Unified Training of Universal Time Series Forecasting Transformers

Gerald Woo [1 2] **Chenghao Liu** [1] **Akshat Kumar** [2] **Caiming Xiong** [1] **Silvio Savarese** [1] **Doyen Sahoo** [1]

Abstract

Deep learning for time series forecasting has traditionally operated within a one-model-per-dataset framework, limiting its potential to leverage the game-changing impact of large pre-trained models. The concept of *universal forecasting*, emerging from pre-training on a vast collection of time series datasets, envisions a single Large Time Series Model capable of addressing diverse downstream forecasting tasks. However, constructing such a model poses unique challenges specific to time series data: i) cross-frequency learning, ii) accommodating an arbitrary number of variates for multivariate time series, and iii) addressing the varying distributional properties inherent in large-scale data. To address these challenges, we present novel enhancements to the conventional time series Transformer architecture, resulting in our proposed **M**asked Enc**O**der-based Un**I**ve**R**s**A**l **T**ime Series Forecasting Transformer (**MOIRAI**). Trained on our newly introduced Large-scale Open Time Series Archive (LOTSA) featuring over 27B observations across nine domains, MOIRAI achieves competitive or superior performance as a zero-shot forecaster when compared to full-shot models. Code, data, and model weights can be found at https://github.com/SalesforceAIResearch/uni2ts.

1. Introduction

In the era of foundation models (FMs) (Bommasani et al., 2021), the landscape of deep learning for time series forecasting is experiencing a revolution. In contrast to FMs capable of tackling a multitude of downstream tasks, the current deep forecasting paradigm, involving training a model on a single dataset with a fixed context and prediction length, appears increasingly antiquated, lacking the capacity to generalize or adapt to diverse scenarios or datasets. Given the unreasonable effectiveness of large pre-trained models in improving performance and data efficiency via transfer learning in modalities like vision and language (Dosovitskiy et al., 2020; Brown et al., 2020), we are starting to see a push to transition away from the existing paradigm, towards a *universal forecasting* paradigm (see Figure 1), where a single large pre-trained model is able to handle any time series forecasting problem. However, the road to building a universal time series forecasting model is mired with challenges.

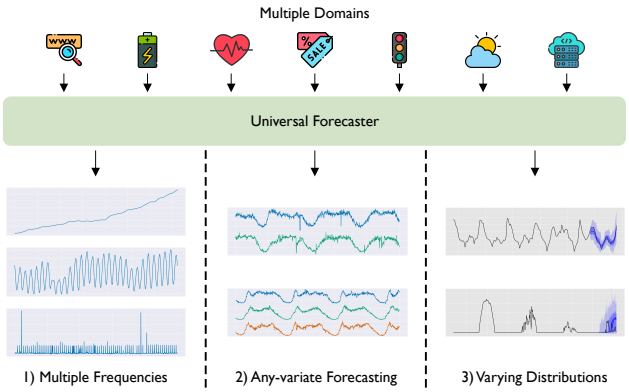


Figure 1. A *universal forecaster* is a large pre-trained model capable of handling any time series forecasting problem. It is trained on a large-scale time series dataset spanning multiple domains. Compared to the existing paradigm, universal forecasting faces the three key issues of i) multiple frequencies, ii) any-variate forecasting, and iii) varying distributions.

Unlike the modalities of vision and language which have the unified formats of images and text respectively, time series data is highly heterogeneous. Firstly, the frequency (e.g. minutely, hourly, daily sampling rates) of time series plays an important role in determining the patterns present in the time series. Cross-frequency learning has been shown to be a challenging task due to negative interference (Van Ness et al., 2023), with existing work simply avoiding this problem for multi-frequency datasets by learning one model per frequency (Oreshkin et al., 2020). Secondly, time series data are heterogeneous in terms of dimensionality, whereby multivariate time series can have different number of variates. Furthermore, each variate measures a semantically different quantity across datasets. While considering each

[1]Salesforce AI Research [2]School of Computing and Information Systems, Singapore Management University. Correspondence to: Gerald Woo <gwoo@salesforce.com>, Chenghao Liu <chenghao.liu@salesforce.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Table 1. Comparison between pre-trained forecasting models. Further discussion on the notion of a "flexible distribution" can be found in Appendix B.3.

	Any-variate (Zero-shot)	Probabilistic Forecasting	Flexible Distribution	Pre-training Data (Size)	Open-source
MOIRAI	✓	✓	✓	LOTSA (> 27B)	✓
TimeGPT-1	✓	✓	✗	Unknown (100B)	✗
ForecastPFN	✗	✗	-	Synthetic Data (60M)	✓
Lag-Llama	✗	✓	✗	Monash (< 1B)	✓
TimesFM	✗	✗	-	Wiki + Trends + Others (> 100B)	✓
TTM	✗	✗	-	Monash (< 1B)	✓
LLMTime	✗	✓	✓	Web-scale Text	✓

variate of a multivariate time series independently (Nie et al., 2023; Ekambaram et al., 2023) can sidestep this problem, we expect a universal model to be sufficiently flexible to consider multivariate interactions and take into account exogenous covariates. Thirdly, probabilistic forecasting is a critical feature often required by practitioners, yet, different datasets have differing support and distributional properties – for example, using a symmetric distribution (e.g. Normal, Student-T) as the predictive distribution is not suitable for positive time series – making standard approaches of pre-defining a simple parametric distribution (Salinas et al., 2020) to be insufficiently flexible to capture a wide variety of datasets. Lastly, a large pre-trained model capable of universal forecasting requires a large-scale dataset from diverse domains. Existing time series datasets are insufficiently large to support the training of such models.

Starting from a masked encoder architecture which has been shown to be a strong candidate architecture for scaling up pre-trained time series forecasting models (Woo et al., 2023), we alleviate the above issues by introducing novel modifications which allows the architecture to handle the heterogeneity of arbitrary time series data. Firstly, we propose to learn multiple input and output projection layers to handle the differing patterns from time series of varying frequencies. Using patch-based projections with larger patch sizes for high-frequency data and vice versa, projection layers are specialized to learn the patterns of that frequency. Secondly, we address the problem of varying dimensionality with our proposed Any-variate Attention, which simultaneously considers both time and variate axes as a single sequence, leveraging Rotary Position Embeddings (RoPE) (Su et al., 2024), and learned binary attention biases (Yang et al., 2022b) to encode time and variate axes respectively. Importantly, Any-variate Attention allows the model to take as input arbitrary number of variates. Thirdly, we overcome the issue of requiring flexible predictive distributions with a mixture of parametric distributions. Furthermore, optimizing the negative log-likelihood of a flexible distribution has the added benefit of being competitive with target metric optimization (Awasthi et al., 2022), a powerful feature for pre-training universal forecasters, given that it can be evaluated with any target metric subsequently.

To power the training of our Large Time Series Model (LTM), we introduce the Large-scale Open Time Series Archive (LOTSA), the largest collection of open time series datasets with 27B observations across nine domains. We optimize the negative log-likelihood of the mixture distribution, and randomly sample context and prediction lengths during training, allowing for flexible downstream usage of the pre-trained model. We train our proposed method, **M**asked Enc**O**der-based Un**I**ve**R**s**A**l **T**Ime Series Forecasting Transformer (MOIRAI[1]), in three sizes – $\text{MOIRAI}_{\text{Small}}$, $\text{MOIRAI}_{\text{Base}}$, and $\text{MOIRAI}_{\text{Large}}$, with 14m, 91m, and 311m parameters respectively. We perform experimental evaluations on both in and out-of-distribution settings, and show that MOIRAI consistently achieves competitive or superior performance compared to state-of-the-art full-shot baselines. Our contributions are summarized as follows:

1. We introduce a novel Transformer architecture to support the requirements of universal forecasting. Crucially, the components we propose extend beyond masked encoders and are versatile, applicable to a broad range of Transformer variants.

2. We introduce LOTSA, a new large-scale collection of open time series datasets to empower pre-training of LTMs. LOTSA, the model weights, and our library for unified training of universal time series models, UNI^2TS, will be fully open sourced.

3. Trained on LOTSA data, MOIRAI achieves competitive or superior performance as a zero-shot forecaster when compared to full-shot models.

2. Related Work

Pre-training for Zero-shot Forecasting Table 1 provides a summary of the key differences between recent pre-trained models with zero-shot forecasting capabilities, which is a recently emerging field. TimeGPT-1 (Garza & Mergenthaler-Canseco, 2023) first presented a closed-source model, offering zero-shot forecasting capabilities as well as supporting fine-tuning through an API, currently only available to their beta users. ForecastPFN (Dooley et al., 2023) proposes to pre-train on synthetic time series, which can be subsequently be leveraged as a zero-shot forecaster, albeit specialized for data or time limited settings. Lag-llama (Rasul et al., 2023) works towards a foundation model for time series forecasting, leveraging the LLaMA (Touvron et al., 2023) architecture design with lagged time series features, and also presents neural scaling laws for time series forecasting. TimesFM (Das et al., 2023b) is a patch-based decoder-only foundation model for time series forecasting, introducing a

[1]In ancient Greek religion and mythology, the Moirai, often known in English as the Fates, were the personifications of destiny. (Wikipedia contributors, 2024)

larger output patch size for faster decoding. They collected a massive amount of data from Google Trends and Wiki pageviews to pre-train their model in combination with open-data. Tiny Time Mixers (TTMs) (Ekambaram et al., 2024) is a concurrent work leveraging lightweight mixer-style architecture. They perform data augmentation by downsampling high-frequency time series, and support multivariate downstream tasks by fine-tuning an exogenous mixer. leverage Large Language Models (LLMs), pre-trained on web-scale text data, have been leveraged for zero-shot forecasting. Specifically, LLMTime (Gruver et al., 2023) treats time series as strings, applying careful pre-processing based on the specific LLMs' tokenizer, showing that LLMs have the inherent capability to perform zero-shot forecasting.

Pre-train + Fine-tune for Time Series Forecasting Pre-training with subsequent fine-tuning on downstream forecasting tasks has predated the recent zero-shot forecasting efforts. Denoising autoencoders (Zerveas et al., 2021) and contrastive learning (Yue et al., 2022; Woo et al., 2022) have been shown to be effective pretext tasks for time series forecasting, but have largely been applied to the existing paradigm of pre-training and fine-tuning on the same dataset, without exploring their generalization capabilities. More recently, Dong et al. (2023) explored combining both reconstruction and contrastive based pre-training approaches, and performed initial explorations into cross-dataset transfer. The topic has been well explored, and we refer readers to more comprehensive surveys (Zhang et al., 2023; Ma et al., 2023). "Reprogramming" is a recent direction which involves fine-tuning the model weights of an LLM which has been pre-trained on text data, for downstream tasks for other modalities. Zhou et al. (2023); Jin et al. (2023) introduce modules and fine-tuning methods to adapt LLMs for time series tasks including forecasting. Liu et al. (2024) has explored leveraging pre-trained LLMs on the cross-dataset setting.

3. Method

Problem Formulation Consider a dataset of N time series $\mathcal{D} = \{(\boldsymbol{Y}^{(i)}, \boldsymbol{Z}^{(i)})\}_{i=1}^{N}$, where $\boldsymbol{Y}^{(i)} = (\boldsymbol{y}_1^{(i)}, \boldsymbol{y}_2^{(i)}, \ldots, \boldsymbol{y}_{T_i}^{(i)}) \in \mathbb{R}^{d_{y_i} \times T_i}$ is a target time series of d_{y_i} variates and T_i time steps. Each time series is associated with a set of covariates $\boldsymbol{Z}^{(i)} = (\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \ldots, \boldsymbol{z}_{T_i}^{(i)}) \in \mathbb{R}^{d_{z_i} \times T_i}$. The goal is to forecast the predictive distribution $p(\boldsymbol{Y}_{t:t+h}|\boldsymbol{\phi})$ by predicting distribution parameters $\boldsymbol{\phi}$ via a learned model $f_{\boldsymbol{\theta}} : (\boldsymbol{Y}_{t-l:t}, \boldsymbol{Z}_{t-l:t+h}) \mapsto \hat{\boldsymbol{\phi}}$ which maximizes the log-likelihood:

$$\max_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\substack{(\mathbf{Y},\mathbf{Z})\sim p(\mathcal{D}) \\ (\mathrm{t,l,h})\sim p(\mathcal{T}|\mathcal{D})}} \log p(\mathbf{Y}_{\mathrm{t:t+h}}|\hat{\boldsymbol{\Phi}}),$$
$$\text{s.t. } \hat{\boldsymbol{\Phi}} = f_{\boldsymbol{\theta}}(\mathbf{Y}_{\mathrm{t-l:t}}, \mathbf{Z}_{\mathrm{t-l:t+h}}), \tag{1}$$

where $p(\mathcal{D})$ is the data distribution which samples for a time series, $(\boldsymbol{Y}, \boldsymbol{Z})$, and $p(\mathcal{T}|\mathcal{D})$ is the task distribution which defines the lookback window, $\boldsymbol{Y}_{t-l:t} = (\boldsymbol{y}_{t-l}, \ldots, \boldsymbol{y}_{t-1})$ with context length l and forecast horizon, $\boldsymbol{Y}_{t:t+h} = (\boldsymbol{y}_t, \ldots, \boldsymbol{y}_{t+h-1})$ with prediction length h.

3.1. Architecture

Illustrated in Figure 2, MOIRAI follows a (non-overlapping) patch-based approach to modeling time series with a masked encoder architecture. One of our proposed modifications to extend the architecture to the any-variate setting is to "flatten" multivariate time series, considering all variates as a single sequence. Patches are subsequently projected into vector representations via a multi patch size input projection layer. The `[mask]` signifies a learnable embedding which replaces patches falling within the forecast horizon. The output tokens are then decoded via the multi patch size output projection into the parameters of the mixture distribution. While not visualized, (non-learnable) instance normalization (Kim et al., 2022) is applied to inputs/outputs, aligning with the current standard practice for deep forecasting models.

The core Transformer module is an encoder-only Transformer architecture, leveraging various improvements as proposed by recent state-of-the-art LLM architectures. We use pre-normalization (Xiong et al., 2020) and replace all LayerNorms with RMSNorm (Zhang & Sennrich, 2019), and also apply query-key normalization (Henry et al., 2020). The non-linearity in FFN layers are replaced with SwiGLU (Shazeer, 2020), adjusting the hidden dimension to have equal number of parameters as the original FFN layer. We omit biases in all layers of the Transformer module.

3.1.1. MULTI PATCH SIZE PROJECTION LAYERS

In the context of universal forecasting, a single model should possess the capability to handle time series spanning a wide range of frequencies. Existing patch-based architectures rely on a single patch size hyperparameter, a legacy feature from the prevailing one-model-per-dataset paradigm. Instead, we aim for a more flexible strategy: opting for a larger patch size to handle high-frequency data, thereby lower the burden of the quadratic computation cost of attention while maintaining a long context length. Simultaneously, we advocate for a smaller patch size for low-frequency data to transfer computation to the Transformer layers, rather than relying solely on simple linear embedding layers. To implement this approach, we propose learning multiple input and output embedding layers, each associated with varying patch sizes. The selection of the appropriate patch size for a given time series frequency relies on pre-defined settings (see Appendix B.1). Note that we only learn one set of projection weights per patch size, which is shared amongst frequencies if there is an overlap based on the settings.

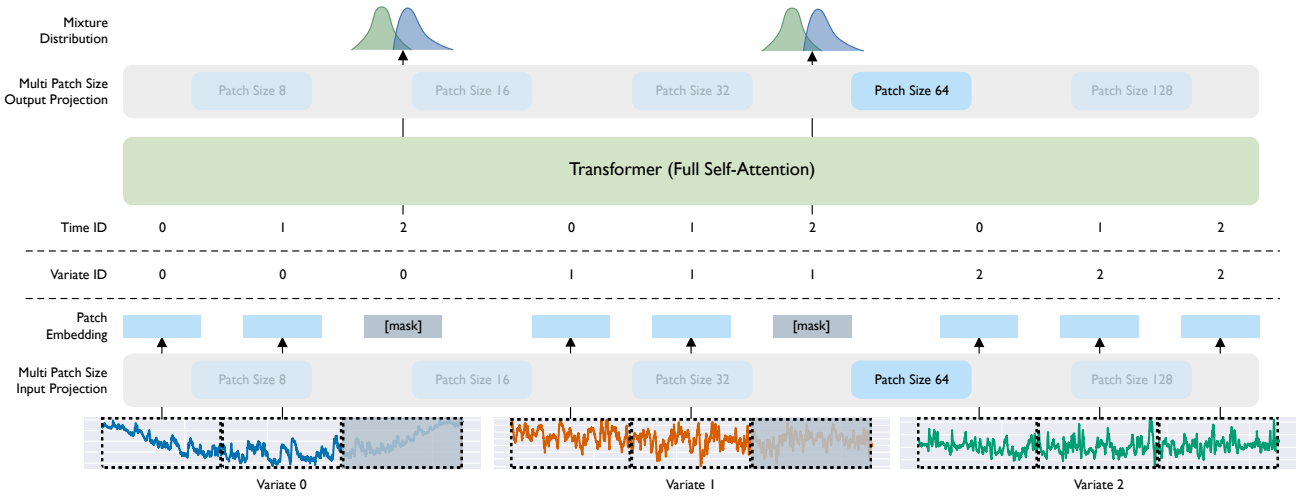


Figure 2. Overall architecture of MOIRAI. Visualized is a 3-variate time series, where variates 0 and 1 are target variables (i.e. to be forecasted, and variate 2 is a dynamic covariate (values in forecast horizon known). Based on a patch size of 64, each variate is patchified into 3 tokens. The patch embeddings along with sequence and variate id are fed into the Transformer. The shaded patches represent the forecast horizon to be forecasted, whose corresponding output representations are mapped into the mixture distribution parameters.

3.1.2. ANY-VARIATE ATTENTION

Universal forecasters must be equipped to handle arbitrary multivariate time series. Existing time series Transformers often rely on an independent variate assumption or are limited to a single dimensionality due to embedding layers mapping $\mathbb{R}^{d_y} \rightarrow \mathbb{R}^{d_h}$, where $\mathbb{R}^{d_h}$ is the hidden dimension. We overcome this limitation as shown in Figure 2, by flattening a multivariate time series to consider all variates as a single sequence. This introduces a new requirement of having variate encodings to enable the model to disambiguate between different variates in the sequence. Furthermore, we need to ensure that permutation equivariance w.r.t. variate ordering, and permutation invariance w.r.t. variate indices are respected. Conventional approaches like sinusoidal or learned embeddings do not meet these requirements, and are unable to handle an arbitrary number of variates. To address this, we propose Any-variate Attention, leveraging binary attention biases to encode variate indices.

Dropping layer and attention head indices, and scaling factor for brevity, the attention score between the (i, m)-th query where i represents the time index and m represents the variate index, and the (j, n)-th key, $A_{ij,mn} \in \mathbb{R}$, is given by:

$$E_{ij,mn} = (\boldsymbol{W}^Q \boldsymbol{x}_{i,m})^T \boldsymbol{R}_{i-j} (\boldsymbol{W}^K \boldsymbol{x}_{j,n}) + u^{(1)} * \mathbb{1}_{\{m=n\}} + u^{(2)} * \mathbb{1}_{\{m \neq n\}}, \quad (2)$$

$$A_{ij,mn} = \frac{\exp\{E_{ij,mn}\}}{\sum_{k,o} \exp\{E_{ik,mo}\}}, \quad (3)$$

where $\boldsymbol{W}^Q \boldsymbol{x}_{i,m}, \boldsymbol{W}^K \boldsymbol{x}_{j,n} \in \mathbb{R}^{d_h}$ are the respective query and key vectors, $\boldsymbol{R}_{i-j} \in \mathbb{R}^{d_h \times d_h}$ is the rotary matrix (Su et al., 2024), $u^{(1)}, u^{(2)} \in \mathbb{R}$ are learnable scalars for each head in each layer, and $\mathbb{1}_{\{\text{cond}\}} = \begin{cases} 1, \text{if cond} \\ 0, \text{otherwise} \end{cases}$ is the indicator function. The binary attention bias component allows for disambiguation between variates via attention scores, fulfills the criteria of permutation equivariance/invariance w.r.t. variate ordering/indices, and can extend to arbitrary number of variates.

3.1.3. MIXTURE DISTRIBUTION

To achieve the goal of having a flexible distribution, yet ensuring that operations of sampling and evaluating the loss function remains simple, we propose to use a mixture of parametric distributions. A mixture distribution of c components has p.d.f.:

$$p(\mathbf{Y}_{t:t+h} | \hat{\boldsymbol{\phi}}) = \sum_{i=1}^{c} w_i p_i(\mathbf{Y}_{t:t+h} | \hat{\boldsymbol{\phi}}_i), \quad (4)$$

where $\hat{\boldsymbol{\phi}} = \{w_1, \hat{\boldsymbol{\phi}}_1, \ldots, w_c, \hat{\boldsymbol{\phi}}_c\}$, and p_i is the i-th component's p.d.f. While the choice of mixture components is flexible and implementing any combination of parametric distributions is straightforward, we specifically propose to use the following mixture components: $i)$ a Student's t-distribution which has shown to be a robust option for general time series, $ii)$ a negative binomial distribution for positive count data, $iii)$ a log-normal distribution to model right-skewed data commonly across economic and and natural phenomena, and $iv)$ a low variance normal distribution for high confidence predictions. Further details can be found in Appendix B.2.

Table 2. Key statistics of LOTSA by domain.

	Energy	Transport	Climate	CloudOps	Web	Sales	Nature	Econ/Fin	Healthcare
# Datasets	30	23	6	3	3	6	5	23	6
# Obs.	16,358,600,896	4,900,453,419	4,188,011,890	1,518,268,292	428,082,373	197,984,339	28,547,647	24,919,596	1,594,281
%	59.17%	17.73%	15.15%	5.49%	1.55%	0.72%	0.09%	0.10%	0.01%

Table 3. Key statistics of LOTSA by frequency.

	Yearly	Quarterly	Monthly	Weekly	Daily	(Multi) Hourly	(Multi) Minute-level	(Multi) Second-level
# Datasets	4	5	10	7	21	31	25	2
# Obs.	873,297	2,312,027	11,040,648	18,481,871	709,017,118	19,875,993,973	7,013,949,430	14,794,369
%	0.003%	0.008%	0.040%	0.067%	2.565%	71.893%	25.370%	0.054%

3.2. Unified Training

3.2.1. LOTSA Data

Existing work has predominantly relied on three primary sources of data – the Monash Time Series Forecasting Archive (Godahewa et al., 2021), datasets provided by the GluonTS library (Alexandrov et al., 2020), and datasets from the popular long sequence forecasting benchmark (Lai et al., 2018; Wu et al., 2021). While Monash and GluonTS comprise of datasets from diverse domains, they are constrained in size, with approximately 1B observations combined. In comparison, LLMs are trained on *trillions* of tokens. Das et al. (2023b) builds a private dataset mainly based on Google Trends and Wiki pageviews, but lacks diversity in terms of the domains these time series originate from.

The effectiveness of FMs heavily stem from large-scale pre-training data. Given that existing data sources fall short of supporting such a paradigm, attempting to train an LTM on them may result in misleading conclusions. Thus, we tackle this issue head-on by building a large-scale archive of open time series datasets by collating publicly available sources of time series datasets. This effort aims to cover a broad spectrum of domains, consolidating datasets from diverse sources with varying formats. We design a unified storage format using Arrow (Richardson et al., 2023) which is ready for deep learning pipelines. The resulting collection, LOTSA, spans nine domains, with a total of $27,646,462,733$ observations, with key statistics in Tables 2 and 3, and in-depth details in Appendix A.

3.2.2. Pre-training

As introduced in Equation (1), our pre-training task is formulated to optimize the mixture distribution log-likelihood. The design of both the data distribution and task distribution are two critical aspects of the pre-training pipeline. This design imparts versatile capabilities to our LTM, enabling it to adapt to a range of downstream tasks. This flexibility stands in contrast to the prevailing deep forecasting paradigm, where models are typically specialized for specific datasets and settings.

Data Distribution The data distribution, $(\mathbf{Y}, \mathbf{Z}) \sim p(\mathcal{D})$, defines how time series are sampled from the dataset. Trained on LOTSA, which is a dataset of datasets, we introduce the notion of sub-datasets, by decomposing the data distribution into a sub-dataset distribution, and a time series distribution conditioned on a sub-dataset, $p(\mathcal{D}) = p(\mathbf{Y}, \mathbf{Z}|\mathbf{D})p(\mathbf{D})$. Thus, we first sample a sub-dataset from $p(\mathbf{D})$, and given that sub-dataset, we sample a time series. For K sub-datasets, where $\boldsymbol{D}_k$ represents the set of indices of time series belonging to sub-dataset k, the structure of $p(\boldsymbol{Y}^{(i)}, \boldsymbol{Z}^{(i)}|\boldsymbol{D}_k) = \frac{T_i * \mathbb{1}_{\{i \in \boldsymbol{D}_k\}}}{\sum_{j \in \boldsymbol{D}_k} T_j}$, proportionate to the number of observations, is straightforward.

However, due to data imbalance across domains and frequency, we avoid sampling sub-datasets proportionately, and instead cap the contribution of each sub-dataset at $\epsilon = 0.001$, before re-normalizing: $p(\boldsymbol{D}_k) = \frac{\omega_k}{\sum_{i=1}^{K} \omega_i}$, where $\omega_k = \min(\frac{|\boldsymbol{D}_k|}{\sum_i^K |\boldsymbol{D}_i|}, \epsilon)$, and $|\boldsymbol{D}_k| = \sum_{i \in \boldsymbol{D}_k} T_i$.

Task Distribution Different from the existing deep forecasting paradigm, we aim to train a model with forecasting capabilities over varying context and prediction lengths. Rather than defining a fixed context and prediction length, we sample from a task distribution, $(\mathrm{t}, \mathrm{l}, \mathrm{h}) \sim p(\mathcal{T}|\mathcal{D})$ which defines the lookback window and forecasting horizon, given a time series. In practice, rather than sampling t, l, h, given a time series, we crop a uniformly sampled window, whose length is uniformly sampled from a range. This range is defined by a minimum sequence length per variate of 2, and a total maximum sequence length of 512. The window is then split into lookback and horizon segments, where the prediction length is uniformly sampled as a proportion (within the range $[0.15, 0.5]$) of the window. We further augment training by $i)$ uniformly subsampling multivariate time series in the variate dimension, and $ii)$ constructing multivariate time series from sub-datasets with univariate time series, by randomly concatenating them. The number of variates is sampled from a beta-binomial distribution with parameters $n = 128, a = 2, b = 5$ which supports a maximum of 128 variates, with mean ≈ 37 for efficiency.

Table 4. Details of MOIRAI model sizes.

	Layers	d_{model}	d_{ff}	Heads	d_{kv}	Params
MOIRAI$_{\text{Small}}$	6	384	1536	6	64	14m
MOIRAI$_{\text{Base}}$	12	768	3072	12	64	91m
MOIRAI$_{\text{Large}}$	24	1024	4096	16	64	311m

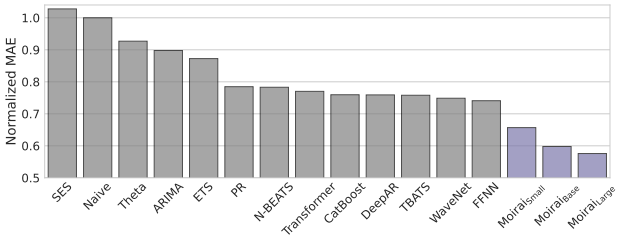


Figure 3. Aggregate results of the Monash Time Series Forecasting Benchmark. The normalized MAE is reported, which normalizes the MAE of each dataset by the naive forecast's MAE, and aggregated by taking the geometric mean across datasets.

Training We train MOIRAI in three sizes – small, base, and large, with key parameter details listed in Table 4. The small model is trained for $100,000$ steps, while base and large models are trained for $1,000,000$ steps with a batch size of 256. For optimization, we use the AdamW optimizer with the following hyperparameters, $\text{lr} = 1\text{e-}3, \text{weight_decay} = 1\text{e-}1, \beta_1 = 0.9, \beta_2 = 0.98$. We also apply a learning rate scheduler with linear warmup for the first $10,000$ steps, and cosine annealing thereafter. Models are trained on NVIDIA A100-40G GPUs with TF32 precision. We implement sequence packing (Raffel et al., 2020) to avoid large amounts of padding due to the disparity of sequence lengths in the new setting with varying context, prediction, and variate lengths, thereby increasing the effective batch size.

4. Experiments

4.1. In-distribution Forecasting

We first perform an in-distribution evaluation using the Monash benchmark, which aim to measure generalization capability across diverse domains. Described in Appendix A, LOTSA includes the Monash Time Series Forecasting Archive as a source of data. For a large portion of these datasets, we only include the train set, holding out the test set which we now use for in-distribution evaluation. In this evaluation, we consider a standard setting with a context length of 1000, and a patch size of 32 for all frequencies, except for quarterly data with a patch size of 8. Figure 3 summarizes the results based on the normalized mean absolute error (MAE), in comparison with the baselines presented in the Monash benchmark. It is worth noting that each baseline in the Monash benchmark is typically trained individually per dataset or per time series within a dataset. In contrast, MOIRAI stands out by being a single model evaluated across various datasets. Full results as well as a comparison with LLMTime (Gruver et al., 2023) can be found in Appendix D.1.

We observe that MOIRAI outperforms all baselines from the Monash benchmark regardless of model size, displaying the strong in-distribution and cross-domain capabilities arising from our unified training methodology. We highlight that each instance of MOIRAI is a **single** model evaluated across datasets, compared to baselines for which one model is trained per dataset. Further analysis on computational costs can be found in Appendix D.4.

4.2. Out-of-distribution / Zero-shot Forecasting

Next, we perform an out-of-distribution evaluation on **unseen target datasets**. Here, MOIRAI is a zero-shot forecaster compared with state-of-the-art full-shot baselines which have been trained on the individual target datasets. While the ideal scenario would be to include other universal forecasters, this proves to be a challenging task. As a nascent field, most universal forecasters currently do not yet have open weights avaiable for evaluation. Furthermore, the problem of comparing zero-shot methods is exacerbated by not having a standard held-out test split, making it challenging to collate a set of datasets which all the models have not been trained on. Thus, we establish the strong zero-shot capabilities of MOIRAI by displaying competitive or stronger results compared with SOTA full-shot methods – datasets used in the following have **not** been included in LOTSA.

Probabilistic Forecasting We evaluate on six datasets across energy, transport, climate, and sales domains, following a rolling evaluation setup with stride equal to prediction length. Prediction lengths and number of rolling evaluations are defined for each dataset based on frequency. We report the Continuous Ranked Probability Score (CRPS) and Mean Scaled Interval Score (MSIS) metrics (definitions in Appendix C), comparing against four full-shot baselines – DeepAR (Salinas et al., 2020), PatchTST (Nie et al., 2023), and TiDE (Das et al., 2023a) with Student's t-distribution prediction heads, and TFT based on quantile prediction (Lim et al., 2021), all implemented with the GluonTS library (Alexandrov et al., 2020), as well as simple baselines AutoARIMA (Garza et al., 2022) and Seasonal Naive (Hyndman & Athanasopoulos, 2018). For each dataset and baseline, we perform hyperparameter tuning on a validation CRPS, and report results averaged over five training runs with different seeds. For MOIRAI, we perform inference time tuning, selecting context length from $\{1000, 2000, 3000, 4000, 5000\}$ and patch sizes based on frequency, on the validation CRPS. Full details of the evaluation setting can be found in Appendix C.

Table 5. Probabilistic forecasting results. Best results are highlighted in **bold**, and second best results are underlined. Baseline results are aggregated over five training runs with different seeds, reporting the mean and standard deviation.

		Zero-shot			Full-shot				Baseline	
		MOIRAI$_{\text{Small}}$	MOIRAI$_{\text{Base}}$	MOIRAI$_{\text{Large}}$	PatchTST	TiDE	TFT	DeepAR	AutoARIMA	Seasonal Naive
Electricity	**CRPS**	0.072	0.055	0.050	0.052±0.00	**0.048±0.00**	0.050±0.00	0.065±0.01	0.327	0.070
	MSIS	7.999	6.172	5.875	5.744±0.12	**5.672±0.08**	6.278±0.24	6.893±0.82	29.412	35.251
Solar	**CRPS**	0.471	0.419	**0.406**	0.518±0.09	0.420±0.00	0.446±0.03	0.431±0.01	1.055	0.512
	MSIS	8.425	7.011	**6.250**	8.447±1.59	13.754±0.32	8.057±3.51	11.181±0.67	25.849	48.130
Walmart	**CRPS**	0.103	0.093	0.098	0.082±0.01	**0.077±0.00**	0.087±0.00	0.121±0.00	0.124	0.151
	MSIS	9.371	8.421	8.520	**6.005±0.21**	6.258±0.12	8.718±0.10	12.502±0.03	9.888	49.458
Weather	**CRPS**	0.049	**0.041**	0.051	0.059±0.01	0.054±0.00	0.043±0.00	0.132±0.11	0.252	0.068
	MSIS	5.236	5.136	**4.962**	7.759±0.49	8.095±1.74	7.791±0.44	21.651±17.34	19.805	31.293
Istanbul Traffic	**CRPS**	0.173	0.116	0.112	0.112±0.00	0.110±0.01	0.110±0.01	**0.108±0.00**	0.589	0.257
	MSIS	5.937	4.461	4.277	**3.813±0.09**	4.752±0.17	4.057±0.44	4.094±0.31	16.317	45.473
Turkey Power	**CRPS**	0.048	0.040	**0.036**	0.054±0.01	0.046±0.01	0.039±0.00	0.066±0.02	0.116	0.085
	MSIS	7.127	6.766	**6.341**	8.978±0.51	8.579±0.52	7.943±0.31	13.520±1.17	14.863	36.256

Table 6. Long sequence forecasting results. Results are averaged across prediction lengths $\{96, 192, 336, 720\}$. Best results are highlighted in **bold**, and second best results are underlined. Full-shot results are obtained from Liu et al. (2023b).

		Zero-shot			Full-shot							
		MOIRAI$_{\text{Small}}$	MOIRAI$_{\text{Base}}$	MOIRAI$_{\text{Large}}$	iTransformer	TimesNet	PatchTST	Crossformer	TiDE	DLinear	SCINet	FEDformer
ETTh1	**MSE**	**0.400**	0.434	0.510	0.454	0.458	0.469	0.529	0.541	0.456	0.747	0.44
	MAE	**0.424**	0.438	0.469	0.448	0.450	0.455	0.522	0.507	0.452	0.647	0.46
ETTh2	**MSE**	**0.341**	0.345	0.354	0.383	0.414	0.387	0.942	0.611	0.559	0.954	0.437
	MAE	0.379	0.382	**0.376**	0.407	0.497	0.407	0.684	0.550	0.515	0.723	0.449
ETTm1	**MSE**	0.448	**0.381**	0.390	0.407	0.400	0.387	0.513	0.419	0.403	0.486	0.448
	MAE	0.409	**0.388**	0.389	0.410	0.406	0.400	0.495	0.419	0.407	0.481	0.452
ETTm2	**MSE**	0.300	**0.272**	0.276	0.288	0.291	0.281	0.757	0.358	0.35	0.571	0.305
	MAE	0.341	0.321	**0.320**	0.332	0.333	0.326	0.611	0.404	0.401	0.537	0.349
Electricity	**MSE**	0.233	0.188	0.188	**0.178**	0.193	0.216	0.244	0.252	0.212	0.268	0.214
	MAE	0.320	0.274	0.273	**0.270**	0.295	0.304	0.334	0.344	0.3	0.365	0.327
Weather	**MSE**	0.242	**0.238**	0.259	0.258	0.259	0.259	0.259	0.271	0.265	0.292	0.309
	MAE	0.267	**0.261**	0.275	0.278	0.287	0.281	0.315	0.320	0.317	0.363	0.36

Table 5 reports the CRPS and MSIS, with full results including deterministic metrics in Appendix D.2. We observe that MOIRAI$_{\text{Base}}$ and MOIRAI$_{\text{Large}}$ consistently achieve strong zero-shot performance, obtaining either best or second best results for all datasets except Walmart and Istanbul Traffic. Even for these datasets, performance is still close to the best performance, despite being a single zero-shot model compared to baselines which have been tuned and trained on the train sets.

Long Sequence Forecasting We evaluate on a subset of the popular long sequence forecasting benchmark (Wu et al., 2021), omitting datasets which have datasets from the same source present in our pre-training data and cannot be considered zero-shot. We report the Mean Squared Error (MSE) and MAE, comparing against six state-of-the-art baselines, iTransformer (Liu et al., 2023b), TimesNet (Wu et al., 2023), PatchTST, Crossformer (Zhang & Yan, 2023), TiDE, DLinear (Zeng et al., 2023), SCINet (Liu et al., 2022), and FEDformer (Zhou et al., 2022b). Point forecasts are obtained from MOIRAI by taking the median from the samples of the predictive distribution. Tuning for MOIRAI was based on the average validation MSE across prediction lengths, further including the options between channel indepedent and channel mixing strategies (Nie et al., 2023) for the low dimension datasets (ETT and Weather).

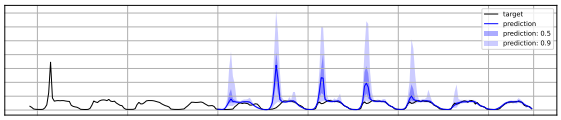
Table 6 reports the average performance across prediction lengths, with full results in Appendix D.3. We observe that MOIRAI achieves strong results compared to full-shot baselines. While MOIRAI$_{\text{Base}}$ consistently achieves strong performance across datasets with either best or second-best performance, the large model is less consistent, with slightly weaker but competitive results. The relationship between performance and model size is tenuous in this setting, however, this does not constitute strong evidence against the potential of scaling, since these results are based on models trained on a fixed dataset size and settings. Rather, this calls for more comprehensive neural scaling laws (Kaplan et al., 2020) for LTMs, to build a stronger understanding of their scaling behavior.

4.3. Ablation Study

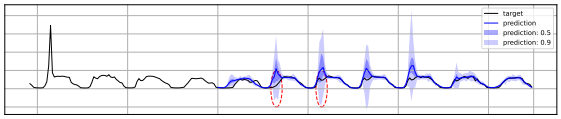
Architecture We perform a series of ablations in Table 7, starting from the default MOIRAI$_{\text{Small}}$. Firstly, we ablate the multi patch size component, removing the constraints by allowing any frequency to have any patch size during training, and also simply fixing the patch size to 32. In both cases, we observe a deterioration in normalized MAE. Removing Any-variate Attention and using additive learned embeddings (randomizing variate index during training to encourage permutation invariance) instead, leads to suboptimal results, showcasing the strength of Any-variate

Table 7. Ablation study on Monash benchmark. The aggregated normalized MAE, similarly calculated as in Figure 3 is reported.

	Normalized MAE
MOIRAI$_{\text{Small}}$	**0.655**
w/o patch size constraints	0.720
w/o multi patch size	1.156
w/o Any-variate Attention	0.904
w/o mixture distribution	0.740
w/o LOTSA	0.809
w/o packing	0.785



(a) Mixture distribution.



(b) Student's t-distribution.

Figure 4. Visualization of probabilistic forecasts by two variants of MOIRAI$_{\text{Small}}$ on the Traffic Hourly dataset. Both models forecast peaks, however, the Student's t-distribution has a symmetric distribution, giving inappropriate prediction intervals for a peak, as highlighted in red.

Attention. We see similar deterioration when replacing the mixture distribution with a Student's t-distribution, and further visualize the necessity of flexible distributions for probabilistic forecasts in Figure 4.

Training Methodology We study the impact of a large and diverse dataset by training MOIRAI$_{\text{Small}}$ only on the GluonTS and Monash datasets, observing that diversity of data is critical for cross-domain training even on in-distribution evaluation. Finally, given the same batch size and training iterations, we show that packed training significantly boosts performance. This is because packing increases the effective batch size and increases the number of observations the model is trained on, given the same amount of compute.

4.4. Further Analysis

Context Length Our pre-training methodology varies context length defined by the task distribution. We verify that MOIRAI has the capability to take as input arbitrary context lengths by visualizing in Figure 5 the relationship between performance and increasing context lengths over three datasets in the zero-shot setting. Zeng et al. (2023); Liu et al. (2023b) previously observed that the desiderata of continuously improving performance with increasing context length is not present in conventional Transformer-based forecasters. Here, we observe that MOIRAI indeed achieves this desired property, in fact, capable of handling thousands of time steps.

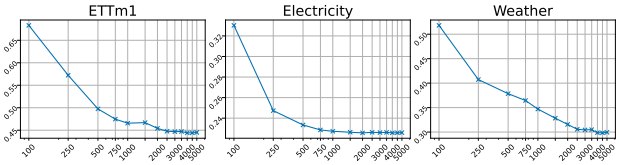


Figure 5. Plot of performance (MAE) against context length (x-axis in log scale) with prediction length 96 and patch size 32 on the validation set of the ETTm1, Electricity, and Weather datasets.

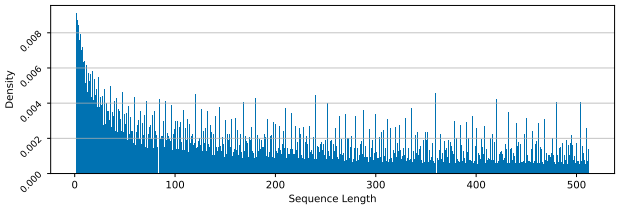


Figure 6. Histogram of sequence length when sampling data from LOTSA based on the proposed task distribution. Sequence length refers to the number of tokens after patching and flattening.

Packing Packing has long been applied in training LLMs and other Transformer-based models, but not for time series Transformers. While we can get away with inefficiencies when dealing with small-scale data, we start to suffer from longer training times as we scale towards the paradigm of FMs and LTMs. This is further exacerbated by our "flattened" setting which increases the disparity in sequence lengths. As evidenced in Section 4.3, keeping compute (batch size, iterations, etc.) constant, packing improves performance by 16%. To understand why this is the case, we visualize sequence length distribution in Figure 6. With a large portion of the data being shorter than the maximum sequence length, padding represents a whopping 61.08% of input tokens without packed training, and only 0.38% with our packed implementation (calculated over 1000 iterations).

5. Conclusion

In this work, we introduced MOIRAI, a masked encoder-based universal time series forecasting Transformer which alleviates the issues faced in the universal forecasting paradigm. We also introduce the LOTSA, the largest collection of open-data for pre-training time series forecasting models. MOIRAI is evaluated on the in-distribution and out-of-distribution settings, and is capable of probabilistic and long sequence forecasting. We show that as a zero-shot forecaster, MOIRAI achieves competitive or superior performance compared to full-shot models.

Limitations & Future Work While MOIRAI achieves phenomenal in and out-of-distribution performance, this is just a first step in the universal forecasting paradigm.

Due to resource constraints, little to no hyperparameter tuning was performed – efficient tuning techniques such as μP (Yang et al., 2022a) can be applied. In terms of architecture, our approach to tackling cross-frequency learning with a multi patch size mapping is somewhat heuristic, and future work should design a more flexible and elegant approach. Also, the current architecture has limited support for high-dimensional time series, and efficient methods for extending Transformer input length can alleviate this issue. The masked encoder structure also makes it amenable to exploration of a latent diffusion architecture (Feng et al., 2024). In terms of data, `LOTSA` can be further enhanced with greater diversity in terms of domain and frequency. Finally, incorporating multi-modality such as tabular or text inputs is an exciting new direction which universal forecasting has unlocked.

Impact Statement

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

References

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., Stella, L., TÃ¼rkmen, 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.

Awasthi, P., Das, A., Sen, R., and Suresh, A. T. On the benefits of maximum likelihood estimation for regression and forecasting. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=zrW-LVXj2k1.

Bergmeir, C., Bui, Q., de Nijs, F., and Stuckey, P. Residential power and battery data, August 2023. URL https://doi.org/10.5281/zenodo.8219786.

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., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

CDC. Flu portal dashboard, 2017. URL https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html.

Chen, C., Petty, K., Skabardonis, A., Varaiya, P., and Jia, Z. Freeway performance measurement system: mining loop detector data. *Transportation Research Record*, 1748(1): 96–102, 2001.

Chen, S. Beijing Multi-Site Air-Quality Data. UCI Machine Learning Repository, 2019. DOI: https://doi.org/10.24432/C5RK5G.

Das, A., Kong, W., Leach, A., Mathur, S. K., Sen, R., and Yu, R. Long-term forecasting with tiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023a. ISSN 2835-8856. URL https://openreview.net/forum?id=pCbC3aQB5W.

Das, A., Kong, W., Sen, R., and Zhou, Y. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023b.

Dong, J., Wu, H., Zhang, H., Zhang, L., Wang, J., and Long, M. SimMTM: A simple pre-training framework for masked time-series modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=ginTcBUnL8.

Dooley, S., Khurana, G. S., Mohapatra, C., Naidu, S. V., and White, C. ForecastPFN: Synthetically-trained zero-shot forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=tScBQRNgjk.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Ekambaram, V., Jati, A., Nguyen, N., Sinthong, P., and Kalagnanam, J. Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, pp. 459–469, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599533. URL https://doi.org/10.1145/3580305.3599533.

Ekambaram, V., Jati, A., Nguyen, N. H., Dayama, P., Reddy, C., Gifford, W. M., and Kalagnanam, J. Ttms: Fast multi-level tiny time mixers for improved zero-shot and few-shot forecasting of multivariate time series. *arXiv preprint arXiv:2401.03955*, 2024.

Emami, P., Sahu, A., and Graf, P. Buildingsbench: A large-scale dataset of 900k buildings and benchmark for short-term load forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=c5rqd6PZn6.

Feng, S., Miao, C., Zhang, Z., and Zhao, P. Latent diffusion transformer for probabilistic time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11979–11987, 2024.

Garza, A. and Mergenthaler-Canseco, M. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

Garza, F., Canseco, M. M., Challú, C., and Olivares, K. G. StatsForecast: Lightning fast forecasting with statistical and econometric models. PyCon Salt Lake City, Utah, US 2022, 2022. URL https://github.com/Nixtla/statsforecast.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

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-.

Gruver, N., Finzi, M. A., Qiu, S., and Wilson, A. G. Large language models are zero-shot time series forecasters. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=md68e8iZK1.

Henry, A., Dachapally, P. R., Pawar, S. S., and Chen, Y. Query-key normalization for transformers. In Cohn, T., He, Y., and Liu, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.379. URL https://aclanthology.org/2020.findings-emnlp.379.

Hyndman, R. J. Errors on percentage errors, 4 2014. URL https://robjhyndman.com/hyndsight/smape/.

Hyndman, R. J. and Athanasopoulos, G. *Forecasting: principles and practice*. OTexts, 2018.

Hyndman, R. J. and Koehler, A. B. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

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. *arXiv preprint arXiv:2310.01728*, 2023.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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.

Lai, G., Chang, W.-C., Yang, Y., and Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Lim, B., Arık, S. Ö., Loeff, N., and Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

Liu, M., Zeng, A., Chen, M., Xu, Z., Lai, Q., Ma, L., and Xu, Q. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

Liu, X., Xia, Y., Liang, Y., Hu, J., Wang, Y., Bai, L., Huang, C., Liu, Z., Hooi, B., and Zimmermann, R. Largest: A benchmark dataset for large-scale traffic forecasting. *arXiv preprint arXiv:2306.08259*, 2023a.

Liu, X., Hu, J., Li, Y., Diao, S., Liang, Y., Hooi, B., and Zimmermann, R. Unitime: A language-empowered unified model for cross-domain time series forecasting. In *Proceedings of the ACM Web Conference 2024*, 2024.

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., and Long, M. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023b.

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.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1): 54–74, 2020.

Mancuso, P., Piccialli, V., and Sudoso, A. M. A machine learning approach for forecasting hierarchical time series. *Expert Systems with Applications*, 182:115102, 2021.

Mouatadid, S., Orenstein, P., Flaspohler, G. E., Oprescu, M., Cohen, J., Wang, F., Knight, S. E., Geogdzhayeva, M., Levang, S. J., Fraenkel, E., and Mackey, L. SubseasonalclimateUSA: A dataset for subseasonal forecasting and benchmarking. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=pWkrU6raMt.

Nguyen, T., Jewik, J. K., Bansal, H., Sharma, P., and Grover, A. Climatelearn: Benchmarking machine learning for weather and climate modeling. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=RZJEkLFlPx.

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*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1ecqn4YwB.

Park, Y., Maddix, D., Aubet, F.-X., Kan, K., Gasthaus, J., and Wang, Y. Learning quantile functions without quantile crossing for distribution-free time series forecasting. In *International Conference on Artificial Intelligence and Statistics*, pp. 8127–8150. PMLR, 2022.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Rasul, K., Ashok, A., Williams, A. R., Khorasani, A., Adamopoulos, G., Bhagwatkar, R., Biloš, M., Ghonia, H., Hassen, N. V., Schneider, A., Garg, S., Drouin, A., Chapados, N., Nevmyvaka, Y., and Rish, I. Lag-llama: Towards foundation models for time series forecasting, 2023.

Richardson, N., Cook, I., Crane, N., Dunnington, D., François, R., Keane, J., Moldovan-Grünfeld, D., Ooms, J., Wujciak-Jens, J., and Apache Arrow. *arrow: Integration to 'Apache' 'Arrow'*, 2023. URL https://github.com/apache/arrow/. R package version 14.0.2, https://arrow.apache.org/docs/r/.

Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Shazeer, N. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 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.

Trindade, A. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

Van Ness, M., Shen, H., Wang, H., Jin, X., Maddix, D. C., and Gopalswamy, K. Cross-frequency time series meta-forecasting. *arXiv preprint arXiv:2302.02077*, 2023.

van Panhuis, W. G., Cross, A., and Burke, D. S. Project tycho 2.0: a repository to improve the integration and reuse of data for global population health. *Journal of the American Medical Informatics Association*, 25:1608–1617, 2018.

Walmart Competition Admin, W. C. Walmart recruiting - store sales forecasting, 2014.

Wang, J., Jiang, J., Jiang, W., Han, C., and Zhao, W. X. Towards efficient and comprehensive urban spatial-temporal prediction: A unified library and performance benchmark. *arXiv preprint arXiv:2304.14343*, 2023a.

Wang, Z., Wen, Q., Zhang, C., Sun, L., Von Krannichfeldt, L., and Wang, Y. Benchmarks and custom package for electrical load forecasting. *arXiv preprint arXiv:2307.07191*, 2023b.

Wikipedia contributors. Moirai — Wikipedia, the free encyclopedia, 2024. URL https://en.wikipedia.org/wiki/Moirai. [Online; accessed 21-January-2024].

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*, 2022. URL https://openreview.net/forum?id=PilZY3omXV2.

Woo, G., Liu, C., Kumar, A., and Sahoo, D. Pushing the limits of pre-training for time series forecasting in

the cloudops domain. *arXiv preprint arXiv:2310.05063*, 2023.

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.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pp. 10524–10533. PMLR, 2020.

Yang, G., Hu, E. J., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022a.

Yang, J., Gupta, A., Upadhyay, S., He, L., Goel, R., and Paul, S. TableFormer: Robust transformer modeling for table-text encoding. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 528–537, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.40. URL https://aclanthology.org/2022.acl-long.40.

Yu, H.-F., Rao, N., and Dhillon, I. S. Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in neural information processing systems*, 29, 2016.

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.

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, B. and Sennrich, R. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhang, K., Wen, Q., Zhang, C., Cai, R., Jin, M., Liu, Y., Zhang, J., Liang, Y., Pang, G., Song, D., et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *arXiv preprint arXiv:2306.10125*, 2023.

Zhang, Y. and Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eie.

Zheng, Y., Yi, X., Li, M., Li, R., Shan, Z., Chang, E., and Li, T. Forecasting fine-grained air quality based on big data. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2267–2276, 2015.

Zhou, J., Lu, X., Xiao, Y., Su, J., Lyu, J., Ma, Y., and Dou, D. Sdwpf: A dataset for spatial dynamic wind power forecasting challenge at kdd cup 2022. *arXiv preprint arXiv:2208.04360*, 2022a.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proc. 39th International Conference on Machine Learning (ICML 2022)*, 2022b.

Zhou, T., Niu, P., Wang, X., Sun, L., and Jin, R. One fits all: Power general time series analysis by pretrained LM. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=gMS6FVZvmF.

A. Large-scale Open Time Series Archive

LOTSA is a collection of time series datasets curated for pre-training of LTMs. In the following, we describe its constituent datasets and their respective sources, listing any pre-processing and data splitting performed. We further details on the key properties of these datasets, providing the domain, frequency, number of time series, number of target variates, number of past covariates (covariates whose values in the forecast horizon are unknown), and total number of observations in the dataset (defined as $\sum_{i=1}^{N} T_i$ for a dataset with N time series). Of note, if we consider number of observations to include the number of variates, i.e. $\sum_{i=1}^{N} T_i * d_{y_i}$, LOTSA would have 231,082,956,489 (231B) total observations.

Table 8. Datasets and key properties from BuildingsBench.

Dataset	Domain	Frequency	# Time Series	# Targets	# Past Covariates	# Obs.
BDG-2 Panther	Energy	H	105	1	0	919,800
BDG-2 Fox	Energy	H	135	1	0	2,324,568
BDG-2 Rat	Energy	H	280	1	0	4,728,288
BDG-2 Bear	Energy	H	91	1	0	1,482,312
Low Carbon London	Energy	H	713	1	0	9,543,348
SMART	Energy	H	5	1	0	95,709
IDEAL	Energy	H	219	1	0	1,265,672
Sceaux	Energy	H	1	1	0	34,223
Borealis	Energy	H	15	1	0	83,269
Buildings900K	Energy	H	1,792,328	1	0	15,702,590,000

BuildingsBench BuildingsBench (Emami et al., 2023) (Table 8) provides a collection of datasets for residential and commercial building energy consumption. The BDG-2 datasets, Low Carbon London, SMART, IDEAL, Sceaux, and Borealis are real building energy consumption datasets from various sources. The Electricity dataset (Trindade, 2015) is also included in BuildingsBench but we omit it from LOTSA and use it for out-of-distribution evaluation instead. They further release the Buildings-900K dataset a large-scale dataset of 900K simulated buildings. Emami et al. (2023) introduce a pre-train/test split based on Public Use Microdata Area, but we use include both splits in LOTSA for pre-training. No special pre-processing was applied to these datasets. More information regarding these datasets can be found in Emami et al. (2023).

Table 9. Datasets and key properties from ClimateLearn.

Dataset	Domain	Frequency	# Time Series	# Targets	# Past Covariates	# Obs.
CMIP6	Climate	6H	1,351,680	53	0	1,973,453,000
ERA5	Climate	H	245,760	45	0	2,146,959,000

ClimateLearn We include the ERA5 and CMIP6 datasets from the ClimateLearn library (Nguyen et al., 2023) (Table 9). The ERA5 and CMIP6 datasets provide time series of various climate related variables such as temperature, and humidity and various pressure levels, based on a grid structure. We use the $2.8125°$ resolution which contains time series in a 64×128 grid.

Table 10. Datasets and key properties from CloudOps TSF

Dataset	Domain	Frequency	# Time Series	# Targets	# Past Covariates	# Obs.
Azure VM Traces 2017	CloudOps	5T	159,472	1	2	885,522,908
Borg Cluster Data 2011	CloudOps	5T	143,386	2	5	537,552,854
Alibaba Cluster Trace 2018	CloudOps	5T	58,409	2	6	95,192,530

CloudOps TSF Woo et al. (2023) introduces three large-scale CloudOps time series datasets (Table 10) measuring various variables such as CPU and memory utilization. We follow their pre-train/test split and only include the pre-train time series in LOTSA, holding out the test set.

Table 11. Datasets and key properties from the GluonTS library.

Dataset	Domain	Frequency	# Time Series	# Targets	# Past Covariates	# Obs.
Taxi	Transport	30T	67,984	1	0	54,999,060
Uber TLC Daily	Transport	D	262	1	0	47,087
Uber TLC Hourly	Transport	H	262	1	0	1,129,444
Wiki-Rolling	Web	D	47,675	1	0	40,619,100
M5	Sales	D	30,490	1	0	58,327,370

GluonTS The GluonTS library (Alexandrov et al., 2020) provides many datasets popularly used in time series forecasting. We only include the datasets presented in Table 11 as we either hold out the other datasets, or are already included in the Monash repository.

Table 12. Key properties of the LargeST Benchmark Dataset.

Dataset	Domain	Frequency	# Time Series	# Targets	# Past Covariates	# Obs.
LargeST	Transport	5T	42,333	1	0	4,452,510,528

LargeST LargeST (Liu et al., 2023a) (Table 12) collects the largest dataset from the California Department of Transportation Performance Measurement System (PeMS) (Chen et al., 2001) to date. The PeMS is a popular source of data for many traffic forecasting datasets such as PEMS03, PEMS04, PEMS07, PEMS08, and PEMS Bay, as well as the popular Traffic dataset from (Lai et al., 2018). Since we use such a large amount of dataset from the same source, we can no longer consider forecasting on any of these datasets as an out-of-distribution or zero-shot forecasting task anymore, and thus omit the Traffic dataset of the LSF benchmark from our evaluations.

Table 13. Datasets and key properties from the LibCity library.

Dataset	Domain	Frequency	# Time Series	# Targets	# Past Covariates	# Obs.
PEMS03	Transport	5T	358	1	0	9,382,464
PEMS04	Transport	5T	307	3	0	5,216,544
PEMS07	Transport	5T	883	1	0	24,921,792
PEMS08	Transport	5T	170	3	0	3,035,520
PEMS Bay	Transport	5T	325	1	0	16,937,700
Los-Loop	Transport	5T	207	1	0	7,094,304
Loop Seattle	Transport	5T	323	1	0	33,953,760
SZ-Taxi	Transport	15T	156	1	0	464,256
Beijing Subway	Transport	30T	276	2	11	248,400
SHMetro	Transport	15T	288	2	0	1,934,208
HZMetro	Transport	15T	80	2	0	146,000
Rotterdam	Transport	2T	208	1	0	4,813,536
Q-Traffic	Transport	15T	45,148	1	0	264,386,688

LibCity LibCity (Wang et al., 2023a) (Table 13) provides a collection urban spatio-temporal datasets. We drop the spatial aspect of these datsets and consider them as time series data.

Monash The Monash Time Series Forecasting Repository (Godahewa et al., 2021) (Table 14) is a large collection of diverse time series datasets, the most popular source for building LTMs. We use Monash for in-distribution evaluation, and thus apart from the exceptions listed below, we only include the train region in `LOTSA`, by holding out the final forecast horizon as the test set. The final forecast horizon is defined for each dataset by (Godahewa et al., 2021). For the following datasets, we include their entirety in `LOTSA` without a held-out test set for the following reasons:

Table 14. Datasets and key properties from the Monash Time Series Forecasting Repository.

Dataset	Domain	Frequency	# Time Series	# Targets	# Past Covariates	# Obs.
London Smart Meters	Energy	30T	5,520	1	0	166,238,880
Wind Farms	Energy	T	337	1	0	172,165,370
Wind Power	Energy	4S	1	1	0	7,397,147
Solar Power	Energy	4S	1	1	0	7,397,222
Oikolab Weather	Climate	H	8	1	0	800,456
Elecdemand	Energy	30T	1	1	0	17,520
Covid Mobility	Transport	D	362	1	0	148,602
Kaggle Web Traffic Weekly	Web	W	145,063	1	0	16,537,182
Extended Web Traffic	Web	D	145,063	1	0	370,926,091
M1 Yearly	Econ/Fin	Y	106	1	0	3,136
M1 Quarterly	Econ/Fin	Q	198	1	0	9,854
M1 Monthly	Econ/Fin	M	617	1	0	44,892
M3 Yearly	Econ/Fin	Y	645	1	0	18,319
M3 Quarterly	Econ/Fin	Q	756	1	0	37,004
M3 Monthly	Econ/Fin	M	1,428	1	0	141,858
M3 Other	Econ/Fin	Q	174	1	0	11,933
M4 Yearly	Econ/Fin	Y	22,739	1	0	840,644
M4 Quarterly	Econ/Fin	Q	24,000	1	0	2,214,108
M4 Monthly	Econ/Fin	M	48,000	1	0	10,382,411
M4 Weekly	Econ/Fin	W	359	1	0	366,912
M4 Hourly	Econ/Fin	H	414	1	0	353,500
M4 Daily	Econ/Fin	D	4,227	1	0	9,964,658
NN5 Daily	Econ/Fin	D	111	1	0	81,585
NN5 Weekly	Econ/Fin	W	111	1	0	11,655
Tourism Yearly	Econ/Fin	Y	419	1	0	11,198
Tourism Quarterly	Econ/Fin	Q	427	1	0	39,128
Tourism Monthly	Econ/Fin	M	366	1	0	100,496
CIF 2016	Econ/Fin	M	72	1	0	6,334
Traffic Weekly	Transport	W	862	1	0	82,752
Traffic Hourly	Transport	H	862	1	0	14,978,112
Australian Electricity Demand	Energy	30T	5	1	0	1,153,584
Rideshare	Transport	H	2,304	1	0	859,392
Saugeen	Nature	D	1	1	0	23,711
Sunspot	Nature	D	1	1	0	73,894
Temperature Rain	Nature	D	32,072	1	0	22,290,040
Vehicle Trips	Transport	D	329	1	0	32,512
Weather	Climate	D	3,010	1	0	42,941,700
Car Parts	Sales	M	2,674	1	0	104,286
FRED MD	Econ/Fin	M	107	1	0	76,612
Pedestrian Counts	Transport	H	66	1	0	3,130,762
Hospital	Healthcare	M	767	1	0	55,224
COVID Deaths	Healthcare	D	266	1	0	48,412
KDD Cup 2018	Energy	H	270	1	0	2,897,004
Bitcoin	Econ/Fin	D	18	1	0	74,824
US Births	Healthcare	D	1	1	0	7,275

- London Smart Meters, Wind Farms, Wind Power, Solar Power, Oikolab Weather, Covid Mobility: Originally not included in the Monash benchmark.
- Extended Web Traffic, Kaggle Web Traffic Weekly: We include the extended version of Web Traffic which contains overlap with the original Web Traffic dataset.
- M1 Yearly, M1 Quarterly, M3 Yearly, M3 Quarterly, M4 Yearly, M4 Quarterly, Tourism Yearly: Some time series in these datasets are too short after train/test splits, thus we do not split them (setting a minimum of 16 time steps to achieve at least 2 patches).

We omit Electricity (Trindade, 2015) and Solar (Lai et al., 2018) datasets for out-of-distribution evaluation. Note that the "Weather" from Monash is a different dataset from that used in the out-of-distribution evaluations.

Table 15. Datasets and key properties from the ProEnFo library.

Dataset	Domain	Frequency	# Time Series	# Targets	# Past Covariates	# Obs.
Covid19 Energy	Energy	H	1	1	6	31,912
GEF12	Energy	H	20	1	1	788,280
GEF14	Energy	H	1	1	1	17,520
GEF17	Energy	H	8	1	1	140,352
PDB	Energy	H	1	1	1	17,520
Spanish	Energy	H	1	1	1	35,064
BDG-2 Hog	Energy	H	24	1	5	421,056
BDG-2 Bull	Energy	H	41	1	3	719,304
BDG-2 Cockatoo	Energy	H	1	1	5	17,544
ELF	Energy	H	1	1	0	21,792

ProEnFo ProEnFo (Wang et al., 2023b) (Table 15) provides a range of datasets for load forecasting. Unlike Buildings-Bench, ProEnFo provides various covariates such as temperature, humidity, and wind speed. We again omit Electricity (Trindade, 2015) which is originally included in ProEnFo for out-of-distribution evaluations.

Table 16. Datasets and key properties from the SubseasonalClimateUSA library.

Dataset	Domain	Frequency	# Time Series	# Targets	# Past Covariates	# Obs.
Subseasonal	Climate	D	862	4	0	14,097,148
Subseasonal Precipitation	Climate	D	862	1	0	9,760,426

SubseasonalClimateUSA The SubseasonalClimateUSA library (Mouatadid et al., 2023) (Table 16) provides climate time series data at a lower granularity (daily) for subseasonal level forecasting. We extract two datasets Subseasonal Precipitation which is the precipitation data from 1948 - 1978, and Subseasonal, which is precipitation and temperature data from 1979 - 2023.

Others Finally, detailed in Table 17, we further collect datasets from miscellaneous sources not provided by a library or collection. These datasets require more extensive pre-processing since they are not provided by a library, and are raw data instead. We take a standard approach of filtering out time series which are either too short, or have too many missing values. Fo each time series, we consider all variates to be targets, unless otherwise specified by the creators of the dataset (e.g. KDD Cup 2022 is a competition dataset, for which only the "Patv" variate is defined to be the target, with 9 other covariates).

Table 17. Datasets and key properties from other miscellaneous sources.

Dataset	Source	Domain	Frequency	# Time Series	# Targets	# Past Covariates	# Obs.
KDD Cup 2022	Zhou et al. (2022a)	Energy	10T	134	1	9	4,727,519
GoDaddy	Kaggle	Econ/Fin	M	3,135	2	0	128,535
Favorita Sales	Kaggle	Sales	D	111,840	1	0	139,179,538
Favorita Transactions	Kaggle	Sales	D	54	1	0	84,408
Restaurant	Kaggle	Sales	D	216	1	0	76,573
Hierarchical Sales	Mancuso et al. (2021)	Sales	D	118	1	0	212,164
China Air Quality	Zheng et al. (2015)	Nature	H	437	6	0	5,739,234
Beijing Air Quality	Chen (2019)	Nature	H	12	11	0	420,768
Residential Load Power	Bergmeir et al. (2023)	Energy	T	271	3	0	145,994,559
Residential PV Power	Bergmeir et al. (2023)	Energy	T	233	3	0	125,338,950
CDC Fluview ILINet	CDC (2017)	Healthcare	W	75	5	0	63,903
CDC Fluview WHO NREVSS	CDC (2017)	Healthcare	W	74	4	0	41,760
Project Tycho	van Panhuis et al. (2018)	Healthcare	W	1,258	1	0	1,377,707

B. MOIRAI Architecture Details

B.1. Multi Patch Size Projection Layers

Each multi patch size projection is a simple Linear layer, for input projections, mapping patch size to hidden state, and for output projections, mapping hidden state to distribution parameters. In practice, we pre-define the frequency to patch size mapping heuristically, selecting smaller patch sizes for low frequency data and larger patch sizes for high frequency data as follows:

- Yearly, Quarterly: 8
- Monthly: 8, 16, 32
- Weekly, Daily: 16, 32
- Hourly: 32, 64
- Minute-level: 32, 64, 128
- Second-level: 64, 128

Note that we only learn one Linear layer per patch size, and share them across frequencies if there is overlap. This means that we learn five input projection layers and five output projection layers.

B.2. Mixture Distribution

As described in Salinas et al. (2020), our model predicts the parameters of a probability distribution, in this case, a mixture distribution. We apply a softmax layer to the parameters associated to the mixture weights, constraining them to the probability simplex. The mixture components are as described.

Student's t-distribution A random variable x following the Student's t-distribution has p.d.f.:

$$p(x;\nu,\mu,\tau) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\tau}\left(1+\frac{1}{\nu}\left(\frac{x-\mu}{\tau}\right)^2\right)^{-(\nu+1)/2}$$

with parameters $\nu > 0, \mu \in \mathbb{R}, \tau > 0$, the degrees-of-freedom (df), location, and scale parameters respectively, and Γ is the gamma function. We predict the df, location, and scale parameters, and apply a softplus function for the positivity constraint. We further lower bound the df parameter to 2, since variance is undefined otherwise.

Log-normal distribution A random variable x which follows a log-normal distribution has p.d.f.:

$$p(x;\mu,\sigma) = \frac{1}{x\sigma\sqrt{2\pi}}\exp\left(-\frac{(\ln x-\mu)^2}{2\sigma^2}\right)$$

with parameters $\mu \in \mathbb{R}, \sigma > 0$. We predict both parameters, applying softplus function for positivity.

Negative binomial distribution Following Awasthi et al. (2022), we implement a continuous extension of the negative binomial distribution. A random variable x following such a distribution has p.d.f.:

$$p(x; r, p) \propto \frac{\Gamma(x+r)}{\Gamma(x+1)\Gamma(r)}(1-p)^r p^x$$

given parameters $r > 0$ and $p \in [0, 1]$, and Γ is the gamma function. We predict both parameters, applying the softplus function for positivity, and sigmoid function to constrain to a probability.

Low variance normal distribution A random variable x following a normal distribution has p.d.f.:

$$p(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

where $\mu \in \mathbb{R}, \sigma > 0$. For a low variance normal distribution, we only predict μ, and fix σ to a small number, in this case we fix $\sigma = 1\text{e-}3$.

B.3. Discussion on "Flexible Distribution"

Table 1 categorizes various pre-trained forecasting models with the notion of a "flexible distribution" – this is largely a qualitative categorization rather than a quantitative one. As of writing, only 3 other models considered probabilistic forecasting – Lag-llama, TimeGPT, and LLMTime. The other models only considered point forecasts, and thus the concept of "flexible distribution" does not apply to them. The following are specific reasons on why we made the categorization for the 3 models which can handle probabilistic forecasting:

- Lag-llama uses a Student-T distribution which is only able to model symmetric distributions. This is an inflexible distribution which is unable to model asymmetric distributions, as demonstrated in Figure 4 of our paper. They also raise this issue in their paper (Section 4.3), citing the use of more expressive distribution heads such as normalizing flows and copulas in future work.
- TimeGPT uses conformal prediction to construct prediction intervals. We refer to a tweet[2] from the creators, which claim: "Some prediction intervals don't account for domain constraints. A few users highlighted intervals suggesting negative values for time series that only take positive values." Thus, we consider it to be inflexible.
- LLMTime uses a categorical distribution. In their paper (paragraph titled "Language models as flexible distributions" in Section 3), they demonstrated that this approach is a flexible distribution which can approximate many different kinds of continuous distributions.

[2] https://twitter.com/nixtlainc/status/1694466983927153131

C. Probabilistic Forecasting

C.1. Evaluation Metrics

Continuous Ranked Probability Score The CRPS (Gneiting & Raftery, 2007) is a probabilistic forecasting evaluation metric, given a predicted distribution with c.d.f. F and ground truth y, it is defined as:

$$\text{CRPS} = \int_0^1 2\Lambda_\alpha(F^{-1}(\alpha), y)d\alpha$$
$$\Lambda_\alpha(q, y) = (\alpha - \mathbf{1}_{\text{y<q}})(y - q),$$

where Λ_α is the α-quantile loss, also known as the pinball loss at quantile level α.

In practice, the CRPS is intractable or computationally expensive to compute, and we also want to compute a normalized metric, thus we compute a normalized discrete approximation, the mean weighted sum quantile loss (Park et al., 2022), defined as the average of K quantiles:

$$\text{CRPS} \approx \frac{1}{K}\sum_{k=1}^{K} \text{wQL}[\alpha_k]$$
$$\text{wQL}[\alpha] = 2\frac{\sum_t \Lambda_\alpha(\hat{q}_t(\alpha), y_t)}{\sum_t |y_t|},$$

where $\hat{q}_t(\alpha)$ is the predicted α-quantile at time step t. We take $K = 9, \alpha_1 = 0.1, \alpha_2 = 0.2, \ldots, \alpha_9 = 0.9$ in practice.

Mean Scaled Interval Score The MSIS is a metric to evaluate uncertainty around point forecasts, introduced in the M4 Competition (Makridakis et al., 2020). Given an upper bound prediction, U_t, and lower bound prediction L_t, the MSIS is defined as:

$$\text{MSIS} = \frac{1}{h}\frac{\sum_{t=1}^{h}(U_t - L_t) + \frac{2}{a}(L_t - Y_t)\mathbb{1}_{\{Y_t<L_t\}} + \frac{2}{a}(Y_t - U_t)\mathbb{1}_{\{Y_t>U_t\}}}{\frac{1}{n-m}\sum_{t=m+1}^{n}|Y_t - Y_{t-m}|}$$

where $a = 0.05$ is the significance level for a 95% prediction interval, over a forecast horizon of length h, and m is the seasonal factor.

C.2. Evaluation Setup

Table 18. Summary of datasets used in the out-of-distribution probabilistic forecasting evaluation setting.

Dataset	Domain	Frequency	Prediction Length	Rolling Evaluations
Electricity (Trindade, 2015)	Energy	H	24	7
Solar (Lai et al., 2018)	Energy	H	24	7
Walmart (Walmart Competition Admin, 2014)	Sales	W	8	4
Weather	Climate	10T	144	7
Istanbul Traffic [3]	Transport	H	24	7
Turkey Power [4]	Energy	H	24	7

We perform evaluation in a non-overlapping rolling window fashion, i.e. stride is equal to prediction length. The test set is defined as the last $h * r$ time steps where h is the prediction length of the forecast horizon, and r is the number of rolling evaluation windows. We take the validation set to be the last forecast horizon before the test set, and the train set to be everything before that. From Table 18, our evaluation spans four domains, from minute-level to weekly frequencies. Prediction length and rolling evaluations are defined for each dataset based on frequency, making day ahead predictions for sub-hourly frequencies for seven days, and eight week ahead predictions for 32 weeks for weekly frequency.

[3] https://www.kaggle.com/datasets/leonardo00/istanbul-traffic-index

[4] https://www.kaggle.com/datasets/dharanikra/electrical-power-demand-in-turkey

C.3. Baselines

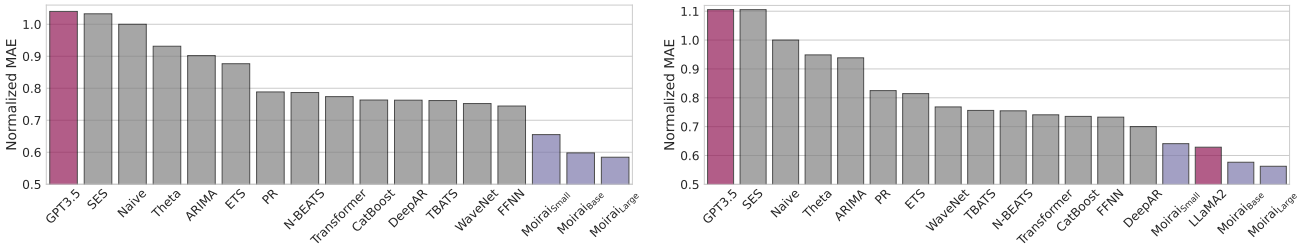
Table 19. Hyperparameter search values for probabilistic forecasting evaluation baselines.

	Hyperparameter	Values
PatchTST	d_model	$\{64, 128, 256\}$
	num_encoder_layers	$[2, 6]$
DeepAR	hidden_size	$\{64, 128, 256\}$
	num_layers	$[2, 6]$
TFT	hidden_dim	$\{64, 128, 256\}$
TiDE	hidden_dim	$\{64, 128, 256\}$
	num_encoder_layers = num_decoder_layers	$[2, 6]$

For the four deep learning baselines, DeepAR (Salinas et al., 2020), PatchTST (Nie et al., 2023), TiDE (Das et al., 2023a), and TFT (Lim et al., 2021), we perform hyperparameter tuning based on the values presented in Table 19, and also tune learning rate $[1\text{e-}6, 1\text{e-}3]$ in log scale, and the context length as $l = m * h$, where m is tuned in the range $[2, 20]$, and h is the prediction length. We perform random search through these values over 15 training runs, and report results on 5 independent training runs based on the best validation CRPS.

D. Full Experimental Results

D.1. In-distribution Forecasting: Monash Time Series Forecasting Benchmark



(a) Results aggregated over full all datasets in Table 20.

(b) Results aggregated over LLaMA2 subset in Table 20.

Figure 7. Extended aggregate results of the Monash Time Series Forecasting Benchmark as per Figure 3. GPT3.5 is our reproduction of LLMTime based on the GPT3.5 API, whereas LLaMA2 is based on the results reported in Gruver et al. (2023).

Table 20. Full results of Monash Time Series Forecasting Benchmark. MAE is reported.

	MOIRAI$_{\text{Small}}$	MOIRAI$_{\text{Base}}$	MOIRAI$_{\text{Large}}$	Naive	SES	Theta	TBATS	ETS	(DHR-)ARIMA	PR	CatBoost	FFNN	DeepAR	N-BEATS	WaveNet	Transformer	GPT3.5	LLaMA2
M1 Monthly	2,082.26	2,068.63	1,983.18	2,707.75	2,259.04	2,166.18	2,237.50	1,905.28	2,080.13	2,088.25	2,052.32	2,162.58	1,860.81	1,820.37	2,184.42	2,723.88	2562.84	-
M3 Monthly	713.41	658.17	664.03	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	877.97	-
M3 Other	263.54	198.62	202.41	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	300.30	-
M4 Monthly	597.6	592.09	584.36	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	728.27	-
M4 Weekly	339.76	328.08	301.52	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	518.44	-
M4 Daily	189.1	192.66	189.78	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	266.52	-
M4 Hourly	268.04	209.87	197.79	1,218.06	1,218.06	1,220.97	386.27	3,358.10	1,310.85	257.39	285.35	385.49	886.02	425.75	393.63	320.54	576.06	-
Tourism Quarterly	18,352.44	17,196.86	15,820.02	15,845.10	15,014.19	7,656.49	9,972.42	8,925.52	10,475.47	9,092.58	10,267.97	8,981.04	9,511.37	8,640.56	9,137.12	9,521.67	16918.86	9311.98
Tourism Monthly	3,569.85	2,862.06	2,688.55	5,636.83	5,302.10	2,069.96	2,940.08	2,004.51	2,536.77	2,187.28	2,537.04	2,022.21	1,871.69	2,003.02	2,095.13	2,146.98	5608.61	3145.48
CIF 2016	655,888.58	539,222.03	695,156.92	578,596.53	581,875.97	714,818.58	855,578.40	642,421.42	469,059.49	563,205.57	603,551.30	1,495,923.44	3,200,418.00	679,034.80	5,998,224.62	4,057,973.00	599313.84	684057.87
Aus. Elec. Demand	266.57	201.39	177.68	659.6	659.6	665.04	370.74	1,282.99	1,045.92	247.18	241.77	258.76	302.41	213.83	227.5	231.45	760.81	560.48
Bitcoin	1.76E+18	1.62E+18	1.87E+18	7.78E+17	5.33E+18	5.33E+18	9.90E+17	1.10E+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	1.74E+18	8.57E+17
Pedestrian Counts	54.88	54.08	41.66	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	97.77	65.92
Vehicle Trips	24.46	23.17	21.85	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	31.48	-
KDD cup	39.81	38.66	39.09	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	42.72	-
Weather	1.96	1.8	1.75	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	2.17	2.09
NN5 Daily	5.37	4.26	3.77	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	7.10	6.67
NN5 Weekly	15.07	16.42	15.3	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	15.76	15.60
Carparts	0.53	0.47	0.49	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	0.44	-
FRED-MD	2,568.48	2,679.29	2,792.55	2,825.67	2,798.22	3,492.84	1,989.97	2,041.42	2,957.11	8,921.94	2,475.68	2,339.57	4,264.36	2,557.80	2,508.40	4,666.04	2804.64	1781.41
Traffic Hourly	0.02	0.02	0.01	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	0.03	0.02
Traffic Weekly	1.17	1.14	1.13	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	1.15	1.15
Rideshare	1.35	1.39	1.29	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	6.28	-
Hospital	23	19.4	19.44	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	25.68	22.75
COVID Deaths	124.32	126.11	117.11	353.71	353.71	321.32	96.29	85.59	85.77	347.98	475.15	144.14	201.98	158.81	1,049.48	408.66	653.31	66.14
Temperature Rain	5.3	5.08	5.27	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	6.37	-
Sunspot	0.11	0.08	0.13	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	5.07	0.28
Saugeen River Flow	24.07	24.4	24.76	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	34.84	23.01
US Births	872.51	624.3	476.5	1,152.67	1,192.20	586.93	399	419.73	526.33	574.93	441.7	557.87	424.93	422	504.4	452.87	1374.99	638.82

We include the full breakdown of results for the Monash benchmark in Table 20, including two versions of LLMTime: GPT3.5 (our reproduction), and LLaMA2 (results from Gruver et al. (2023)). GPT3.5 is our reproduction by running their code[5] on the full dataset, using GPT3.5-Turbo-Instruct since text-davinci-003 has been deprecated. LLaMA2 only has results for a subset of datasets in Table 20, thus in Figure 7, we present two aggregated results, one aggregated on the full Table 20, and one on the subset with results available for LLaMA2. As observed, MOIRAI retains the top rankings for with the base and large models in all settings.

D.2. Out-of-distribution Forecasting: Probabilistic Forecasting

Table 21 provides the full results of the probabilistic forecasting experiments with additional point forecasting metrics, including the symmetric mean absolute percentage error (sMAPE) (Hyndman, 2014), mean absolute scaled error (MASE) (Hyndman & Koehler, 2006), normalized deviation (ND), and normalized root mean squared error (NRMSE) (Yu et al., 2016).

D.3. Out-of-distribution Forecasting: Long Sequence Forecasting

Table 22 provides the full breakdown of results for the long sequence forecasting experiments, listing results for each prediction length.

[5] https://github.com/ngruver/llmtime

Table 21. Full results for probabilistic forecasting experiments. Best results are highlighted in **bold**, and second best results are underlined.

		Zero-shot			Full-shot				Baseline	
		MOIRAI$_{\text{Small}}$	MOIRAI$_{\text{Base}}$	MOIRAI$_{\text{Large}}$	PatchTST	TiDE	TFT	DeepAR	AutoARIMA	Seasonal Naive
Electricity	**CRPS**	0.072	0.055	0.050	0.052±0.00	**0.048±0.00**	0.050±0.00	0.065±0.01	0.327	0.070
	MSIS	7.999	6.172	5.875	5.744±0.12	**5.672±0.08**	6.278±0.24	6.893±0.82	29.412	35.251
	sMAPE	0.134	0.111	0.106	0.107±0.00	**0.102±0.00**	0.106±0.01	0.118±0.02	0.318	0.108
	MASE	0.981	0.792	0.751	0.753±0.01	**0.706±0.02**	0.747±0.03	0.844±0.16	3.229	0.881
	ND	0.092	0.069	0.063	0.065±0.00	**0.061±0.00**	0.063±0.00	0.080±0.02	0.357	0.070
	NRMSE	0.840	0.551	**0.465**	0.506±0.02	0.514±0.02	0.511±0.02	0.704±0.11	3.296	0.478
Solar	**CRPS**	0.471	0.419	**0.406**	0.518±0.09	0.420±0.00	0.446±0.03	0.431±0.01	1.055	0.512
	MSIS	8.425	7.011	**6.250**	8.447±1.59	13.754±0.32	8.057±3.51	11.181±0.67	25.849	48.130
	sMAPE	1.445	1.410	1.400	1.501±0.10	1.400±0.00	1.391±0.01	1.385±0.00	1.685	**0.691**
	MASE	1.465	1.292	1.237	1.607±0.25	1.265±0.02	1.399±0.11	1.222±0.01	2.583	**1.203**
	ND	0.624	0.551	0.528	0.685±0.11	0.538±0.01	0.594±0.05	0.520±0.00	1.098	**0.512**
	NRMSE	1.135	1.034	**1.014**	1.408±0.26	1.093±0.00	1.236±0.06	1.033±0.01	1.784	1.168
Walmart	**CRPS**	0.103	0.093	0.098	0.082±0.01	**0.077±0.00**	0.087±0.00	0.121±0.00	0.124	0.151
	MSIS	9.371	8.421	8.520	**6.005±0.21**	6.258±0.12	8.718±0.10	12.502±0.03	9.888	49.458
	sMAPE	0.179	0.168	0.174	0.150±0.01	**0.145±0.00**	0.172±0.00	0.216±0.00	0.219	0.205
	MASE	1.048	0.964	1.007	0.867±0.09	**0.814±0.01**	0.948±0.02	1.193±0.02	1.131	1.236
	ND	0.129	0.117	0.124	0.105±0.01	**0.097±0.00**	0.108±0.00	0.147±0.00	0.141	0.151
	NRMSE	0.324	0.291	0.332	0.218±0.02	**0.204±0.00**	0.235±0.01	0.298±0.00	0.305	0.328
Weather	**CRPS**	0.049	**0.041**	0.051	0.059±0.01	0.054±0.00	0.043±0.00	0.132±0.11	0.252	0.068
	MSIS	5.236	5.136	**4.962**	7.759±0.49	8.095±1.74	7.791±0.44	21.651±17.34	19.805	31.293
	sMAPE	0.686	0.623	0.688	0.668±0.01	0.636±0.01	0.672±0.01	0.776±0.05	0.770	**0.401**
	MASE	0.521	**0.487**	0.515	0.844±0.19	0.832±0.13	0.692±0.02	3.170±3.47	0.938	0.782
	ND	0.063	**0.048**	0.063	0.072±0.01	0.066±0.01	0.051±0.00	0.163±0.15	0.139	0.068
	NRMSE	0.229	0.417	0.331	0.260±0.01	0.214±0.00	**0.211±0.00**	0.486±0.43	0.465	0.290
Istanbul Traffic	**CRPS**	0.173	0.116	0.112	0.112±0.00	0.110±0.01	0.110±0.01	**0.108±0.00**	0.589	0.257
	MSIS	5.937	4.461	4.277	**3.813±0.09**	4.752±0.17	4.057±0.44	4.094±0.31	16.317	45.473
	sMAPE	0.359	0.284	0.288	0.287±0.01	0.280±0.01	0.287±0.01	**0.249±0.01**	1.141	0.391
	MASE	0.990	0.644	0.631	0.653±0.02	0.618±0.03	0.620±0.03	**0.613±0.03**	3.358	1.137
	ND	0.224	0.146	0.143	0.148±0.01	0.140±0.01	0.141±0.01	**0.139±0.01**	0.758	0.257
	NRMSE	0.294	0.194	0.186	0.190±0.01	0.185±0.01	0.185±0.01	**0.181±0.01**	0.959	0.384
Turkey Power	**CRPS**	0.048	0.040	**0.036**	0.054±0.01	0.046±0.01	0.039±0.00	0.066±0.02	0.116	0.085
	MSIS	7.127	6.766	**6.341**	8.978±0.51	8.579±0.52	7.943±0.31	13.520±1.17	14.863	36.256
	sMAPE	0.389	0.378	0.375	0.416±0.01	0.389±0.00	0.383±0.00	0.404±0.01	0.244	**0.125**
	MASE	0.948	0.888	**0.870**	1.234±0.12	0.904±0.02	0.890±0.05	1.395±0.30	1.700	0.906
	ND	0.061	0.051	**0.046**	0.071±0.01	0.059±0.01	0.049±0.00	0.083±0.02	0.150	0.085
	NRMSE	0.149	0.118	**0.102**	0.158±0.01	0.139±0.03	0.104±0.01	0.181±0.05	0.383	0.231

Table 22. Full results of long sequence forecasting experiments. Best results are highlighted in **bold**, and second best results are underlined. Full-shot results are obtained from Liu et al. (2023b).

		Zero-shot						Full-shot																	
		MOIRAI$_{\text{Small}}$		MOIRAI$_{\text{Base}}$		MOIRAI$_{\text{Large}}$		iTransformer		TimesNet		PatchTST		Crossformer		TiDE		DLinear		SCINet		FEDformer	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.375**	0.402	0.384	0.402	0.380	0.398	0.386	0.405	0.384	0.402	0.414	0.419	0.423	0.448	0.479	0.464	0.386	0.400	0.654	0.599	0.376	0.419
	192	**0.399**	**0.419**	0.425	0.429	0.440	0.434	0.441	0.436	0.436	0.429	0.460	0.445	0.471	0.474	0.525	0.492	0.437	0.432	0.719	0.631	0.420	0.448
	336	**0.412**	**0.429**	0.456	0.450	0.514	0.474	0.487	0.458	0.491	0.469	0.501	0.466	0.570	0.546	0.565	0.515	0.481	0.459	0.778	0.659	0.459	0.465
	720	**0.413**	**0.444**	0.470	0.473	0.705	0.568	0.503	0.491	0.521	0.500	0.500	0.488	0.653	0.621	0.594	0.558	0.519	0.516	0.836	0.699	0.506	0.507
ETTh2	96	0.281	0.334	**0.277**	0.327	0.287	0.325	0.297	0.349	0.340	0.374	0.302	0.348	0.745	0.584	0.400	0.440	0.333	0.387	0.707	0.621	0.358	0.397
	192	0.340	0.373	**0.340**	0.374	0.347	0.367	0.380	0.400	0.402	0.414	0.388	0.400	0.877	0.656	0.528	0.509	0.477	0.476	0.860	0.689	0.429	0.439
	336	**0.362**	0.393	0.371	0.401	0.377	0.393	0.428	0.432	0.452	0.541	0.426	0.433	1.043	0.731	0.643	0.571	0.594	0.541	1.000	0.744	0.496	0.487
	720	**0.380**	0.416	0.394	0.426	0.404	0.421	0.427	0.445	0.462	0.657	0.431	0.446	1.104	0.763	0.874	0.679	0.831	0.657	1.249	0.838	0.463	0.474
ETTm1	96	0.404	0.383	0.335	0.360	0.353	0.363	0.334	0.368	0.338	0.375	**0.329**	0.367	0.404	0.426	0.364	0.387	0.345	0.372	0.418	0.438	0.379	0.419
	192	0.435	0.402	**0.366**	0.379	0.376	0.380	0.377	0.391	0.374	0.387	0.367	0.385	0.450	0.451	0.398	0.404	0.380	0.389	0.439	0.450	0.426	0.441
	336	0.462	0.416	**0.391**	0.394	0.399	0.395	0.426	0.420	0.410	0.411	0.399	0.410	0.532	0.515	0.428	0.425	0.413	0.413	0.490	0.485	0.445	0.459
	720	0.490	0.437	0.434	0.419	0.432	**0.417**	0.491	0.459	0.478	0.450	0.454	0.439	0.666	0.589	0.487	0.461	0.474	0.453	0.595	0.550	0.543	0.490
ETTm2	96	0.205	0.282	0.195	0.269	0.189	0.260	0.180	0.264	0.187	0.267	**0.175**	0.259	0.287	0.366	0.207	0.305	0.193	0.292	0.286	0.377	0.203	0.287
	192	0.261	0.318	0.247	0.303	0.247	0.300	0.250	0.309	0.249	0.309	**0.241**	0.302	0.414	0.492	0.290	0.364	0.284	0.362	0.399	0.445	0.269	0.328
	336	0.319	0.355	**0.291**	0.333	0.295	0.334	0.311	0.348	0.321	0.351	0.305	0.343	0.597	0.542	0.377	0.422	0.369	0.427	0.637	0.591	0.325	0.366
	720	0.415	0.410	**0.355**	0.377	0.372	0.386	0.412	0.407	0.408	0.403	0.402	0.400	1.730	1.042	0.558	0.524	0.554	0.522	0.960	0.735	0.421	0.415
Electricity	96	0.205	0.299	0.158	0.248	0.152	0.242	**0.148**	0.240	0.168	0.272	0.195	0.285	0.219	0.314	0.237	0.329	0.197	0.282	0.247	0.345	0.193	0.308
	192	0.220	0.310	0.174	0.263	0.171	0.259	**0.162**	0.253	0.184	0.289	0.199	0.289	0.231	0.322	0.236	0.330	0.196	0.285	0.257	0.355	0.201	0.315
	336	0.236	0.323	0.191	0.278	0.192	0.278	**0.178**	0.269	0.198	0.300	0.215	0.305	0.246	0.337	0.249	0.344	0.209	0.301	0.269	0.369	0.214	0.329
	720	0.270	0.347	0.229	0.307	0.236	0.313	0.225	0.317	**0.220**	0.320	0.256	0.337	0.280	0.363	0.284	0.373	0.245	0.333	0.299	0.390	0.246	0.355
Weather	96	0.173	0.212	0.167	0.203	0.177	0.208	0.174	0.214	0.172	0.220	0.177	0.218	**0.158**	0.230	0.202	0.261	0.196	0.255	0.221	0.306	0.217	0.296
	192	0.216	0.250	0.209	0.241	0.219	0.249	0.221	0.254	0.219	0.261	0.225	0.259	**0.206**	0.277	0.242	0.298	0.237	0.296	0.261	0.340	0.276	0.336
	336	0.260	0.282	**0.256**	0.276	0.277	0.292	0.278	0.296	0.280	0.306	0.278	0.297	0.272	0.335	0.287	0.335	0.283	0.335	0.309	0.378	0.339	0.380
	720	**0.320**	0.322	0.321	0.323	0.365	0.350	0.358	0.349	0.365	0.359	0.354	0.348	0.398	0.418	0.351	0.386	0.345	0.381	0.377	0.427	0.403	0.428

D.4. Computation Costs

Table 23. Computational cost in terms of seconds of various models in terms of seconds for inference for a batch size of 32. "(32)" for MOIRAI refers to patch size.

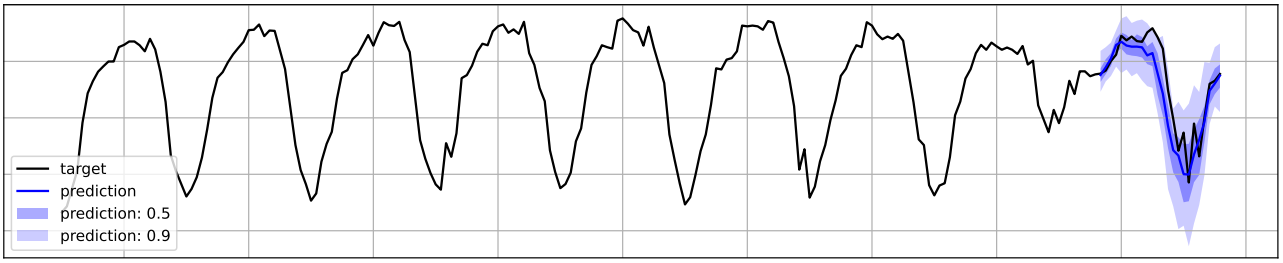
	Context Length					Prediction Length				
	1000	2000	3000	4000	5000	1000	2000	3000	4000	5000
$\text{MOIRAI}_{\text{Small}}$ (32)	0.03	0.04	0.05	0.06	0.07	0.03	0.04	0.05	0.06	0.07
$\text{MOIRAI}_{\text{Base}}$ (32)	0.05	0.06	0.08	0.11	0.13	0.05	0.06	0.08	0.11	0.13
$\text{MOIRAI}_{\text{Large}}$ (32)	0.09	0.14	0.19	0.25	0.3	0.09	0.14	0.19	0.25	0.3
PatchTST	0.01	0.02	0.02	0.03	0.04	0.01	0.01	0.01	0.01	0.02
TiDE	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
TFT	0.02	0.04	0.06	0.08	0.09	0.03	0.07	0.12	0.17	OOM
DeepAR	0.26	0.32	0.37	0.43	0.49	2.02	4.06	6.1	8.17	10.24

We perform an analysis on the computation cost of MOIRAI compared to other deep learning based models, while varying the context and prediction lengths. Overall, given the same model size and setting, the cost of inference compared to other deep learning models would be similar. From an architecture perspective, MOIRAI has the following benefits:

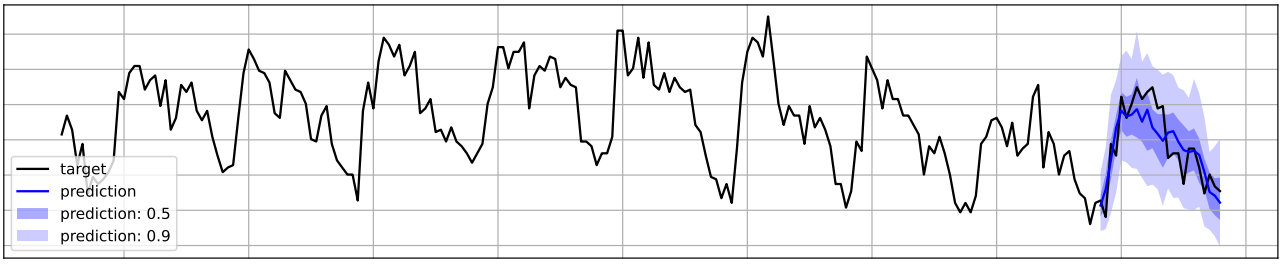
- Patch based inputs: This decreases the computation cost significantly by reducing the number of input tokens.
- Masked encoder architecture: Unlike decoder-only Transformers, the masked encoder architecture can make multi step predictions in a single forward pass. For decoder-only Transformers and RNNs, they need to autoregressively make predictions, making multiple forward passes for a multi step forecast. For long horizons, this can be quite costly.

Furthermore, compared to standard baselines, MOIRAI performs zero-shot forecasting. The standard baseline approach has to be trained (multiple times with hyperparameter tuning) for each dataset, leading to increased costs. As MOIRAI continues to be utilized on new datasets, the pre-training costs are amortized and only becomes cheaper, while standard approaches need to be trained over and over again on new datasets. We note that while MOIRAI indeed incurs increased costs due to model size, inference is still highly competitive, taking under 1 second to construct forecasts even with extremely long context/prediction lengths.

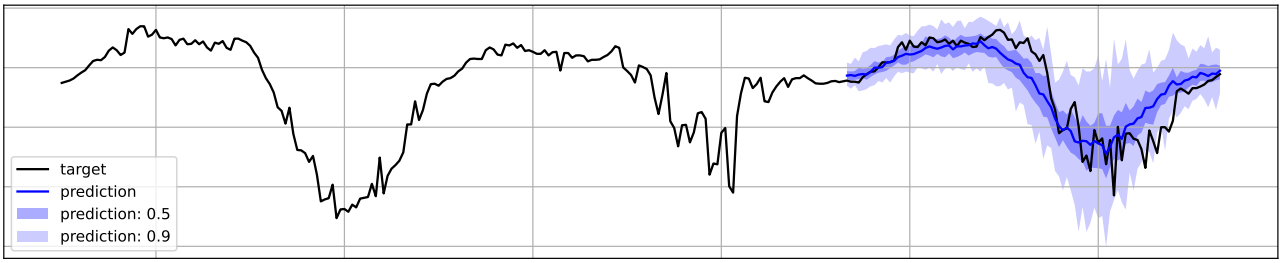
E. Forecast Visualizations



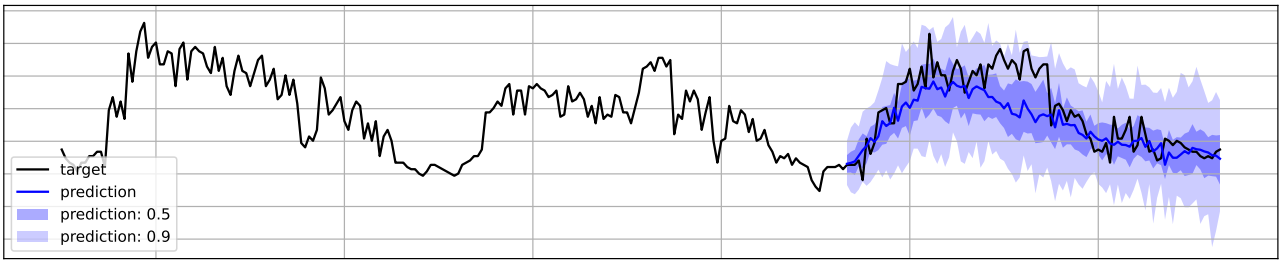
(a) ETTh1-1



(b) ETTh1-2

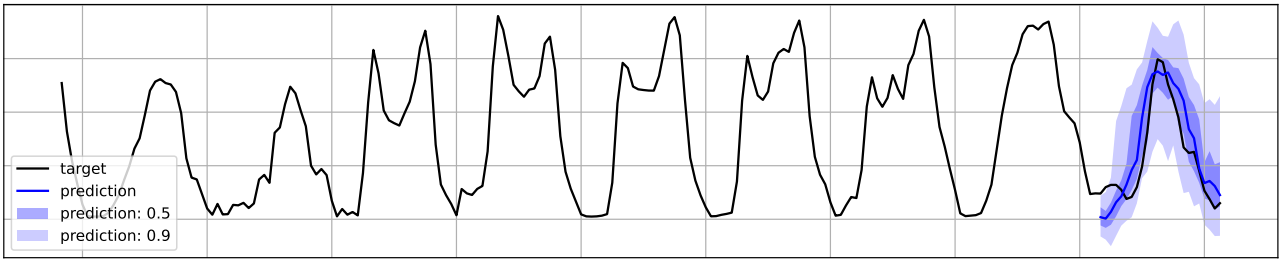


(c) ETTm1-1

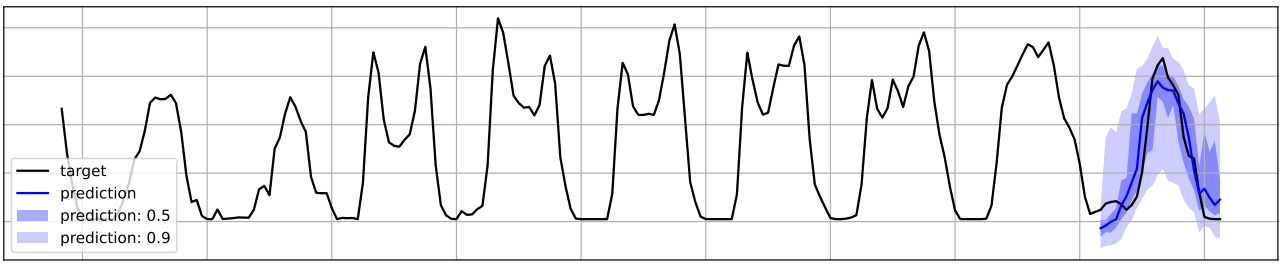


(d) ETTm1-2

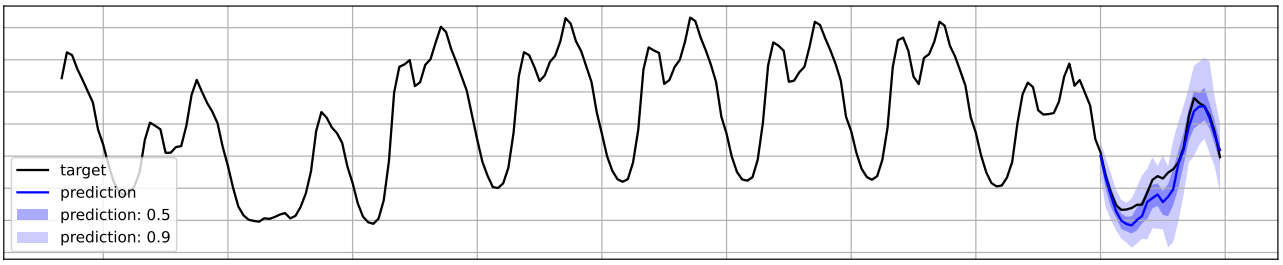
Figure 8. Visualizations of **zero-shot forecasts** from MOIRAI$_{\text{Base}}$ on ETTh1 and ETTm1 datasets.



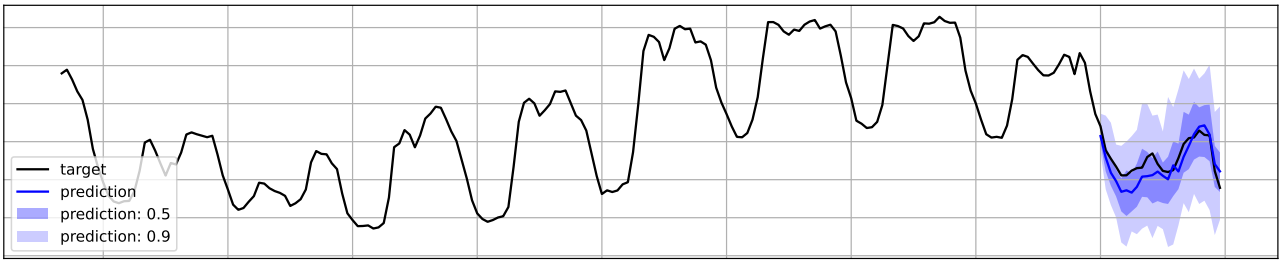
(a) Istanbul Traffic-1



(b) Istanbul Traffic-2



(c) Turkey Power-1



(d) Turkey Power-2

Figure 9. Visualizations of **zero-shot forecasts** from MOIRAI$_{\text{Base}}$ on Istanbul Traffic and Turkey Power datasets.