	0.538	0.504	1.008	0.792
	336	**0.407**	**0.415**	0.430	0.438	0.448	0.431	0.439	0.443	0.422	0.436	0.491	0.469	0.459	0.465	0.521	0.496	0.588	0.535	0.574	0.521	1.107	0.809
	720	**0.439**	**0.443**	0.457	0.468	0.441	0.449	0.472	0.490	0.447	0.466	0.521	0.500	0.506	0.507	0.514	0.512	0.643	0.616	0.562	0.535	1.181	0.865
	avg	**0.395**	**0.409**	0.418	0.432	0.418	0.421	0.423	0.437	0.413	0.431	0.458	0.450	0.440	0.460	0.496	0.487	0.570	0.537	0.542	0.510	1.040	0.795
ETTh2	96	**0.269**	**0.328**	0.286	0.346	0.280	0.335	0.289	0.353	0.274	0.336	0.340	0.374	0.358	0.397	0.346	0.388	0.476	0.458	0.340	0.391	3.755	1.525
	192	**0.332**	**0.374**	0.361	0.391	0.348	0.380	0.383	0.418	0.339	0.379	0.402	0.414	0.429	0.439	0.456	0.452	0.512	0.493	0.430	0.439	5.602	1.931
	336	0.351	0.395	0.390	0.414	0.380	0.405	0.448	0.465	**0.329**	**0.380**	0.452	0.452	0.496	0.487	0.482	0.486	0.552	0.551	0.485	0.479	4.721	1.835
	720	0.390	0.430	0.405	0.434	0.406	0.436	0.605	0.551	**0.379**	**0.422**	0.462	0.468	0.463	0.474	0.515	0.511	0.562	0.560	0.500	0.497	3.647	1.625
	avg	0.336	0.382	0.361	0.396	0.354	0.389	0.431	0.447	**0.330**	**0.379**	0.414	0.427	0.437	0.449	0.450	0.459	0.526	0.516	0.439	0.452	4.431	1.729
ETTm1	96	**0.281**	**0.322**	0.291	0.341	0.300	0.340	0.299	0.343	0.290	0.342	0.338	0.375	0.379	0.419	0.505	0.475	0.386	0.398	0.375	0.398	0.672	0.571
	192	**0.322**	**0.353**	0.341	0.369	0.343	0.368	0.335	0.365	0.332	0.369	0.374	0.387	0.426	0.441	0.553	0.496	0.459	0.444	0.408	0.410	0.795	0.669
	336	**0.356**	**0.379**	0.359	**0.379**	0.376	0.386	0.369	0.386	0.366	0.392	0.410	0.411	0.445	0.459	0.621	0.537	0.495	0.464	0.435	0.428	1.212	0.871
	720	**0.391**	**0.413**	0.433	0.419	0.431	0.416	0.425	0.421	0.416	0.420	0.478	0.450	0.543	0.490	0.671	0.561	0.585	0.516	0.499	0.462	1.166	0.823
	avg	**0.338**	**0.367**	0.356	0.377	0.363	0.378	0.357	0.379	0.351	0.381	0.400	0.406	0.448	0.452	0.588	0.517	0.481	0.456	0.429	0.425	0.961	0.734
ETTm2	96	0.169	0.256	**0.162**	**0.248**	0.163	0.249	0.167	0.269	0.165	0.255	0.187	0.267	0.203	0.287	0.255	0.339	0.192	0.274	0.189	0.280	0.365	0.453
	192	0.225	0.294	0.235	0.304	0.222	**0.291**	0.224	0.303	**0.220**	0.292	0.249	0.309	0.269	0.328	0.281	0.340	0.280	0.339	0.253	0.319	0.533	0.563
	336	0.278	0.334	0.280	0.329	**0.273**	**0.327**	0.281	0.342	0.274	0.329	0.321	0.351	0.325	0.366	0.339	0.372	0.334	0.361	0.314	0.357	1.363	0.887
	720	0.372	0.392	0.366	0.382	**0.357**	**0.376**	0.397	0.421	0.362	0.385	0.408	0.403	0.421	0.415	0.433	0.432	0.417	0.413	0.414	0.413	3.379	1.338
	avg	0.261	0.319	0.261	0.316	**0.254**	**0.311**	0.267	0.334	0.255	0.315	0.291	0.333	0.305	0.349	0.327	0.371	0.306	0.347	0.293	0.342	1.410	0.810
Illness	24	2.034	0.937	1.792	0.807	1.869	0.823	2.215	1.081	**1.319**	**0.754**	2.317	0.934	3.228	1.260	3.483	1.287	2.294	0.945	2.527	1.020	5.764	1.677
	36	1.866	0.888	1.833	**0.833**	1.853	0.854	1.963	0.963	**1.430**	0.834	1.972	0.920	2.679	1.080	3.103	1.148	1.825	0.848	2.615	1.007	4.755	1.467
	48	1.784	0.870	2.269	1.012	1.886	0.855	2.130	1.024	**1.553**	**0.815**	2.238	0.940	2.622	1.078	2.669	1.085	2.010	0.900	2.359	0.972	4.763	1.469
	60	1.910	0.912	2.177	0.925	1.877	0.877	2.368	1.096	**1.470**	**0.788**	2.027	0.928	2.857	1.157	2.770	1.125	2.178	0.963	2.487	1.016	5.264	1.564
	avg	1.899	0.902	2.018	0.894	1.871	0.852	2.169	1.041	**1.443**	**0.798**	2.139	0.931	2.847	1.144	3.006	1.161	2.077	0.914	2.497	1.004	5.137	1.544
Weather	96	**0.142**	0.192	0.155	0.199	0.148	**0.188**	0.176	0.237	0.149	0.198	0.172	0.220	0.217	0.296	0.266	0.336	0.173	0.223	0.197	0.281	0.300	0.384
	192	**0.191**	0.238	0.223	0.261	0.192	**0.230**	0.220	0.282	0.194	0.241	0.219	0.261	0.276	0.336	0.307	0.367	0.245	0.285	0.237	0.312	0.598	0.544
	336	0.246	0.282	0.251	0.279	0.246	**0.273**	0.265	0.319	**0.245**	0.282	0.280	0.306	0.339	0.380	0.359	0.395	0.321	0.338	0.298	0.353	0.578	0.523
	720	0.328	0.337	0.345	0.342	0.320	**0.328**	0.333	0.362	**0.314**	0.334	0.365	0.359	0.403	0.428	0.419	0.428	0.414	0.410	0.352	0.388	1.059	0.741
	avg	0.227	0.262	0.244	0.270	0.227	**0.255**	0.249	0.300	**0.226**	0.264	0.259	0.287	0.309	0.360	0.338	0.382	0.288	0.314	0.271	0.334	0.634	0.548
Traffic	96	**0.344**	**0.236**	0.392	0.267	0.396	0.264	0.410	0.282	0.360	0.249	0.593	0.321	0.587	0.366	0.613	0.388	0.612	0.338	0.607	0.392	0.719	0.391
	192	**0.372**	**0.249**	0.409	0.271	0.412	0.268	0.423	0.287	0.379	0.256	0.617	0.336	0.604	0.373	0.616	0.382	0.613	0.340	0.621	0.399	0.696	0.379
	336	**0.383**	**0.257**	0.434	0.296	0.421	0.273	0.436	0.296	0.392	0.264	0.629	0.336	0.621	0.383	0.622	0.337	0.618	0.328	0.622	0.396	0.777	0.420
	720	**0.422**	**0.280**	0.451	0.291	0.455	0.291	0.466	0.315	0.432	0.286	0.640	0.350	0.626	0.382	0.660	0.408	0.653	0.355	0.632	0.396	0.864	0.472
	avg	**0.380**	**0.256**	0.422	0.281	0.421	0.274	0.434	0.295	0.391	0.264	0.620	0.336	0.610	0.376	0.628	0.379	0.624	0.340	0.621	0.396	0.764	0.416
Electricity	96	**0.126**	**0.218**	0.137	0.233	0.141	0.239	0.140	0.237	0.129	0.222	0.168	0.272	0.193	0.308	0.201	0.317	0.169	0.273	0.187	0.304	0.274	0.368
	192	**0.144**	**0.237**	0.152	0.247	0.158	0.253	0.153	0.249	0.157	0.240	0.184	0.289	0.201	0.315	0.222	0.334	0.182	0.286	0.199	0.315	0.296	0.386
	336	**0.162**	**0.256**	0.169	0.267	0.172	0.266	0.169	0.267	0.163	0.259	0.198	0.300	0.214	0.329	0.231	0.338	0.200	0.304	0.212	0.329	0.300	0.394
	720	**0.192**	**0.286**	0.200	0.290	0.207	0.293	0.203	0.301	0.197	0.290	0.220	0.320	0.246	0.355	0.254	0.361	0.222	0.321	0.233	0.345	0.373	0.439
	avg	**0.156**	**0.249**	0.165	0.259	0.170	0.263	0.166	0.264	0.162	0.253	0.193	0.295	0.214	0.327	0.227	0.338	0.193	0.296	0.208	0.323	0.311	0.397
1st count		**46**		4		12		0		19		0		0		0		0		0		0	

- **VM2Attn** replaces both the encoder and decoder with a self-attention layer, matching `MAE` structure but with random initialization.
- **VM2Trsf** is similar to **VM2Attn** but replaces them with a Transformer block (*i.e.*, a self-attention layer plus an MLP layer).
- **Rand-VM** keeps the same architecture as the vanilla `MAE`, but all the weights are randomly initialized.

We also compare fine-tuning different components in `MAE` as follows:

- **All** fine-tunes all the trainable weights in `MAE`.
- **LN** fine-tunes only the layer normalization, which is the default setting used in our experiments.
- **Bias** fine-tunes only the bias term of all the linear layers, proposed by Zaken et al. (2022).
- **MLP** and **Attn** fine-tune only the feed-forward layer and the self-attention layer, respectively.

Table 22. Ablation studies (left) and fine-tuning strategies (right). Results are averaged on four prediction lengths: {96, 192, 336, 720}.

		Ablation on Visual MAE (VM)						*Ablation on trained parameters*						
		-	**w/o VM**	**VM2Attn**	**VM2Trsf**	**Rand-VM**		**All**	**LN**	**Bias**	**MLP**	**Attn**	**Freeze**
ETTh1	MSE	**0.395**	0.785	0.448	0.459	0.534	ETTh1	0.534	**0.395**	0.401	0.534	0.554	0.419
	MAE	**0.409**	0.649	0.458	0.462	0.470		0.470	**0.409**	0.414	0.471	0.479	0.418
ETTh2	MSE	**0.336**	0.420	0.418	0.448	0.411	ETTh2	0.411	**0.336**	0.347	0.401	0.392	0.340
	MAE	**0.382**	0.453	0.445	0.457	0.432		0.432	0.382	0.392	0.419	0.414	**0.376**
ETTm1	MSE	**0.338**	0.676	0.397	0.398	0.433	ETTm1	0.433	**0.338**	0.343	0.441	0.444	0.374
	MAE	**0.367**	0.562	0.415	0.410	0.413		0.413	**0.367**	0.368	0.415	0.415	0.372
ETTm2	MSE	**0.261**	0.379	0.274	0.292	0.288	ETTm2	0.288	0.261	**0.256**	0.292	0.289	0.305
	MAE	**0.319**	0.415	0.334	0.344	0.341		0.341	0.319	**0.318**	0.342	0.339	0.334
Average	MSE	**0.333**	0.565	0.384	0.399	0.417	**Average**	0.417	**0.333**	0.337	0.417	0.420	0.360
	MAE	**0.369**	0.520	0.413	0.418	0.414		0.414	**0.369**	0.373	0.412	0.412	0.375
1^{st} count		**10**	0	0	0	0	**1^{st} count**	0	**7**	2	0	0	1

- **Freeze** does not fine-tune any weight. Note that it differs from the previous zero-shot experiment, where a longer context length was used (see Table 8 and Table 19).

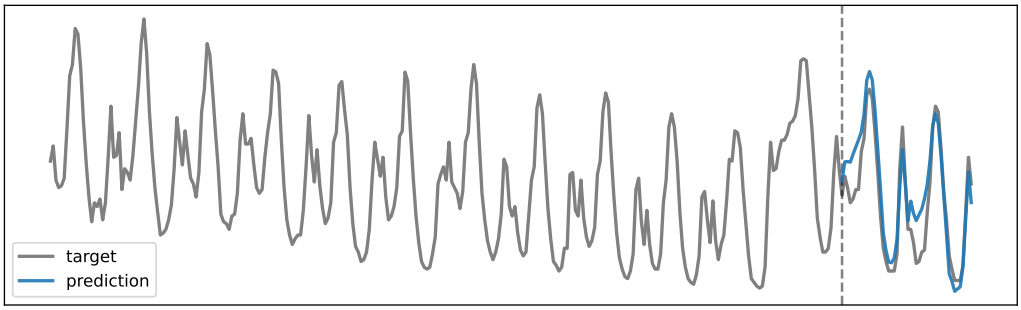
The results are shown in Table 22, suggesting that visual knowledge is crucial for VISIONTS and fine-tuning the layer normalization is the best.

D. Visualization

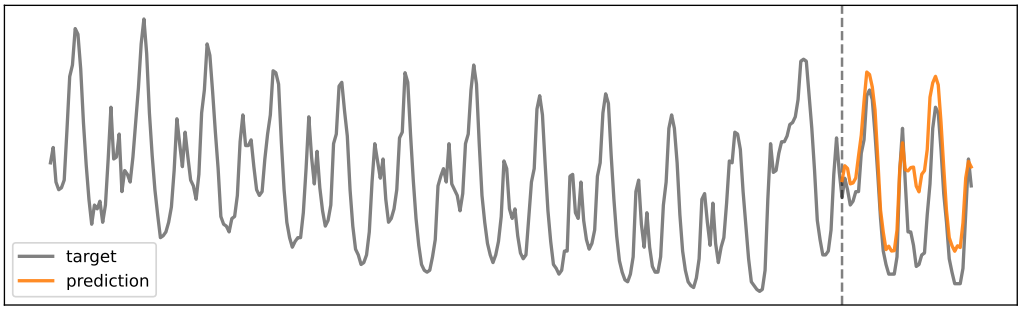
We visualized the predictions of VISIONTS in the zero-shot setting, including its input and reconstructed images. We also visualized the predictions of MOIRAI$_{\text{Large}}$ and Seasonal Naïve, with their MAE metrics for comparison. Figs. 11 to 13 show examples where VISIONTS performed well, with Fig. 11 depicting a more regular pattern, while Figs. 12 and 13 display less obvious patterns. Fig. 14 illustrates a case where VISIONTS underperformed, as it aggressively predicted the trend despite the lack of clear patterns in the input sequence, whereas MOIRAI$_{\text{Large}}$ made more conservative predictions.

(a) Input Image

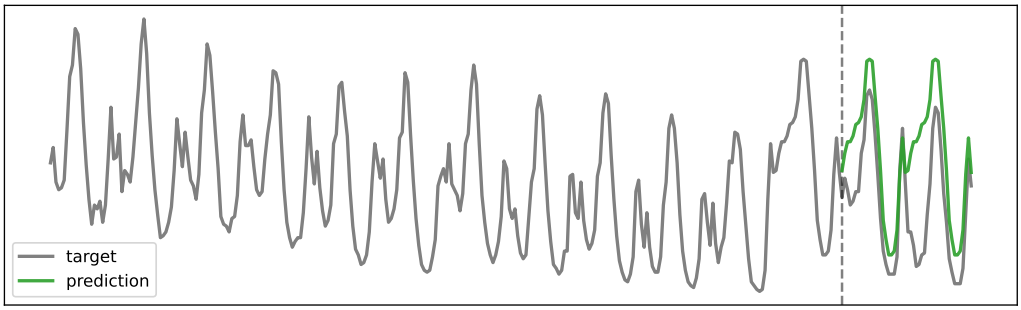
(b) Reconstructed Image



(c) VISIONTS (MAE = 0.312)



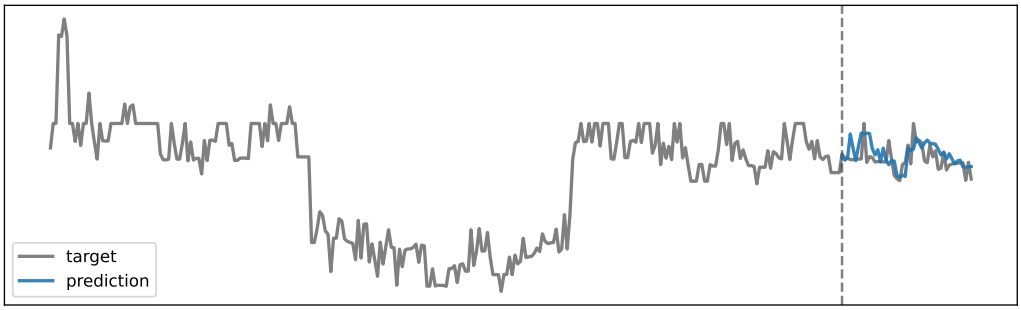
(d) MOIRAI$_{\text{LARGE}}$ (MAE = 0.503)



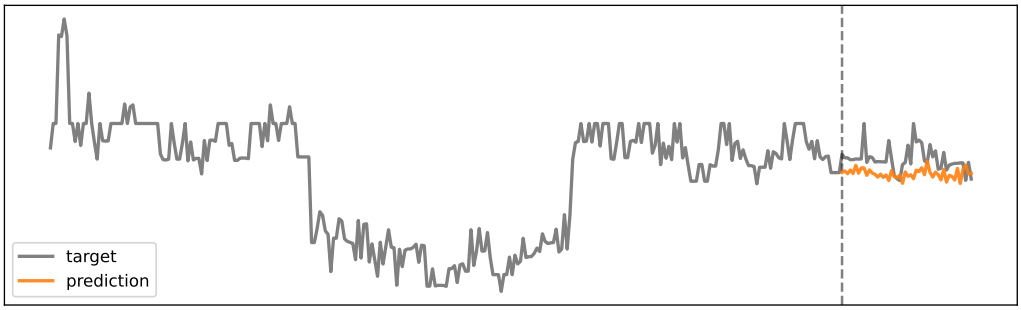
(e) Seasonal Naïve (MAE = 0.774)

Figure 11. Forecasting visualization on a sample from ETTh1. (a-b) Input/output images of VISIONTS. (c-e) Forecasting visualization.

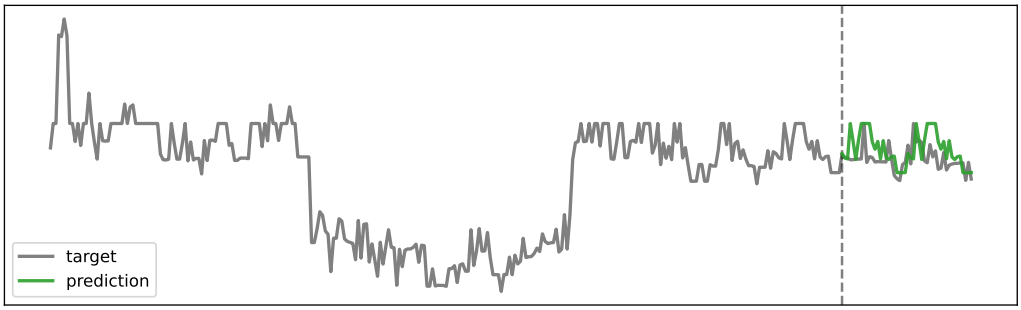
(a) Input Image (b) Reconstructed Image



(c) VISIONTS (MAE = 0.157)



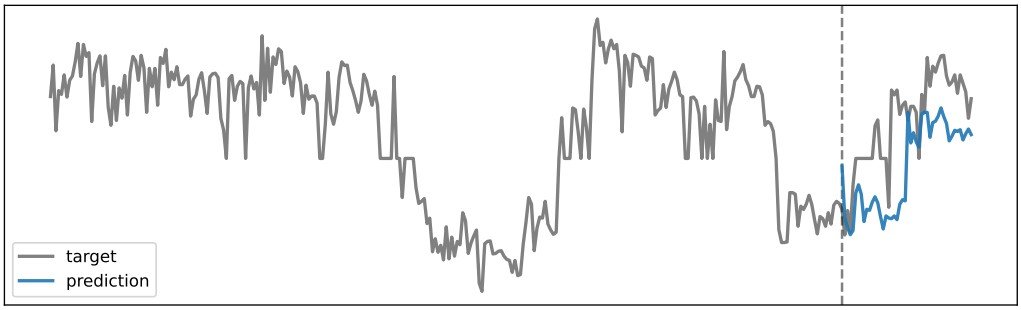
(d) MOIRAI$_{\text{LARGE}}$ (MAE = 0.251)



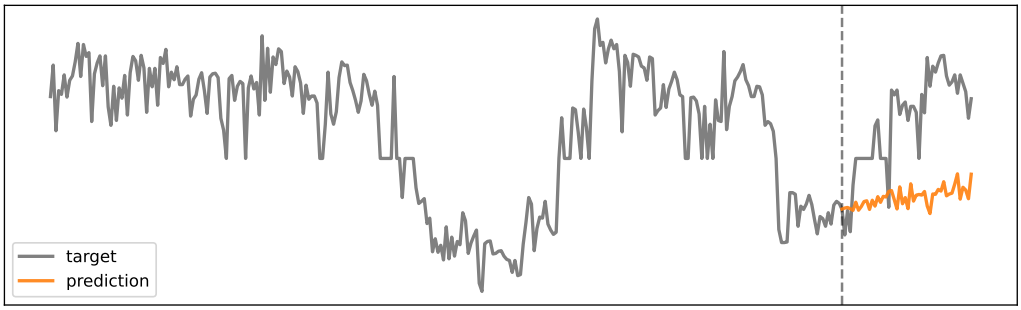
(e) Seasonal Naïve (MAE = 0.235)

Figure 12. Forecasting visualization on a sample from ETTh2. (a-b) Input/output images of VISIONTS. (c-e) Forecasting visualization.

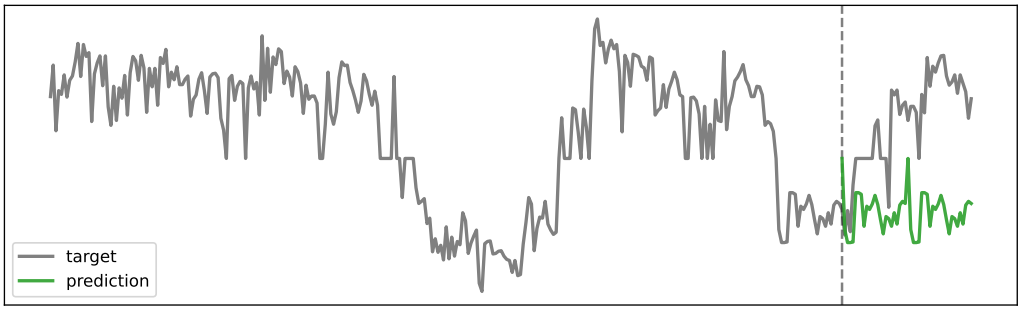
(a) Input Image (b) Reconstructed Image



(c) VISIONTS (MAE = 0.821)



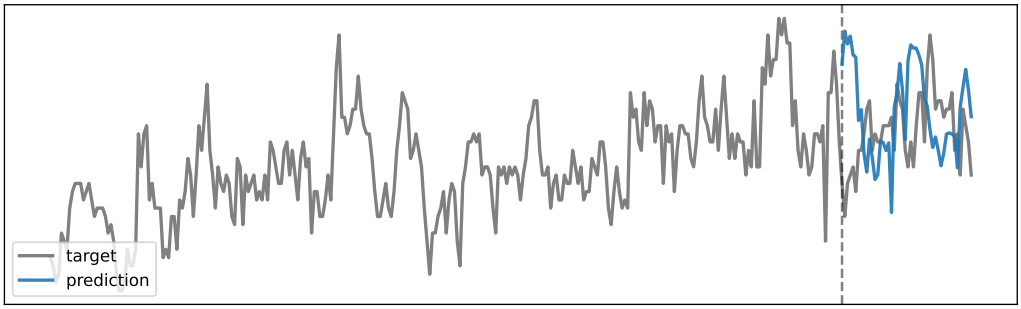
(d) MOIRAI$_{\text{LARGE}}$ (MAE = 1.285)



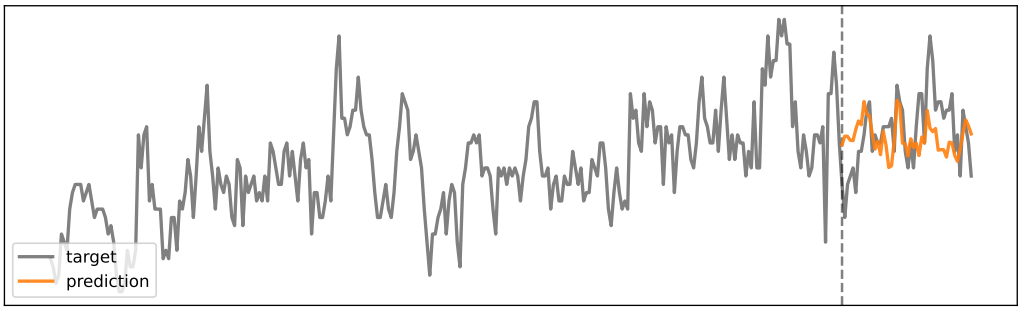
(e) Seasonal Naïve (MAE = 1.523)

Figure 13. Forecasting visualization on a sample from ETTh2. (a-b) Input/output images of VISIONTS. (c-e) Forecasting visualization.

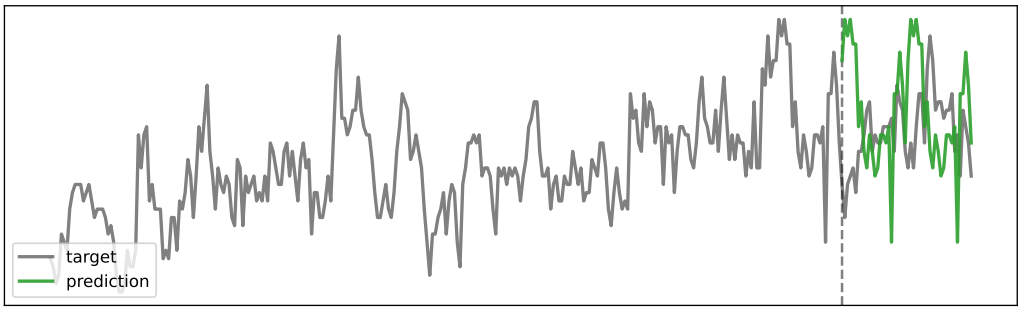
(a) Input Image (b) Reconstructed Image



(c) VISIONTS (MAE = 0.327)



(d) MOIRAI$_{\text{LARGE}}$ (MAE = 0.172)



(e) Seasonal Naïve (MAE = 0.364)

Figure 14. Forecasting visualization on a sample from ETTh1, where MOIRAI outperforms VISIONTS in terms of MAE. (a-b) Input/output images of VISIONTS. (c-e) Forecasting visualization.