RETHINKING AUTOREGRESSIVE TIME SERIES PRETRAINING WITH NEXT-x-TOKEN PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive pretraining is a central paradigm for building foundation models, yet its effectiveness for time series foundation models (TSFMs) remains controversial. Conventional next-token prediction (NTP) provides only local supervision and suffers from error accumulation. Existing Next-N-Token Prediction (NNTP) approaches that predict multiple future tokens per step alleviate these issues to some extent, but introduce a fixed-horizon bias that may hinder generalization across diverse forecasting horizons. To address these limitations, we reformulate autoregressive time series pretraining as **Next-x-Token Prediction (NxTP)**, which dynamically samples the prediction horizon for each token based on the available context. By enabling models to directly predict sufficiently long future sequences within a single decoding step, NxTP substantially reduces the need for autoregressive rollouts while improving generalization across forecasting horizons. Extensive experiments on two well-established benchmarks demonstrate that NxTP consistently improves forecasting performance and horizon generalization while substantially accelerating inference in long-term forecasting scenarios.

1 INTRODUCTION

Time series foundation models (TSFMs) have demonstrated strong zero-shot capabilities in time series forecasting and have catalyzed a paradigm shift in the field. Instead of training task-specific models from scratch for each dataset (Zhou et al., 2021; Liu et al., 2024a), a single model can be pretrained on large-scale time series corpus and then applied to a wide range of forecasting tasks, facilitating more adaptable and efficient deployment across diverse real-world scenarios.

The choice of pretraining paradigm is central to the success of foundation models. Autoregressive modeling, which learns to predict future tokens from historical context, has underpinned the remarkable progress of large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023). Its simple and scalable objective naturally supports generative modeling and flexible inference. Motivated by this success and the ordered dependencies of time series, early TSFMs widely adopted autoregressive pretraining (Das et al., 2024; Liu et al., 2024b). However, autoregressive modeling has not reached the same consensus in TSFMs as in language modeling. Recent work, such as Chronos-2 (Ansari et al., 2025) and Zeus (Fu et al., 2026), has instead explored masked reconstruction as an alternative objective. This divergence suggests that directly transferring autoregressive pretraining from language to time series may be insufficient, owing to challenges specific to time series data.

Early TSFMs typically adopt **Next-Token Prediction (NTP)**, predicting only the immediate next token at each decoding step, as shown in Figure 1 (a). While simple and general, NTP provides inherently local supervision, which can bias the model toward short-term fluctuations rather than long-rang dependencies. Moreover, autoregressive rollouts inevitably lead to error accumulation, as prediction errors propagate and compound over successive decoding steps. To alleviate these limitations, recent approaches extend the prediction horizon of each decoding step by predicting multiple future tokens simultaneously (Das et al., 2024; Feng et al., 2025), which we refer to as **Next-N-Token Prediction (NNTP)**, as illustrated in Figure 1 (b). However, it introduces a fundamental trade-off in choosing N. A small N still requires multiple decoding steps and remains susceptible to error accumulation, whereas a large N substantially increases the difficulty of predicting distant futures from limited context. More fundamentally, both NTP and NNTP impose a fixed prediction horizon during pretraining, which can lead to horizon-specific bias and limit the model's capacity

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

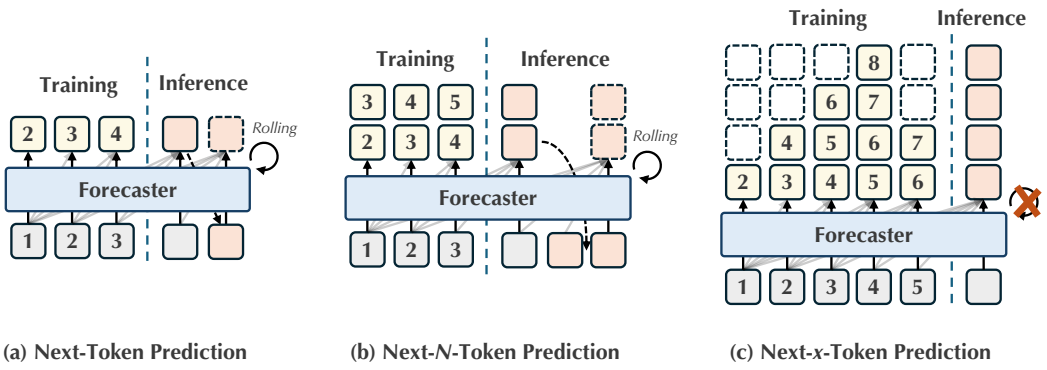

Figure 1: Comparison of autoregressive time series pretraining paradigms.

to generalize across forecasting horizons. Figure 2 provides empirical evidence of these limitations. As the inference horizon of TimesFM increases, the forecasting MAE generally rises, and grows particularly sharply whenever an additional decoding step is required, reflecting error accumulation. Yet the MAE drops periodically at integer multiples of the pretraining prediction length, suggesting that the model is biased toward the horizons seen during pretraining.

To address these issues, we redefine autoregressive learning for time series as **Next-x-Token Prediction (NxTP)**. Rather than specifying a fixed horizon, NxTP dynamically samples the prediction length for each token based on the currently available context, as illustrated in Figure 1 (c). By exposing the model to diverse prediction lengths during training, NxTP avoids bias toward a fixed horizon and improves generalization to arbitrary forecasting horizons. Moreover, NxTP can produce substantially longer outputs in a single decoding step, eliminating error accumulation from autoregressive rollouts in most forecasting scenarios.

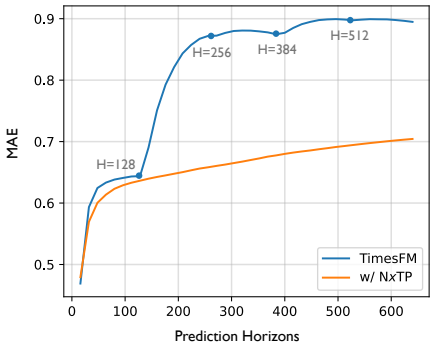

Figure 2: **Validation MAE of across prediction horizons.** TimesFM shows periodic drops at multiples of its pretraining prediction length (128), while TimesFM with NxTP achieves smoother and consistently lower errors.

To evaluate the effectiveness of NxTP, we retrain four representative TSFMs, yielding six variants in total, from scratch. Extensive experiments on two well-established benchmarks demonstrate that NxTP substantially improves forecasting performance and generalization to arbitrary horizons, while also delivering significant gains in inference efficiency. In summary, our key contributions are as follows:

- We introduce Next-x-Token Prediction (NxTP), a novel formulation of autoregressive time series pretraining that dynamically samples the prediction horizon for each token instead of fixing it to a predetermined length.
- NxTP enables one-step decoding for most forecasting horizons, largely avoiding the error accumulation caused by autoregressive rollouts.
- NxTP effectively improves the model's generalization across arbitrary horizons and overall forecasting performance, as validated on two well-established benchmarks.

2 Related Work

2.1 Time Series Foundation Models

Recent advances in time series forecasting have witnessed a paradigm shift from end-to-end modeling toward large-scale pretraining (Liang et al., 2024). Instead of training a separate model for each dataset, an increasing body of work seeks to develop time series foundation models (TSFMs) by pretraining on large-scale and divers time series corpora (Liu et al., 2024b; Ansari et al., 2024).

With a few notable exceptions (Ekambaram et al., 2024; Auer et al., 2026), most recent TSFMs adopt the Transformer architecture (Vaswani et al., 2017) or its variants, leveraging its ability to capture long-term dependencies and model complex temporal patterns. TSFM research has explored a diverse range of architectural choices. *Encoder-only* models, exemplified by early efforts such as MOMENT (Goswami et al., 2024) and Moirai (Woo et al., 2024), as well as more recent models such as Chronos-2 (Ansari et al., 2025), Zeus (Fu et al., 2026), and FalconX (Liu et al., 2026a), leverage bidirectional attention for globally contextualized temporal representations. *Encoder-decoder* models, such as Chronos (Ansari et al., 2024), ChronosBolt, and Kairos (Feng et al., 2025), separate context encoding from future generation. In contrast, *decoder-only* models, such as the Timer family (Liu et al., 2024b) and the TimesFM family (Das et al., 2024), adopt causal architectures with autoregressive generation. This design naturally respects the temporal order of time series, while enabling efficient streaming inference through key-value caching (Pope et al., 2023) and providing a unified pathway for context encoding and future generation. While decoder-only architectures with autoregressive modeling have emerged as the dominant paradigm for LLM pretraining (Naveed et al., 2025), it has yet to reach a consensus on which paradigm is best suited for TSFM.

2.2 Autoregressive Time Series Pretraining

Autoregressive pretraining is a widely adopted training paradigm for TSFMs. Next-Token Prediction (NTP) is the most conventional objective adopted by representative TSFMs such as Timer (Liu et al., 2024b) and Chronos (Ansari et al., 2024), where each decoding step predicts only the immediate next token. To improve rollout efficiency and alleviate error accumulation, recent approaches, including the TimesFM family and Moirai 2.0 (Liu et al., 2025), extend NTP to Next-N-Token Prediction (NNTP), jointly predicting multiple future tokens within a single decoding step. Building on this idea, Time-MoE (Shi et al., 2024) introduces multi-resolution forecasting, jointly training multiple prediction heads at different horizons to support arbitrary prediction lengths while reducing the number of autoregressive steps. More recently, Timer-S1 (Liu et al., 2026b) proposes Serial Token Prediction (STP), which increases the depth of prediction heads to directly generate successive future tokens without autoregressive rollouts. Despite these advances, existing approaches generally rely on fixed prediction horizons, either through a single prediction head or a predefined set of heads, limiting their ability to generalize seamlessly to arbitrary horizons.

3 Methodology

3.1 Preliminaries

Given a time series $\mathbf{y}_{1:T} = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$, where $\mathbf{y}_t \in \mathbb{R}^C$ denotes the observation at time step t across C variables, a time series forecasting task aims to predict its future observations $\mathbf{y}_{T+1:T+H}$ over a forecasting horizon H. In conventional task-specific forecasting, H is typically fixed for each task. In contrast, a time series foundation model (TSFM) f_θ seeks to generalize across diverse datasets and forecasting horizons. Autoregressive pretraining offers a natural paradigm for achieving this goal by learning to generate future observations conditioned on historical context.

Time Series Token To represent time series in TSFMs, the original time series is commonly transformed into a sequence of tokens. Specifically, let $\mathbf{x}_{1:L} = \{\mathbf{x}_1, \ldots, \mathbf{x}_L\}$ denote the resulting token sequence. Existing TSFMs typically adopt either point-wise tokenization (Ansari et al., 2024; Shi et al., 2024), where each token $\mathbf{x}_i$ corresponds to a single time step $\mathbf{y}_i$, or patch-wise tokenization (Nie et al., 2023; Liu et al., 2024b), where $\mathbf{x}_i$ corresponds to a patch $\mathbf{y}_{i+1:i+P}$ of length P.

Next-Token Prediction (NTP) NTP predicts the immediate next token from the observed history, i.e., $\hat{\mathbf{x}}_{t+1} = f_\theta(\mathbf{x}_{1:t})$, and minimizes

$$\mathcal{L}_{\text{NTP}}(\theta) = \mathbb{E}_t[\ell(\hat{\mathbf{x}}_{t+1}, \mathbf{x}_{t+1})], \tag{1}$$

where $\ell(\cdot, \cdot)$ denotes the prediction loss, such as MSE, MAE, or quantile loss.

Next-N-Token Prediction (NNTP) An alternative autoregressive formulation is to predict a short future segment of length N at each decoding step. The model is trained to generate next N tokens given all preceding tokens, $\mathbf{x}_{t+1:t+N} = f_\theta(\mathbf{x}_{1:t})$. Formally, the objective is to minimize:

$$\mathcal{L}_{\mathrm{NNTP}}(\theta) = \mathbb{E}_t \left[\frac{1}{N} \sum_{i=1}^{N} \ell(\hat{\mathbf{x}}_{t+i}, \mathbf{x}_{t+i}) \right]. \tag{2}$$

3.2 Limitations of Existing Paradigms

NTP has two fundamental limitations. First, its supervision is inherently local, which can bias the model toward short-term fluctuations rather than long-range temporal structures. Since individual time series tokens often carry limited semantic information and can be strongly affected by noise and outliers (Fu et al., 2025), such supervision may also be weak and noisy, encouraging spurious patterns. Second, NTP suffers from error accumulation, as prediction errors can propagate and compound over successive decoding steps. NNTP mitigates error accumulation by predicting multiple tokens at each step, but introduces a trade-off in selecting N. A small N remains vulnerable to error accumulation, whereas a large N requires the model to infer increasingly distant futures from the same historical context, making prediction substantially more difficult.

Moreover, both NTP and NNTP adopt a fixed prediction length during pretraining, which introduces a distinct and underexplored issue. The fixed-horizon objective can induce horizon-specific bias, thereby limiting generalization across the diverse forecasting horizons in downstream tasks. We identify this limitation empirically in Figure 2, and formalize it in the following proposition.

Proposition 1 (Horizon-Specific Bias under Fixed-Length Pretraining) *Consider a forecaster with prediction length N. Let $\ell_i[\theta]$ denote the expected loss of its i-th output, and define the risk up to horizon $K \leq N$ as*

$$\mathcal{L}_K[\theta] = \frac{1}{K} \sum_{i=1}^{K} \ell_i[\theta]. \tag{3}$$

Let $\theta_N^\star$ be an interior minimizer of $\mathcal{L}_{1:N}[\theta]$, and let $\mathcal{L}_{1:H}[\theta]$ have a unique minimizer $\theta_H^\star$ for some $H < N$. If

$$\sum_{i=H+1}^{N} \nabla_\theta \ell_i[\theta_N^\star] \neq \mathbf{0}, \tag{4}$$

then

$$\theta_N^\star \neq \theta_H^\star, \qquad \mathcal{L}_{1:H}^{[N]}[\theta_N^\star] > \mathcal{L}_{1:H}^{[N]}[\theta_H^\star]. \tag{5}$$

The proof is provided in Appendix A.1. This proposition shows that an optimum for a fixed pretraining horizon generally fails to remain optimal for other horizons. The reason is that optimizing the fixed-horizon objective balances gradients from all prediction positions, including those outside the target horizon; when these excluded positions exert nonzero gradients, they shift the optimum away from that of the target-horizon objective. Since downstream forecasting horizons are not fixed, fixed-horizon methods may generalize poorly to unseen horizons.

3.3 Next-x-Token Prediction

To overcome the aforementioned limitations, we reformulate autoregressive learning for time series as **Next-x-Token Prediction (NxTP)**, where the prediction horizon for each token is no longer fixed but dynamically sampled from a distribution. Formally, we define the corresponding training objective as follows.

$$\mathcal{L}_{\mathrm{NxTP}}(\theta) = \mathbb{E}_t \mathbb{E}_{h \sim q(h|t)} \left[\frac{1}{h} \sum_{i=1}^{h} \ell(\hat{\mathbf{x}}_{t+i}, \mathbf{x}_{t+i}) \right], \tag{6}$$

where $q(h \mid t)$ denotes the conditional distribution of the prediction horizon h given the current context position t. It is worth noting that we formulate the prediction horizon in the original timestep

domain because the scoring function for time series (e.g., MSE, MAE) is typically defined at the point level. This formulation also enables the model to generalize to arbitrary forecasting horizons, rather than restricting them to multiples of the token length. Consequently, for models using patch-wise tokenization, the prediction horizon may correspond to a non-integer number of tokens.

Effectiveness of NxTP. The prediction horizon of the downstream task is not fixed, which is the key to the effectiveness of NxTP. Assuming that the prediction horizon of the downstream task follows a distribution $p(h \mid t)$, we define the risk under a horizon distribution r as

$$\mathcal{R}_r[\theta] = \mathbb{E}_t \mathbb{E}_{h \sim r(h|t)} \left[\mathcal{L}_h[\theta; t] \right], \tag{7}$$

where $\mathcal{L}_h[\theta; t]$ follows the definition of Eq. 3. Thus, NxTP minimizes $\mathcal{R}_q[\theta]$, whereas downstream evaluation is measured by $\mathcal{R}_p[\theta]$. We analyze the effectiveness of NxTP with following theorem.

Theorem 2 (Expected advantage of NxTP) *Let $\theta_q^\star$ be a global minimizer of $\mathcal{R}_q$, and let $\theta_N^\star$ be a model trained with fixed horizon N. Define the expected advantage of NxTP under q as*

$$\gamma_q = \mathcal{R}_q[\theta_N^\star] - \mathcal{R}_q[\theta_q^\star] \geq 0. \tag{8}$$

Assume $0 \leq \mathcal{L}_h[\theta; t] \leq B$ for all contexts, horizons, and models. If

$$\mathbb{E}_t D_{\mathrm{KL}} \left[q(\cdot \mid t) \,\|\, p(\cdot \mid t) \right] < \frac{\gamma_q^2}{2B^2}, \tag{9}$$

then NxTP has lower expected risk under the downstream horizon distribution:

$$\mathcal{R}_p[\theta_q^\star] < \mathcal{R}_p[\theta_N^\star]. \tag{10}$$

The proof is provided in Appendix A.2. The theorem shows that NxTP attains an advantage in expected risk when q and p are sufficiently close. However, it also implies that NxTP may perform suboptimally when q is poorly chosen.

Design Choice of $q(h \mid t)$ Therefore, the choice of q is critical to the effectiveness of NxTP. To align q with realistic downstream scenarios, NxTP is designed to support sufficiently long forecasting horizons, enabling most forecasting tasks to be completed within a single decoding step and thereby mitigating the error accumulation caused by multi-step decoding. Meanwhile, it should retain the flexibility to accommodate short-, medium-, and long-term forecasting, balancing the diverse horizon requirements of downstream tasks. To achieve this balance, we define the horizon sampling distribution as $q(h \mid t) = \mathrm{LogUniform}(1, \min(t, H_{\text{max}}))$, where H_{max} denotes the maximum prediction horizon and is set to a sufficiently large value (e.g., 1024), enabling long-term forecasting within a single decoding step. Despite the wide range of candidate horizons, log-uniform sampling provides more balanced coverage across temporal scales, allocating relatively more probability mass to shorter horizons while maintaining sufficient coverage of longer ones.

3.4 Implementation

We implement the above training objective using a long-horizon prediction head combined with a loss masking scheme.

Prediction Head Specifically, the prediction head is designed to generate predictions for a sufficiently long maximum horizon H_{max}. Therefore, for deterministic forecasting models, we employ a head $g : \mathbb{R}^d \to \mathbb{R}^{H_{\max}}$, which maps each token representation with dimension d to deterministic predictions over the maximum horizon $H_{\max}$.

As quantile forecasting has become a standard paradigm for TSFMs, extending NxTP to quantile prediction is a natural consideration. A straightforward approach is to employ a quantile prediction head $g_q : \mathbb{R}^d \to \mathbb{R}^{H_{\max} \times |\mathcal{Q}|}$ that directly generates predictions for all quantiles in $\mathcal{Q}$ over the maximum horizon $H_{\max}$. However, its computation and parameter count both scale as $\mathcal{O}(dH_{\max}|\mathcal{Q}|)$. Under the commonly adopted setting of nine quantiles, ($\mathcal{Q} = \{0.1, 0.2, \ldots, 0.9\}$), this additional overhead can be prohibitive for lightweight models.

We therefore additionally introduce a lightweight quantile head that substantially reduces this overhead while retaining effective quantile forecasting performance. Rather than predicting each quantile directly, it predicts a parametric distribution $F(\cdot; \boldsymbol{\theta}_t)$ for each future time step t, where $\boldsymbol{\theta}_t$ denotes the corresponding distribution parameters. Specifically, the distribution head $g_d : \mathbb{R}^d \to \mathbb{R}^{H_{\max} \times K}$ maps each token representation to the parameters of the predictive distributions, where K is the number of parameters required by the chosen distribution. In practice, the predictive distribution can be instantiated as a Gaussian distribution ($K = 2$) or a Student's t-distribution ($K = 3$). Quantile forecasts are then obtained by evaluating the inverse cumulative distribution function:

$$\hat{\mathbf{y}}_t^{(q)} = F^{-1}(q; \boldsymbol{\theta}_t), \qquad q \in \mathcal{Q}. \tag{11}$$

Loss Masking We use loss masking to accommodate varying forecasting horizons, with a dynamically constructed mask determining which future timesteps contribute to the training objective. Specifically, for each input position t, we sample a forecasting horizon $h_t \sim q(h \mid t)$ and activate the loss for the first h_t future time steps while masking out the remaining predictions. Formally, we define the NxTP loss mask as $\mathbf{M} \in \{0, 1\}^{T \times H_{\max}}$, with

$$M_{t,\tau} = \mathbb{I}(\tau \leq h_t), \qquad h_t \sim q(h \mid t). \tag{12}$$

The training loss is then computed only over the unmasked prediction positions

$$\mathcal{L}_{\text{NxTP}} = \frac{1}{\|\mathbf{M}\|} \sum_{t=1}^{T} \sum_{i=1}^{H_{\max}} \mathbf{M}_{t,\tau}\, \ell\left(\hat{\mathbf{x}}_{t+i}, \mathbf{x}_{t+i}\right). \tag{13}$$

This masking strategy allows different input positions to receive supervision over different horizons within a single forward pass, thereby promoting generalization across forecasting horizons.

4 Experiments

4.1 Experimental Setup

Backbones We select four representative decoder-only models as backbone: Timer (Liu et al., 2024b), TimesFM (Das et al., 2024), Time-MoE (Shi et al., 2024), and Moirai-2.0 (Liu et al., 2025). For each backbone, we consider two model sizes: 30M and 70M for Timer, 70M and 200M for TimesFM, and base and large for Time-MoE, resulting in six configurations in total. We set the maximum horizon length to 960 for Timer with NxTP, while all other models use 1024. Since the selected models are relatively lightweight, we employ a Gaussian quantile head to support quantile forecasting. Detailed model configurations are provided in Appendix B.

Pretraining Data We use two major open-source datasets, GiftEvalPretrain (Aksu et al., 2024) and Chronos datasets (Ansari et al., 2024), as the pretraining corpus, with balanced sampling applied to improve pretraining efficiency (Shao et al., 2025). To ensure a fair comparison, the NxTP-based models and baseline models are pretrained on exactly the same data under the same sampling strategy, and none of the evaluation datasets are included in the pretraining corpus. See Appendix D for details of the pretraining data.

Pretraining Settings We train all models for 200,000 steps with an initial learning rate of 1×10^{-3} and a cosine scheduler with 10,000 warm-up steps. The global batch size is set to 512. We use a maximum context length of 3,840 for Timer, while a maximum context length of 4,096 is used for the rest of the models. All experiments are conducted on two NVIDIA RTX 6000D 84GB GPUs.

Evaluation Settings We adopt the long-term forecasting benchmark (Wu et al., 2023; Shao et al., 2024) and GIFT-Eval benchmark (Aksu et al., 2024) to evaluate the effectiveness of NxTP. The long-term forecasting benchmark consists of six datasets, including four ETT datasets, Electricity, and Weather. We consider four prediction horizons$\{96, 192, 336, 720\}$, with the context length selected via hyperparameter search. Performance is evaluated using MSE and MAE. The GIFT-Eval benchmark comprises 97 forecasting tasks spanning short-, medium-, and long-term forecasting, providing a comprehensive assessment of predictive performance. Following the official protocol, we report MASE and CRPS as the evaluation metrics. Detailed settings are provided in the Appendix D.

Table 1: Performance comparison of models trained with and without NxTP on long-term forecasting benchmark. Lower MAE and MSE indicate better performance. Better results are in **bold**, and Δ denotes the averaged improvements of NxTP over four prediction horizons.

Models		Timer-30M		+NxTP		Timer-50M		+NxTP		TimesFM-70M		+NxTP		TimesFM-200M		+NxTP		Time-MoE$_b$		+NxTP		Moirai-2.0		+NxTP	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.362	0.380	**0.352**	**0.374**	0.375	0.378	**0.348**	**0.371**	0.376	0.379	**0.349**	**0.373**	0.374	0.378	**0.358**	0.378	**0.353**	**0.377**	0.355	0.379	0.352	0.375	**0.346**	0.375
	192	0.413	0.401	**0.384**	**0.395**	0.414	0.407	**0.384**	**0.394**	0.424	0.405	**0.386**	**0.398**	0.420	0.404	**0.394**	**0.403**	0.390	0.402	0.390	0.402	0.392	0.400	**0.385**	**0.398**
	336	0.442	0.416	**0.400**	**0.407**	0.448	0.418	**0.401**	**0.406**	0.450	0.420	**0.401**	**0.411**	0.441	0.415	**0.412**	0.415	0.414	0.416	**0.408**	**0.413**	0.415	0.413	**0.406**	**0.410**
	720	0.475	0.443	**0.422**	**0.430**	0.477	0.445	**0.429**	**0.432**	0.470	0.445	**0.431**	**0.438**	0.472	0.445	**0.442**	**0.438**	0.444	0.441	**0.430**	**0.436**	0.439	0.435	**0.431**	0.435
	Avg.	0.423	0.410	**0.390**	**0.402**	0.429	0.412	**0.391**	**0.401**	0.430	0.412	**0.392**	**0.405**	0.427	0.411	**0.402**	**0.409**	0.400	0.409	**0.396**	**0.408**	0.400	0.406	**0.392**	**0.405**
	Δ			**7.80%**	**1.95%**			**8.86%**	**2.67%**			**8.84%**	**1.70%**			**5.85%**	**0.49%**			**1.00%**	**0.24%**			**2.00%**	**0.25%**
ETTh2	96	0.299	**0.330**	**0.283**	0.331	0.300	0.332	**0.283**	**0.331**	0.304	0.336	**0.283**	**0.332**	0.312	**0.340**	**0.309**	0.342	0.285	0.333	**0.277**	**0.324**	**0.278**	0.332	0.281	**0.331**
	192	0.365	0.377	**0.350**	**0.376**	0.367	**0.377**	**0.343**	0.378	0.360	0.374	**0.344**	**0.373**	0.370	0.381	**0.362**	**0.379**	0.348	0.378	**0.341**	**0.369**	0.353	0.384	**0.341**	**0.373**
	336	0.390	0.400	**0.366**	**0.391**	0.398	0.406	**0.360**	**0.394**	0.375	**0.389**	**0.362**	0.390	0.390	0.396	**0.366**	0.396	0.373	0.403	**0.366**	**0.391**	0.384	0.407	**0.360**	**0.390**
	720	0.401	0.424	**0.394**	**0.422**	0.415	0.432	**0.390**	**0.423**	0.411	**0.422**	**0.394**	0.423	0.411	0.422	**0.395**	0.422	0.401	0.432	**0.385**	**0.417**	0.400	0.433	**0.388**	**0.420**
	Avg.	0.364	0.383	**0.348**	**0.380**	0.370	0.387	**0.344**	**0.382**	0.363	0.380	**0.346**	0.380	0.371	0.385	**0.358**	0.385	0.352	0.387	**0.342**	**0.375**	0.354	0.389	**0.343**	**0.379**
	Δ			**4.40%**	**0.78%**			**7.03%**	**1.29%**			**4.68%**	0.0%			**3.50%**	0.0%			**2.84%**	**3.10%**			**3.11%**	**2.57%**
ETTm1	96	0.340	0.343	**0.318**	0.343	0.333	0.345	**0.291**	**0.337**	0.342	**0.337**	**0.327**	0.338	0.362	0.344	**0.334**	**0.342**	0.334	0.350	**0.311**	**0.348**	0.320	**0.343**	**0.316**	0.357
	192	0.396	**0.375**	**0.356**	0.376	0.385	0.375	**0.324**	**0.363**	0.406	0.372	**0.378**	**0.368**	0.410	0.390	**0.389**	**0.372**	0.360	0.384	**0.339**	**0.371**	0.369	**0.372**	**0.350**	0.382
	336	0.429	0.400	**0.382**	**0.396**	0.424	0.400	**0.350**	**0.383**	0.459	0.400	**0.420**	**0.391**	0.468	0.402	**0.431**	**0.397**	0.400	0.413	**0.382**	**0.393**	0.403	**0.396**	**0.374**	0.399
	720	0.481	0.432	**0.420**	**0.422**	0.482	0.434	**0.402**	**0.414**	0.531	0.450	**0.440**	**0.427**	0.551	0.458	**0.449**	**0.436**	0.457	0.451	**0.423**	**0.418**	0.468	0.439	**0.431**	**0.425**
	Avg.	0.412	0.388	**0.369**	**0.384**	0.406	0.389	**0.342**	**0.374**	0.435	0.390	**0.391**	**0.381**	0.448	0.399	**0.401**	**0.387**	0.388	0.400	**0.364**	**0.383**	0.390	**0.388**	**0.368**	0.391
	Δ			**10.4%**	**1.03%**			**15.8%**	**3.86%**			**10.1%**	**2.31%**			**10.5%**	**3.01%**			**6.19%**	**4.25%**			**5.64%**	-0.77%
ETTm2	96	0.184	**0.251**	0.184	0.260	0.180	**0.248**	**0.178**	0.253	0.205	0.262	**0.190**	**0.260**	0.214	0.275	**0.195**	**0.263**	**0.174**	0.250	0.175	0.250	**0.184**	**0.264**	0.189	0.269
	192	0.247	**0.296**	**0.244**	0.304	0.241	**0.294**	**0.239**	0.297	0.279	0.310	**0.265**	0.310	0.283	0.319	**0.266**	**0.311**	0.237	0.305	**0.232**	**0.292**	0.249	**0.310**	**0.245**	0.311
	336	0.304	**0.337**	**0.298**	0.341	0.302	0.338	**0.293**	**0.336**	0.335	0.347	**0.320**	0.347	0.345	0.359	**0.319**	**0.345**	0.294	0.344	**0.281**	**0.328**	0.298	0.346	**0.297**	0.346
	720	0.389	0.394	**0.365**	**0.387**	0.391	0.397	**0.360**	**0.383**	0.430	0.407	**0.355**	**0.393**	0.445	0.415	**0.355**	**0.393**	0.376	0.397	**0.353**	**0.377**	0.390	0.401	**0.365**	**0.391**
	Avg.	0.281	**0.320**	**0.273**	0.323	0.279	0.319	**0.268**	**0.317**	0.312	0.332	**0.283**	**0.328**	0.322	0.342	**0.284**	**0.328**	0.270	0.324	**0.260**	**0.311**	0.280	0.330	**0.274**	**0.329**
	Δ			**2.84%**	-0.94%			**3.94%**	**3.94%**			**9.62%**	**1.20%**			**11.8%**	**4.09%**			**3.70%**	**4.01%**			**2.14%**	**0.30%**
ECL	96	0.134	0.222	**0.130**	**0.221**	0.132	0.221	**0.129**	**0.219**	0.140	0.223	**0.128**	**0.217**	0.138	0.221	**0.128**	**0.215**	0.130	0.223	**0.127**	**0.217**	0.138	**0.226**	**0.132**	0.228
	192	0.152	**0.238**	**0.149**	0.239	0.153	0.239	**0.148**	**0.237**	0.158	0.240	**0.148**	**0.235**	0.156	0.237	**0.147**	**0.233**	0.148	0.241	**0.145**	**0.234**	0.155	0.242	**0.150**	**0.240**
	336	0.171	0.257	**0.167**	**0.256**	0.173	0.258	**0.166**	**0.255**	0.176	0.258	**0.166**	**0.254**	0.174	0.255	**0.167**	**0.252**	0.167	0.260	**0.163**	**0.252**	0.173	0.260	**0.167**	**0.257**
	720	0.213	0.292	**0.210**	0.292	0.217	0.295	**0.210**	**0.292**	0.218	0.293	**0.210**	**0.291**	0.216	0.290	**0.209**	**0.289**	0.209	0.296	**0.207**	**0.290**	0.217	0.297	**0.210**	**0.292**
	Avg.	0.168	0.252	**0.164**	0.252	0.169	0.253	**0.163**	**0.251**	0.173	0.254	**0.163**	**0.249**	0.171	0.251	**0.163**	**0.247**	0.164	0.255	**0.161**	**0.248**	0.171	0.256	**0.165**	**0.254**
	Δ			**2.38%**	0.0%			**3.55%**	**0.63%**			**5.78%**	**1.97%**			**4.68%**	**1.59%**			**1.83%**	**2.75%**			**3.51%**	**0.78%**
Weather	96	0.169	**0.203**	**0.168**	0.207	0.170	0.204	**0.167**	0.204	0.171	**0.204**	**0.167**	0.205	0.176	0.207	**0.168**	**0.205**	0.163	0.213	0.163	**0.199**	**0.166**	**0.205**	0.170	0.209
	192	0.228	**0.254**	**0.224**	0.256	0.224	0.252	**0.221**	0.252	0.228	0.255	**0.224**	**0.254**	0.237	0.258	**0.227**	**0.257**	0.215	0.263	0.217	**0.251**	0.228	0.259	**0.225**	**0.257**
	336	0.284	0.296	0.284	**0.295**	0.281	0.292	**0.276**	**0.291**	0.288	0.296	**0.278**	**0.293**	0.296	0.299	**0.280**	**0.296**	0.274	0.298	**0.272**	**0.292**	0.280	0.300	**0.278**	**0.295**
	720	0.358	0.342	**0.343**	**0.340**	0.348	**0.339**	**0.345**	0.340	0.359	0.346	**0.350**	**0.344**	0.365	0.349	**0.352**	**0.346**	0.346	0.349	**0.342**	**0.335**	0.346	0.344	0.346	**0.340**
	Avg.	0.260	**0.274**	**0.255**	0.275	0.256	0.272	**0.252**	0.272	0.262	0.275	**0.255**	**0.274**	0.269	0.278	**0.257**	**0.276**	0.250	0.281	**0.249**	**0.269**	0.255	0.277	0.255	**0.275**
	Δ			**1.92%**	-0.36%			**1.56%**	0.0%			**2.67%**	**0.36%**			**4.46%**	**0.72%**			**0.40%**	**4.27%**			0.0%	**0.72%**

Table 2: Performance comparison of models trained with and without NxTP on GIFT-Eval benchmark. Lower MASE and CRPS indicate better performance. Better results are highlighted in **bold**.

Method	Timer-30M		Timer-50M		TimesFM-70M		TimesFM-200M		Time-MoE$_b$		Moirai-2.0	
	MASE	CRPS	MASE	CRPS	MASE	CRPS	MASE	CRPS	MASE	CRPS	MASE	CRPS
Original	0.794	0.562	0.790	0.560	0.816	0.577	0.828	0.580	0.773	0.554	0.768	0.550
+NxTP	**0.779**	**0.559**	**0.776**	**0.555**	**0.782**	**0.563**	**0.788**	**0.566**	**0.764**	**0.551**	**0.756**	**0.538**

4.2 Main Results

Long-Term Forecasting Benchmark As shown in Table 1, incorporating NxTP consistently improves forecasting performance across six models and datasets. The gains are particularly pronounced for the ETTm1 and ETTm2 datasets, where NxTP reduces the averaged MSE by up to **9.8%** and **5.7%**, respectively. We attribute this to the finer temporal granularity and stronger local fluctuations of minutely ETT data, which exacerbate the weak supervision and error accumulation of existing autoregressive paradigms, the very issues that NxTP is designed to address. Moreover, the improvement generally becomes larger as the forecasting horizon increases. For example, on the ETTm1 dataset, NxTP reduce the averaged MSE by **6.6%** on horizon 96, while the reduction grows to **13.6%** on horizon 720. This trend indicates that NxTP effectively mitigates error accumulation during long-horizon forecasting by avoiding repeated autoregressive decoding. Overall, these results demonstrate that NxTP is a model-agnostic enhancement that not only improves forecasting accuracy, but also strengthens generalization across arbitrary prediction horizons.

GIFT-Eval Benchmark Tabl 2 reports the zero-shot performance on the GIFT-Eval benchmark. Notably, 55 of its 97 forecasting tasks are short-term. Even on such a benchmark that emphasizes short-term forecasting, NxTP still yields clear performance gains. These results indicate that NxTP improves long-term forecasting by reducing error accumulation, while mitigating horizon-specific bias also brings gains to short-term forecasting. Moreover, the improvements in CRPS further validate the effectiveness of our lightweight Gaussian quantile head.

Table 3: Comparison with different autoregressive paradigms. Results are averaged from four prediction horizons $\{96, 192, 336, 720\}$. Best and second-best results are shown in **bold** and underlined, respectively. Full results are provided in Table 7.

Method	ETTh1		ETTh2		ETTm1		ETTm2		Electricity		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NTP	0.429	0.412	0.370	0.387	0.406	0.389	0.279	0.319	0.169	0.253	0.256	**0.272**
NNTP ($N = 2$)	0.409	0.405	0.358	0.381	0.360	0.382	0.288	0.324	0.165	**0.251**	0.256	**0.272**
NNTP ($N = 4$)	0.404	0.404	0.353	0.378	0.362	0.385	0.279	0.322	0.166	**0.251**	0.257	0.274
NNTP ($N = 8$)	0.400	0.402	0.345	**0.376**	0.358	0.387	0.280	0.326	0.166	0.252	0.255	0.274
STP ($N = 2$)	0.411	0.405	0.349	**0.376**	0.356	0.377	0.271	0.317	0.167	0.252	0.256	0.273
STP ($N = 4$)	0.409	0.409	0.358	0.381	0.366	0.386	0.269	**0.314**	0.166	0.252	0.256	0.273
NxTP	**0.391**	**0.401**	**0.344**	0.382	**0.342**	**0.374**	**0.268**	0.317	**0.163**	**0.251**	**0.252**	**0.272**

4.3 Comparison with Different Autoregressive Paradigms

In this section, we comprehensively compare NxTP with other autoregressive paradigms, including NTP, NNTP, and STP (Liu et al., 2026b), in terms of forecasting performance and efficiency. We use Timer-50M as the common backbone for all paradigms. For NNTP, we consider $N = 2$, $N = 4$, and $N = 8$, covering the commonly adopted configurations in existing models. For STP, since each increment of N introduces an additional Transformer decoder layer, we consider only $N = 2$ and $N = 4$ to avoid excessive computational overhead. Figure 3 presents the overall performance-efficiency comparison. Compared with other autoregressive paradigms, NxTP achieves clear advantages in both forecasting performance and computational efficiency. We provide detailed analyses of forecasting performance and efficiency below.

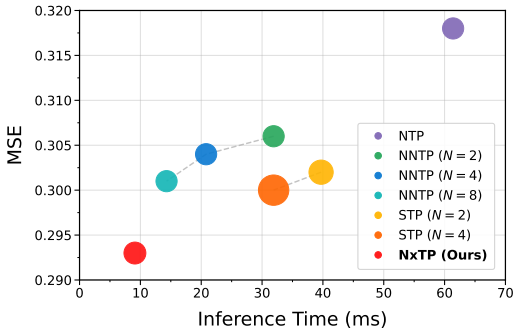

Figure 3: **Performance-efficiency comparison of autoregressive paradigms.** Bubble area indicates the GPU memory footprint.

Performance We compare the performance of different autoregressive paradigms on long-term forecasting benchmarks. Table 3 reports the results averaged over four prediction horizons. Overall, NxTP achieves the best performance on all six datasets, consistently outperforming the second-best methods in MSE, with relative improvements of 3.93% on ETTm1 and 2.25% on ETTh1. Notably, increasing the prediction length in NNTP or STP does not consistently improve performance: while larger N benefits NNTP on several datasets, the gains are dataset-dependent, and STP exhibits similar variability. In contrast, NxTP maintains strong performance across datasets without relying on a fixed prediction length, demonstrating its effectiveness and robustness for long-horizon forecasting.

Efficiency We compare the efficiency of NxTP against other autoregressive paradigms in terms of the number of parameters, inference time, and GPU memory usage. We set the batch size to 8, the context length to 3840, and consider the prediction horizon 96 and 960 to evaluate the efficiency under the short- and long-term setting. We report only the number of parameters of the prediction head (and TimeSTP modules for STP), as the model backbone are identical. Table 4 shows the efficiency comparison results. It can be observed that both NTP and NNTP require multi-step rollout, resulting in slow infer-

Table 4: Efficiency metrics of NxTP and other autoregressive paradigms.

Methods	#Params	Memory	Inference Time 96	960
NTP	0.10M	1.08GB	8.6ms	64.1ms
NNTP (N=2)	0.20M	1.08GB	8.7ms	31.9ms
NNTP (N=4)	0.39M	1.08GB	8.8ms	20.8ms
NNTP (N=8)	0.79M	1.08GB	8.8ms	14.3ms
STP (N=2)	10.6M	1.24GB	9.9ms	39.7ms
STP (N=4)	31.6M	1.55GB	12.8ms	31.9ms
NxTP	1.97M	1.12GB	9.1ms	9.1ms

ence in long-term forecasting scenarios. Although STP eliminates the need for rollout, each additional predicted token requires an extra Transformer decoder layer, introducing a large number of parameters and incurring substantial computational overhead. In contrast, NxTP achieves significantly faster inference than these methods in long-term forecasting, yielding a $7.0\times$ speedup over conventional NTP. Moreover, the additional memory footprint and short-term inference latency introduced by NxTP are marginal.

4.4 Ablation Study

To further study the effectiveness of the NxTP, we conduct ablation studies to examine two key designs: the effect of the prediction horizon distribution $q(h \mid t)$ and the impact of the loss masking.

Effect of $q(h \mid t)$ We consider two alternatives for $q(h \mid t)$. For convenience, we define $H_t = \min(t, H_{\max})$. The first is a deterministic choice, $q(h \mid t) = \delta(h - H_t)$, which always sets the prediction horizon to the largest feasible value, capped by $H_{\max}$. The second is a uniform distribution, $q(h \mid t) = \mathrm{Uniform}(1, H_t)$, which uniformly samples one feasible prediction horizon from 1 to H_t. As shown in Table 5, replacing our choice of $q(h \mid t)$ with either alternative consistently degrades overall performance. The results broken down by prediction length (Table 8) show that NxTP achieves a particularly substantial improvement in the short-term setting (prediction length 96). Both alternative distributions place disproportionately more emphasis on longer horizons, which can compromise short-term forecasting performance. In contrast, the log-uniform distribution allocates probability mass more evenly across scales, enabling NxTP to preserve strong short-term accuracy even when $H_{\max}$ is large.

Effect of Loss Masking To study the role of loss masking, we consider two variants: (1) removing the loss mask entirely and (2) replacing it with a random mask of the same masking ratio. As shown in Table 5, both variants lead to a degradation in overall performance. Without masking, the objective exhibits a bias toward long-horizon prediction, similar to that observed when replacing $q(h \mid t)$, while also making optimization more challenging. Replacing the proposed mask with a random mask of the same ratio also degrades performance, demonstrating that the gains of NxTP do not arise merely from masking itself, but from its horizon-aware masking strategy.

Table 5: Ablation results of NxTP with Timer-50M as backbone. The results areaveraged from all predicted lengths. Full ablation results are provided in Table 8.

Method	ETTh1		ETTh2		ETTm1		ETTm2		Electricity		Weather	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
NxTP	0.391	**0.401**	0.344	0.382	**0.342**	**0.374**	**0.268**	**0.317**	**0.163**	**0.251**	**0.252**	**0.272**
$q(h \mid t) = \delta(h - H_t))$	0.392	0.407	0.342	0.382	0.364	0.390	0.291	0.339	0.170	0.251	0.261	0.279
$q(h \mid t) = \mathrm{Uniform}(1, H_t)$	0.392	0.408	**0.341**	**0.381**	0.362	0.383	0.285	0.334	0.169	0.255	0.259	0.278
w/o Loss masking	**0.389**	0.407	0.343	0.383	0.361	0.389	0.282	0.336	0.169	0.257	0.262	0.280
Random masking	0.392	0.408	0.342	0.382	0.365	0.389	0.287	0.337	0.262	0.280	0.257	0.277

5 Conclusion

In this work, we redefine autoregressive pretraining for time series as Next-x-Token Prediction (NxTP), which dynamically samples the prediction horizon for each token according to the available context, rather than relying on a fixed prediction length. This reformulation exposes models to diverse forecasting horizons, mitigating horizon-specific bias while allowing long sequences to be generated in substantially fewer decoding steps. Across two established forecasting benchmarks and six variants derived from four representative TSFMs, NxTP consistently improves forecasting accuracy, generalization across arbitrary horizons, and inference efficiency. These results suggest that varying the prediction horizon is an effective and broadly applicable design principle for autoregressive time series pretraining. We hope NxTP provides a useful foundation for developing more robust, flexible, and efficient time series foundation models. Due to limited computational resources, we leave the exploration of the adaptive horizon distribution and the scaling behavior of NxTP to future work. See Appendix E for detailed discussion on limitations and future directions.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and for assisting with the verification of mathematical claims. We have not used generative AI tools for research ideation, section drafting, literature search, or data generation, and the remaining disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for improving the clarity and readability of the manuscript. We have reviewed all AI-assisted work. For example, all AI-assisted mathematical derivations and proofs were checked and verified by the authors, and all AI-polished text was reviewed and revised by the authors to ensure correctness and faithfulness to the original content. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Gift-eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. Tirex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. *Advances in Neural Information Processing Systems*, 38:57529–57580, 2026.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H Nguyen, Wesley M. Gifford, Chandra Reddy, and Jayant Kalagnanam. Tiny time mixers (TTMs): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. In *NeurIPS*, 2024.

Kun Feng, Shaocheng Lan, Yuchen Fang, Wenchao He, Lintao Ma, Xingyu Lu, and Kan Ren. Kairos: Towards adaptive and generalizable time series foundation models. *arXiv preprint arXiv:2509.25826*, 2025.

Yisong Fu, Zezhi Shao, Chengqing Yu, Yujie Li, Zhulin An, Qi Wang, Yongjun Xu, and Fei Wang. Selective learning for deep time series forecasting. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 98084–98115. Curran Associates, Inc., 2025.

Yisong Fu, Zezhi Shao, Chengqing Yu, Yujie Li, Yongjun Xu, Xueqi Cheng, and Fei Wang. Zeus: Towards tuning-free foundation model for time series analysis. In *Forty-third International Conference on Machine Learning*, 2026.

Tilmann Gneiting and Adrian Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–378, 03 2007. doi: 10.1198/016214506000001437.

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

R. J Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22, 2006.

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

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025.

Yiding Liu, Yifan Hu, Hongjie Xia, Peiyuan Liu, Hongzhou Chen, Xilin Dai, Zewei Dong, and Jiang-Ming Yang. Falcon-x: A time series foundation model for heterogeneous multivariate modeling. *arXiv preprint arXiv:2605.27286*, 2026a.

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

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning*, 2024b.

Yong Liu, Xingjian Su, Shiyu Wang, Haoran Zhang, Haixuan Liu, Yuxuan Wang, Zhou Ye, Yang Xiang, Jianmin Wang, and Mingsheng Long. Timer-s1: A billion-scale time series foundation model with serial scaling. *arXiv preprint arXiv:2603.04791*, 2026b.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72, 2025.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *ICLR*. OpenReview.net, 2023.

Youngsuk Park, Danielle C. Maddix, Franccois-Xavier Aubet, Kelvin K. Kan, Jan Gasthaus, and Yuyang Wang. Learning quantile functions without quantile crossing for distribution-free time series forecasting. In *International Conference on Artificial Intelligence and Statistics*, 2021. URL https://api.semanticscholar.org/CorpusID:244103049.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, et al. Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):291–305, 2024.

Zezhi Shao, Yujie Li, Fei Wang, Chengqing Yu, Yisong Fu, Tangwen Qian, Bin Xu, Boyu Diao, Yongjun Xu, and Xueqi Cheng. Blast: Balanced sampling time series corpus for universal forecasting models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2502–2513, 2025.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*, 2024.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024.

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

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *ICLR*. OpenReview.net, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, pp. 11106–11115, 2021.

A PROOFS

A.1 PROOF OF PROPOSITION 1

Proof 1 *Since θ_N^* is an interior minimizer of $\mathcal{L}_N$, it satisfies*

$$\nabla_\theta \mathcal{L}_N[\theta_N^*] = \frac{1}{N}\sum_{h=1}^{N} \nabla_\theta \ell_h[\theta_N^*] = \mathbf{0}. \tag{14}$$

Therefore,

$$\begin{aligned} \nabla_\theta \mathcal{L}_H[\theta_N^*] &= \frac{1}{H}\sum_{h=1}^{H} \nabla_\theta \ell_h[\theta_N^*] \\ &= \frac{1}{H}\sum_{h=N+1}^{H} \nabla_\theta \ell_h[\theta_N^*]. \end{aligned} \tag{15}$$

By Eq. 4,

$$\nabla_\theta \mathcal{L}_H[\theta_N^*] \neq \mathbf{0}. \tag{16}$$

Hence, θ_N^* cannot be a minimizer of $\mathcal{L}_H$. Since $\mathcal{L}_H$ has a unique minimizer θ_H^*, we obtain

$$\theta_N^* \neq \theta_H^*, \tag{17}$$

which further implies

$$\mathcal{L}_H[\theta_N^*] > \mathcal{L}_H[\theta_H^*]. \tag{18}$$

A.2 PROOF OF THEOREM 2

Proof 2 *Define the per-task loss difference*

$$\Delta(t,h) = \mathcal{L}_h[\theta_q^\star; t] - \mathcal{L}_h[\theta_N^\star; t]. \tag{19}$$

By the definition of $\theta_q^\star$,

$$\mathbb{E}_{(t,h)\sim q}[\Delta(t,h)] = -\gamma_q. \tag{20}$$

Since both losses lie in $[0, B]$, $\Delta(t,h) \in [-B, B]$. The difference between its expectations under p and q is bounded by

$$\left|\mathbb{E}_{(t,h)\sim p}[\Delta] - \mathbb{E}_{(t,h)\sim q}[\Delta]\right| \leq 2B\,\mathbb{E}_t D_{\mathrm{TV}}\left[p(\cdot \mid t), q(\cdot \mid t)\right]. \tag{21}$$

Applying Pinsker's inequality and Jensen's inequality gives

$$\mathbb{E}_{(t,h)\sim p}[\Delta] \leq -\gamma_q + B\sqrt{2\mathbb{E}_t D_{\mathrm{KL}}\left[q(\cdot \mid t) \,\|\, p(\cdot \mid t)\right]}. \tag{22}$$

Condition 9 therefore implies $\mathbb{E}_{(t,h)\sim p}[\Delta] < 0$, completing the proof.

B MODEL DETAILS

We select four representative model architectures as backbone models for our experiments, and introduce them as follows.

- **Timer** (Liu et al., 2024b): an early decoder-only TSFM trained with NTP, featuring a unified multi-task framework.
- **TimesFM** (Das et al., 2024): A representative decoder-only TSFM trained with NNTP.
- **Time-MoE** (Shi et al., 2024): A mixture-of-experts decoder-only TSFM that scales to billions of parameters. Unlike most models, Time-MoE adopts point-wise tokenization and trains multiple prediction heads, using a greedy decoding strategy for inference, which is essentially NNTP.
- **Moirai-2.0** (Liu et al., 2025): As a successor to Moirai (Woo et al., 2024), Moirai-2.0 shifts to a decoder-only architecture, employs a quantile head to support probabilistic forecasting, is trained with NNTP, and is primarily characterized by its lightweight design.

Table 6 presents the detailed configurations of these models. Except for the prediction head that NxTP needs to modify, all other settings are kept identical to the official configurations. For Timer, we set the maximum prediction horizon to 960, while for the remaining models it is set to 1024. Since the selected models are all relatively lightweight, we adopt a Gaussian quantile head, a choice that is empirically validated in our experiments.

Table 6: **Model configurations.** † denotes the NxTP setting of the model.

Models	**Patch Size**	**#Layers**	d_{model}	d_{ff}	**#Heads**	**Pred. Length**	$\mathbf{H}^{\dagger}_{\text{max}}$
Timer-30M	96	6	768	1536	8	96	960
Timer-50M	96	6	1024	2048	8	96	960
TimesFM-70M	32	10	1024	1024	16	128	1024
TimesFM-200M	32	20	1280	1280	16	128	1024
Time-MoE$_b$	1	12	384	1536	12	$\{1, 8, 32, 64\}$	1024
Moirai-2.0	16	6	384	1024	6	64	1024

C Benchmark Details

C.1 Long-Term Forecasting Benchmark

We conduct experiments on six widely-recognized datasets in long-term forecasting benchmarks (Wu et al., 2023; Shao et al., 2024), including:

- **ETT** (Electricity Transformer Temperature) (Zhou et al., 2021) contains 7 features of electricity transformer data collected from two separate counties from July 2016 to July 2018. It contains four datasets: **ETTh1, ETTh2, ETTm1, ETTm2**, where ETTh1 and ETTh2 are recorded every hour, and ETTm1 and ETTm2 are recorded every 15 minutes.
- **ECL** (Electricity) (Wu et al., 2021) records the hourly electricity consumption data of 321 clients from 2012 to 2014. Each variable represents a client's electricity consumption.
- **Weather**(Wu et al., 2021) includes 21 meteorological factors collected every 10 minutes from the weather station of the Max Planck Biogeochemistry Institute in 2020.

Following common practice, we consider four prediction horizons $\{96, 192, 336, 720\}$. For each horizon, the context length is selected via hyperparameter search.

Metrics We adopt **Mean Squared Error (MSE)** and **Mean Absolute Error (MAE)** as evaluation metrics, where lower values indicate better performance. They are defined as follows.

$$\text{MSE}(\hat{y}, y) = \frac{1}{C \cdot T} \sum_{i=1}^{C} \sum_{t=1}^{T} \left(\hat{y}_t^i - y_t^i\right)^2, \tag{23}$$

$$\text{MAE}(\hat{y}, y) = \frac{1}{C \cdot T} \sum_{i=1}^{C} \sum_{t=1}^{T} |\hat{y}_t^i - y_t^i|. \tag{24}$$

C.2 GIFT-Eval Benchmark

Benchmarks GIFT-Eval (Aksu et al., 2024) comprises 23 datasets spanning multiple domains, including nature, energy, healthcare, finance, transportation, and cloud operations, covering 144,000 time series with 177 million data points. These datasets define a total of 97 forecasting configurations, including 55 short-term, 21 medium-term, and 21 long-term forecasting tasks, thereby providing a comprehensive assessment of the model's predictive capabilities. Compared to long-term forecasting benchmarks, GIFT-Eval places greater emphasis on short-term forecasting and additionally provides evaluation metrics for probabilistic predictions.

Metrics Following the official protocol of GIFT-Eval, we report **Mean Absolute Scaled Error (MASE)** and **Continuous Ranked Probability Score (CRPS)** as evaluation metrics. Both metrics are lower-is-better.

MASE (Hyndman & Koehler, 2006) evaluates point forecasting performance by scaling the mean absolute error (MAE) of a model against that of a seasonal naive benchmark. It is defined as

$$\text{MASE} = \frac{\text{MAE}_{\text{model}}}{\text{MAE}_{\text{seasonal naive}}} \tag{25}$$

with the Seasonal Naive MAE being defined as

$$\text{MAE}_{\text{seasonal naive}} = \frac{1}{T-m} \sum_{t=m+1}^{T} |y_t - y_{t-m}| \tag{26}$$

where T is the length of the *training* split of the series, y_t is the value of the series at time t, and m is the seasonal period.

CRPS (Gneiting & Raftery, 2007) is a scoring rule for probabilistic forecasting. Given a predicted distribution with CDF F and a ground truth value y, the CRPS is defined as:

$$\text{CRPS}(F, y) = \int_0^1 2\Lambda_\alpha(F^{-1}(\alpha), y)\, d\alpha, \tag{27}$$

where $\Lambda_\alpha(q, y)$ denotes the quantile loss.

In practice, CRPS is approximated by a discrete sum over a finite set of quantile levels. This approximation, often referred to as the mean weighted quantile loss (Park et al., 2021), is given by:

$$\text{CRPS} \approx \frac{1}{K} \sum_{k=1}^{K} \text{wQL}[\alpha_k], \tag{28}$$

where K is the number of quantile levels, and $\{\alpha_1, \alpha_2, \ldots, \alpha_K\}$ are the selected quantile levels. The weighted quantile loss $\text{wQL}[\alpha]$ for each quantile level α is calculated as:

$$\text{wQL}[\alpha] = 2\frac{\sum_t \Lambda_\alpha(\hat{q}_t(\alpha), y_t)}{\sum_t |y_t|}, \tag{29}$$

where $\hat{q}_t(\alpha)$ is the predicted α-quantile at time step t.

As in GIFT-Eval, both MASE and CRPS are further normalized by the performance of the Seasonal Naive forecast on the *test* split.

D Pretraining Corpora

The pretraining datasets used in our work primarily come from two major sources.

1. **Chronos Datasets** (Ansari et al., 2024) contain 94B data points. We do not employ the TSMixup data augmentation technique proposed in Chronos.
2. **GiftEvalPretrain Datasets** (Aksu et al., 2024) is the pretraining dataset collection used in the GIFT-Eval benchmark. It comprises 71 univariate and 17 multivariate datasets spanning seven domains and 13 temporal frequencies, totaling 4.5 million time series and 230B data points. This dataset does not overlap with the test sets of the GIFT-Eval benchmark.

The detailed dataset composition is provided in Table 9 and 10. To mitigate the imbalance in pattern distributions across the training data, we adopt the balanced sampling strategy (Shao et al., 2025) rather than assigning fixed probabilities per dataset. This promotes faster convergence and improves

the pretraining efficiency. Apart from the aforementioned open-source datasets, we do not use any additional synthetic or private datasets, which could otherwise substantially affect model performance on the benchmarks. Since our goal is to demonstrate the effectiveness of NxTP, we ensure that both the models with and without NxTP are pretrained on exactly the same data. To prevent data leakage, we exclude all evaluation datasets from pretraining data.

E Limitations and Future Works

While NxTP provides a simple and effective reformulation of autoregressive pretraining for time series, several limitations remain and suggest directions for future work.

Choice of the horizon distribution. As shown in Theorem 2, the advantage of NxTP depends on the closeness between the training horizon distribution $q(h \mid t)$ and the downstream horizon distribution $p(h \mid t)$. If q is poorly chosen, NxTP may perform suboptimally. In this work, we adopt a log-uniform distribution to balance coverage across temporal scales and enable long-horizon forecasting within a single decoding step. However, this choice is fixed and task-agnostic. An interesting direction is to adapt q during pretraining, or condition it on dataset-level or task-level metadata. Since different training datasets have different sampling frequencies and correspond to different downstream settings, conditioning q on such metadata would allow the training horizon distribution to more closely match the downstream horizon distribution.

Scaling behavior. The scaling behavior of TSFMs remains largely unverified. Existing efforts to scale up TSFMs have not consistently translated into better performance, and in many cases lightweight models remain competitive with, or even outperform, much larger ones. This suggests that the scaling behavior of TSFMs may differ from that of language models. Since NxTP modifies the pretraining objective rather than the backbone architecture, its scaling behavior may differ further from that of conventionally pretrained TSFMs. Due to limited computational resources, we leave a systematic study of the scaling behavior of NxTP to future work.

F Full Results

This section provides the full experimental results.

F.1 Full Results of Autoregressive Paradigm Comparison

The full results of autoregressive paradigm comparison are provided in Table 7.

F.2 Full Ablation Results

The full ablation results are provided in Table 8.

Table 7: **Full results of autoregressive paradigm comparison.** Best and second-best results are shown in **bold** and underlined, respectively.

Models		NxTP		STP $N=2$		STP $N=4$		NNTP $N=2$		NNTP $N=4$		NNTP $N=8$		NTP	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.348**	**0.371**	0.359	0.375	0.359	0.377	0.358	0.373	0.354	0.373	0.353	0.373	0.375	0.378
	192	**0.384**	**0.394**	0.403	0.398	0.398	0.405	0.400	0.399	0.394	0.398	0.387	**0.394**	0.414	0.407
	336	**0.401**	**0.406**	0.429	0.411	0.426	0.417	0.424	0.412	0.419	0.411	0.409	0.407	0.448	0.418
	720	**0.429**	**0.432**	0.452	0.435	0.452	0.436	0.455	0.435	0.447	0.435	0.449	0.434	0.477	0.445
	Avg.	**0.391**	**0.401**	0.411	0.405	0.409	0.409	0.409	0.405	0.404	0.404	0.400	0.402	0.429	0.412
ETTh2	96	**0.283**	0.331	0.287	**0.327**	0.290	0.329	0.294	0.332	0.286	**0.327**	0.284	0.328	0.300	0.332
	192	**0.343**	0.378	0.348	**0.369**	0.361	0.379	0.357	0.375	0.353	0.372	0.348	0.371	0.367	0.377
	336	**0.360**	0.394	0.370	0.390	0.381	0.397	0.377	0.394	0.368	0.391	**0.360**	**0.387**	0.398	0.406
	720	0.390	0.423	0.389	**0.416**	0.399	0.418	0.405	0.422	0.403	0.421	**0.387**	0.417	0.415	0.432
	Avg.	**0.344**	0.382	0.349	**0.376**	0.358	0.381	0.358	0.381	0.353	0.378	0.345	**0.376**	0.370	0.387
ETTm1	96	**0.291**	**0.337**	0.300	0.338	0.296	0.343	0.301	0.342	0.307	0.348	0.307	0.351	0.333	0.345
	192	**0.324**	**0.363**	0.337	0.364	0.342	0.372	0.342	0.369	0.344	0.373	0.342	0.376	0.385	0.375
	336	**0.350**	**0.383**	0.365	0.385	0.381	0.396	0.372	0.391	0.370	0.392	0.368	0.394	0.424	0.400
	720	**0.402**	**0.414**	0.421	0.421	0.445	0.433	0.425	0.424	0.425	0.428	0.416	0.425	0.482	0.434
	Avg.	**0.342**	**0.374**	0.356	0.377	0.366	0.386	0.360	0.382	0.362	0.385	0.358	0.387	0.406	0.389
ETTm2	96	**0.178**	0.253	0.182	0.254	0.180	0.252	0.189	0.257	0.185	0.258	0.185	0.262	0.180	**0.248**
	192	**0.239**	0.297	**0.239**	0.296	**0.239**	0.295	0.255	0.303	0.247	0.302	0.249	0.305	0.241	**0.294**
	336	0.293	0.336	**0.288**	**0.331**	0.291	**0.331**	0.310	0.340	0.305	0.339	0.307	0.343	0.302	0.338
	720	**0.360**	0.383	0.374	0.386	0.364	**0.379**	0.398	0.397	0.379	0.387	0.379	0.392	0.391	0.397
	Avg.	**0.268**	0.317	0.271	0.317	0.269	**0.314**	0.288	0.324	0.279	0.322	0.280	0.326	0.279	0.319
ECL	96	**0.129**	**0.219**	0.132	0.221	0.132	0.222	0.130	**0.219**	0.131	0.220	0.133	0.223	0.132	0.221
	192	**0.148**	**0.237**	0.152	0.240	0.150	0.238	0.150	**0.237**	0.149	**0.237**	0.150	0.238	0.153	0.239
	336	**0.166**	0.255	0.169	0.256	0.169	0.255	0.168	**0.254**	0.169	0.255	0.168	0.256	0.173	0.258
	720	**0.210**	0.292	0.213	0.292	0.213	0.292	0.212	0.293	0.215	0.293	0.212	**0.291**	0.217	0.295
	Avg.	**0.163**	**0.251**	0.167	0.252	0.166	0.252	0.165	**0.251**	0.166	**0.251**	0.166	0.252	0.169	0.253
Weather	96	**0.167**	**0.204**	0.171	0.206	0.170	0.206	0.169	**0.204**	0.170	0.206	0.169	0.206	0.170	**0.204**
	192	**0.221**	**0.252**	0.222	0.253	**0.221**	**0.252**	0.224	**0.252**	0.225	0.255	0.224	0.255	0.224	**0.252**
	336	**0.276**	0.291	0.279	0.292	0.277	0.291	0.278	**0.290**	0.282	0.293	0.277	0.293	0.281	0.292
	720	**0.345**	0.340	0.353	0.341	0.354	0.342	0.352	0.342	0.351	0.341	0.348	0.340	0.348	**0.339**
	Avg.	**0.252**	**0.272**	0.256	0.273	0.256	0.273	0.256	**0.272**	0.257	0.274	0.255	0.274	0.256	**0.272**

Table 8: Full ablation results of NxTP with Timer-50M as backbone.

Models		**NxTP**		$q(h \mid t)$ $\delta(h-t)$		$q(h \mid t)$ Uniform$(1,t)$		w/o Loss masking		Random masking	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.348**	**0.371**	0.356	0.380	0.356	0.380	0.353	0.380	0.357	0.382
	192	**0.384**	**0.394**	0.388	0.402	0.390	0.403	0.385	0.402	0.389	0.403
	336	0.401	**0.406**	0.399	0.412	0.401	0.414	**0.395**	0.412	0.400	0.414
	720	0.429	**0.432**	0.424	0.433	0.422	0.434	**0.421**	0.433	0.423	0.434
	Avg.	0.391	**0.401**	0.392	0.407	0.392	0.408	**0.389**	0.407	0.392	0.408
ETTh2	96	0.283	**0.331**	**0.281**	0.335	0.283	0.335	0.286	0.337	0.284	0.336
	192	0.343	0.378	**0.337**	**0.373**	**0.337**	**0.373**	0.339	0.374	0.338	0.374
	336	0.360	0.394	0.354	0.391	**0.352**	**0.390**	0.354	0.392	0.356	0.392
	720	**0.390**	**0.423**	0.395	0.428	**0.390**	0.424	0.394	0.428	0.391	0.425
	Avg.	0.344	0.382	0.342	0.382	**0.341**	**0.381**	0.343	0.383	0.342	0.382
ETTm1	96	**0.291**	**0.337**	0.317	0.355	0.312	0.350	0.313	0.354	0.315	0.355
	192	**0.324**	**0.363**	0.353	0.381	0.349	0.375	0.349	0.380	0.353	0.380
	336	**0.350**	**0.383**	0.377	0.399	0.374	0.395	0.374	0.399	0.377	0.398
	720	**0.402**	0.414	0.410	0.423	0.412	**0.413**	0.409	0.423	0.413	0.424
	Avg.	**0.342**	**0.374**	0.364	0.390	0.362	0.383	0.361	0.389	0.365	0.389
ETTm2	96	**0.178**	**0.253**	0.203	0.277	0.194	0.270	0.195	0.274	0.196	0.273
	192	**0.239**	**0.297**	0.273	0.326	0.264	0.319	0.263	0.323	0.266	0.323
	336	**0.293**	**0.336**	0.323	0.360	0.319	0.357	0.314	0.358	0.321	0.360
	720	0.360	**0.383**	0.365	0.391	0.362	0.390	**0.356**	0.390	0.365	0.393
	Avg.	**0.268**	**0.317**	0.291	0.339	0.285	0.334	0.282	0.336	0.287	0.337
ECL	96	**0.129**	**0.219**	0.135	0.226	0.134	0.224	0.135	0.226	0.136	0.227
	192	**0.148**	**0.237**	0.155	0.244	0.153	0.242	0.154	0.244	0.155	0.245
	336	**0.166**	**0.255**	0.174	0.262	0.173	0.260	0.173	0.262	0.175	0.263
	720	**0.210**	**0.292**	0.214	0.296	0.214	0.295	0.215	0.296	0.215	0.296
	Avg.	**0.163**	**0.251**	0.170	0.257	0.169	0.255	0.169	0.257	0.170	0.258
Weather	96	**0.167**	**0.204**	0.171	0.209	0.172	0.211	0.173	0.211	0.172	0.210
	192	**0.221**	**0.252**	0.231	0.260	0.233	0.261	0.233	0.262	0.229	0.260
	336	**0.276**	**0.291**	0.289	0.300	0.289	0.300	0.289	0.301	0.284	0.299
	720	0.345	0.340	0.352	0.346	**0.341**	**0.339**	0.353	0.346	0.343	0.340
	Avg.	**0.252**	**0.272**	0.261	0.279	0.259	0.278	0.262	0.280	0.257	0.277

Table 9: Statistics of the pretraining datasets.

Dataset	Domain	Frequency	# Time Series	# Observation
Mexico City Bikes	Transport	H	494	38,687,004
Solar	Energy	5T, H	5,166	1,085,664,720
Spanish Energy and Weather	Energy	H	1	35,064
Taxi	Transport	30T, H	4,856	3,346,350
USHCN	Climate	D	6,090	235,016,970
Weatherbench	Nature	H, D, W	225,280	79,375,265,520
Wiki (100k)	Web	D	100,000	274,100,000
Wind Farms	Energy	H, D	100,000	856,800,000
KDD Cup 2018	Energy	H	270	2,897,004
London Smart Meters	Energy	30T	5,560	166,528,896
M4	Econ/Fin	D, H, M, W	52,000	50,318,646
Pedestrian Counts	Transport	H	66	3,132,346
Rideshare	Transport	H	2,304	859,392
Temperature-Rain	Nature	D	32,072	22,290,040
Uber TLC	Transport	H, D	262	1,174,932
BDG-2 Panther	Energy	H	105	919,800
BDG-2 Fox	Energy	H	135	2,324,568
BDG-2 Rat	Energy	H	280	4,728,288
BDG-2 Bear	Energy	H	91	1,482,312
Low Carbon London	Energy	H	713	9,543,348
SMART	Energy	H	5	95,709
IDEAL	Energy	H	219	1,265,672
Sceaux	Energy	H	1	34,223
Borealis	Energy	H	15	83,269
Buildings900K	Energy	H	1,792,328	15,702,590,000
CMIP6	Climate	6H	1,351,680	1,973,453,000
ERA5	Climate	H	245,760	2,146,959,000
Azure VM Traces 2017	CloudOps	5T	159,472	885,522,908
Borg Cluster Data 2011	CloudOps	5T	143,386	537,552,854
Alibaba Cluster Trace 2018	CloudOps	5T	58,409	95,192,530
Taxi	Transport	30T	67,984	54,999,060
Wiki-Rolling	Web	D	47,675	40,619,100
M5	Sales	D	30,490	58,327,370
LargeST	Transport	5T	42,333	4,452,510,528
PEMS03	Transport	5T	358	9,382,464
PEMS04	Transport	5T	307	5,216,544
PEMS07	Transport	5T	883	24,921,792
PEMS08	Transport	5T	170	3,035,520
PEMS Bay	Transport	5T	325	16,937,700
Los-Loop	Transport	5T	207	7,094,304
Beijing Subway	Transport	30T	276	248,400
SHMetro	Transport	15T	288	1,934,208
HZMetro	Transport	15T	80	146,000
Q-Traffic	Transport	15T	45,148	264,386,688
Subseasonal	Climate	D	862	14,097,148
Subseasonal Precipitation	Climate	D	862	9,760,426

Table 10: Statistics of the pretraining datasets.

Dataset	Domain	Frequency	# Time Series	# Observation
Covid19 Energy	Energy	H	1	31,912
GEF12	Energy	H	20	788,280
GEF14	Energy	H	1	17,520
GEF17	Energy	H	8	140,352
PDB	Energy	H	1	17,520
BDG-2 Hog	Energy	H	24	421,056
BDG-2 Bull	Energy	H	41	719,304
BDG-2 Cockatoo	Energy	H	1	17,544
ELF	Energy	H	1	21,792
Wind Power	Energy	4S	1	7,397,147
Solar Power	Energy	4S	1	7,397,222
Oikolab Weather	Climate	H	8	800,456
Elecdemand	Energy	30T	1	17,520
Covid Mobility	Transport	D	362	148,602
Kaggle Web Traffic Weekly	Web	W	145,063	16,537,182
Extended Web Traffic	Web	D	145,063	370,926,091
M1 Yearly	Econ/Fin	Y	106	3,136
M1 Quarterly	Econ/Fin	Q	198	9,854
M1 Monthly	Econ/Fin	M	617	44,892
M3 Yearly	Econ/Fin	Y	645	18,319
M3 Quarterly	Econ/Fin	Q	756	37,004
M3 Monthly	Econ/Fin	M	1,428	141,858
M3 Other	Econ/Fin	Q	174	11,933
NN5 Daily	Econ/Fin	D	111	81,585
NN5 Weekly	Econ/Fin	W	111	11,655
Tourism	Econ/Fin	Y, Q, M	1212	150,822
CIF 2016	Econ/Fin	M	72	6,334
Traffic Weekly	Transport	W	862	82,752
Traffic Hourly	Transport	H	862	14,978,112
Australian Electricity Demand	Energy	30T	5	1,153,584
Sunspot	Nature	D	1	73,894
Vehicle Trips	Transport	D	329	32,512
Weather	Climate	D	3,010	42,941,700
FRED MD	Econ/Fin	M	107	76,612
Bitcoin	Econ/Fin	D	18	74,824
KDD Cup 2022	Energy	10T	134	4,727,519
GoDaddy	Econ/Fin	M	3,135	128,535
Favorita Sales	Sales	D	111,840	139,179,538
Favorita Transactions	Sales	D	54	84,408
China Air Quality	Nature	H	437	5,739,234
Beijing Air Quality	Nature	H	12	420,768
Residential Load Power	Energy	T	271	145,994,559
Residential PV Power	Energy	T	233	125,338,950
CDC Fluview ILINet	Healthcare	W	75	63,903
CDC Fluview WHO NREVSS	Healthcare	W	74	41,760
Project Tycho	Healthcare	W	1,258	1,377,707