

## A Model architecture details

### A.1 Input/output scaling

To ensure numerical stability, we compute the  $\hat{\mu}_t$  and  $\hat{s}_t$  using an efficient vectorized implementation (Listing 1) of Welford’s online algorithm [46], incorporating Bessel’s correction to provide an unbiased estimator of variance, as described in Option A of [83]. We stabilize training against extreme outliers by incorporating weak information from the global statistics.

```
1 def compute_causal_statistics(  
2     data: torch.Tensor,  
3     weights: torch.Tensor,  
4     minimum_scale: float,  
5 ) -> Tuple[torch.Tensor, torch.Tensor]:  
6     # Compute causal means at each time step  
7     weighted_data = weights * data  
8     cum_weights = torch.cumsum(weights, dim=-1)  
9     cum_values = torch.cumsum(weighted_data, dim=-1)  
10    denominator = cum_weights.clamp_min(1.0)  
11    causal_means = cum_values / denominator  
12  
13    # For Welford’s algorithm, we need to compute the correction term  
14    # delta using the difference between the current value and the  
15    # previous running mean.  
16    shifted_means = torch.zeros_like(causal_means)  
17    shifted_means[..., 1:] = causal_means[..., :-1]  
18    delta = data - shifted_means  
19  
20    # Compute m_2, the second moment accumulator for Welford’s  
21    # algorithm.  
22    increment = delta * (data - causal_means) * weights  
23    m_2 = torch.cumsum(increment, dim=-1)  
24  
25    # Compute the variance using Bessel’s correction.  
26    causal_variance = m_2 / torch.clamp(denominator - 1.0, min=1.0)  
27    causal_scale = torch.sqrt(causal_variance + minimum_scale)  
28  
29    return causal_means, causal_scale
```

Listing 1: Vectorized PyTorch implementation of Welford’s algorithm for computing causal statistics

In our ablation study (Section E), we find that causal scaling leads to dramatic performance improvements over naive global scaling.

### A.2 Attention mechanism

To address the unique challenges of time series data, and particularly to adapt transformer architectures for multivariate time-series forecasting, several works have implemented modifications to the attention mechanism. These strategies have included:

- Concatenating variates along the time dimension and computing full self-attention between every variate/time location, as in the “any-variate attention” used by Woo et al. [13]. This can capture every possible variate and time interaction, but it is costly in terms of computation and memory usage.
- Assuming variate independence, and computing attention only in the time dimension as in Nie et al. [19], Shi et al. [14]. This is efficient, but throws away all information about variate-wise interactions.
- Computing attention only in the variate dimension, and using a feed-forward network in the time dimension [84, 35].
- Computing “factorized attention,” where each transformer block contains a separate variate and time attention computation [47–49]. This allows both variate and time mixing, and is

more efficient than full cross-attention. However, it doubles the effective depth of the network.

In Section 3.1, we propose a novel approach that allows for both variate and time interactions, while reducing the computational cost and improving overall scalability.

### A.2.1 Complexity analysis

After the patchwise embedding layer, we have inputs of shape  $\mathbf{X} \in \mathbb{R}^{B \times M \times \frac{L}{P} \times D}$ , where  $B$  is the batch dimension,  $M$  is the number of variates per batch item,  $\frac{L}{P}$  is time steps divided by patch width, and  $D$  is the model embedding dimension.

**Time-wise attention.** We parallelize along the time dimension by reshaping the input tensor from 4 dimensions to 3:

$$\mathbf{X} \in \mathbb{R}^{B \times M \times \frac{L}{P} \times D} \rightarrow \mathbf{X}_{\text{time}} \in \mathbb{R}^{(B \times M) \times \frac{L}{P} \times D}$$

This allows for attention to be calculated independently in parallel per variate, giving a complexity of:

$$\mathcal{O}(M \times (\frac{L}{P})^2 \times D)$$

In the time-wise attention blocks, we use causal masking and rotary positional embeddings [43] with XPOS [44] in order to autoregressively model time-dependent features.

**Variate-wise attention.** We similarly parallelize along the variate dimension by reshaping the input tensor:

$$\mathbf{X} \in \mathbb{R}^{B \times M \times \frac{L}{P} \times D} \rightarrow \mathbf{X}_{\text{variate}} \in \mathbb{R}^{(B \times \frac{L}{P}) \times M \times D}$$

We calculate attention in parallel for each time step, with complexity:

$$\mathcal{O}(\frac{L}{P} \times M^2 \times D)$$

In the variate-wise blocks, we use full bidirectional attention (without causal masking) in order to preserve permutation invariance of the covariates, with a block-diagonal ID mask to ensure that only related variates attend to each other. This masking allows us to pack multiple independent multivariate time series into the same batch, in order to improve training efficiency and reduce the amount of padding.

**Computational complexity.** Each transformer block in our model contains  $N$  time-wise attention layers and 1 variate-wise layer. The complexity for full self-attention over  $N + 1$  layers, where interactions can occur across all variates and sequence positions, would be of complexity:

$$\mathcal{O}\left((N + 1) \times M^2 \times \left(\frac{L}{P}\right)^2 \times D\right) \quad (1)$$

This reflects the quadratic dependence on both the sequence length  $\frac{L}{P}$  and the variate dimension  $M$ , with linear dependence on the embedding dimension  $D$ . However, by utilizing factorized attention, we can reduce the computational complexity of the attention calculation to:

$$\begin{aligned} \mathcal{O}\left(N \times M \times \left(\frac{L}{P}\right)^2 \times D + \frac{L}{P} \times M^2 \times D\right) = \\ \mathcal{O}\left(D \times \frac{L}{P} \times M \times \left(N \times \frac{L}{P} + M\right)\right) \end{aligned} \quad (2)$$

We demonstrate that factorized variate-wise attention is asymptotically smaller in computational complexity than full self-attention (see Equation 1 and Equation 2). When comparing a model with full self-attention, we can assume  $N$  and  $D$  are fixed. Therefore:

$$\mathcal{O}\left(M \times \left(\frac{L}{P}\right)^2 + \frac{L}{P} \times M^2\right) < \mathcal{O}\left(M^2 \times \left(\frac{L}{P}\right)^2\right)$$

which reduces to:

$$\mathcal{O}\left(M + \frac{L}{P}\right) < \mathcal{O}\left(M \times \frac{L}{P}\right).$$

Thus, by factorizing attention into time-wise and variate-wise components, the computational complexity is reduced, especially for large numbers of variates  $M$  or long sequences  $\frac{L}{P}$ , making it more scalable than full self-attention.

### A.3 Probabilistic prediction

Practitioners who rely on time series forecasting typically prefer probabilistic predictions. A common practice in neural time series models is to use an output layer where the model regresses the parameters of a probability distribution. This allows for prediction intervals to be computed using Monte Carlo sampling (see Appendix A.4 [85]).

Common choices for an output layer are Normal [85] and Student-T [86, 21], which can improve robustness to outliers. Moirai [13] allows for more flexible residual distributions by proposing a novel mixture model incorporating a weighted combination of Gaussian, Student-T, Log-Normal, and Negative-Binomial outputs.

However, real-world time series can often have complex distributions that are challenging to fit, with outliers, heavy tails, extreme skew, and multimodality. In order to accommodate these scenarios, we introduce an even more flexible output likelihood in Section 3.1 based on a SMM [55].

TOTO makes predictions using a mixture of  $K$  Student-T distributions (where  $K$  is a hyperparameter) for each time step, as well as a learned weighting. Formally, the SMM is defined by:

$$p(x) = \sum_{k=1}^K \pi_k \mathcal{T}(x \mid \mu_k, \tau_k, \nu_k) \quad (3)$$

where  $\pi_{k \in K}$  are nonnegative mixing coefficients which sum to 1 for the  $k$ th Student's t-distribution  $\mathcal{T}_k$  with  $\nu_k$  degrees of freedom, mean  $\mu_k$ , and scale  $\tau_k$ .  $\mathcal{T}(x \mid \mu, \sigma, \nu)$  is defined as:

$$\mathcal{T}(x \mid \mu, \tau, \nu) = \frac{\Gamma\left(\frac{\nu+d}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right) (\nu\pi)^{d/2} |\tau|^{1/2}} \left(1 + \frac{1}{\nu}(x - \mu)^\top \tau^{-1} (x - \mu)\right)^{-\frac{\nu+d}{2}}, \quad (4)$$

where  $\Gamma(\cdot)$  is the gamma function.

In our ablation study (Appendix E), we find that the SMM improves both point prediction and probabilistic forecasting accuracy when compared with a single Student-T distribution as used in TiDE [86], Lag-Llama [21], and implementations of DeepAR [85], PatchTST [19], iTransformer [42], and others in the popular open-source GluonTS library [87].

The parameters of this mixture model are computed from the flattened features  $h_t \in \mathbb{R}^D$  produced by the transformer backbone for each time step  $t$ , where  $D$  is the model's embedding dimension. Using a set of linear projections with weight matrices  $W \in \mathbb{R}^{K \times D}$  and bias vectors  $b \in \mathbb{R}^K$ , we derive all  $K$  mixture components simultaneously. For each time step  $t$ , the parameters are computed

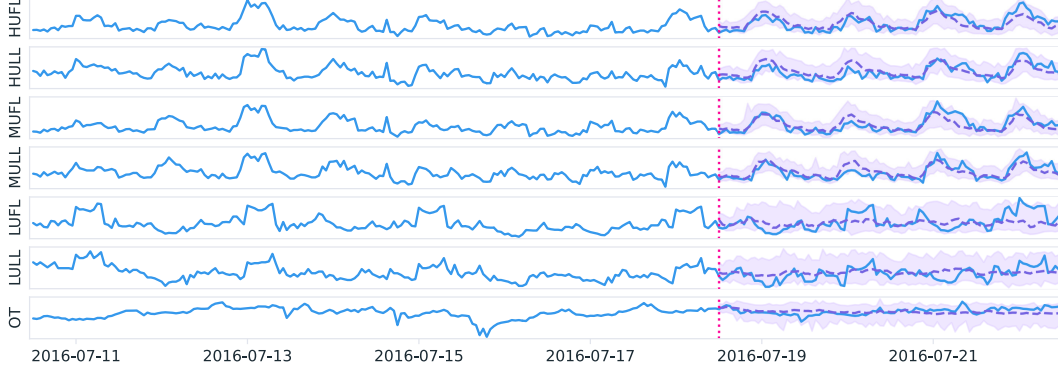


Figure 6: Example of TOTO 's 96-step zero-shot forecasts on the ETTh1 dataset, showing multivariate probabilistic predictions. Solid lines represent ground truth, dashed lines represent median point forecasts, and shaded regions represent 95% prediction intervals.

as:

$$\nu_t = 2 + \max(\text{softplus}(W_\nu h_t + b_\nu), \epsilon) \quad (5)$$

$$\mu_t = W_\mu h_t + b_\mu \quad (6)$$

$$\tau_t = \max(\text{softplus}(W_\tau h_t + b_\tau), \epsilon) \quad (7)$$

$$\tilde{\pi}_t = W_\pi h_t + b_\pi \quad (8)$$

where each equation produces a vector in  $\mathbb{R}^K$  containing the parameters for all mixture components at time  $t$ . The individual component parameters  $\nu_{t,k}$ ,  $\mu_{t,k}$ ,  $\tau_{t,k}$ , and  $\tilde{\pi}_{t,k}$  (the mixture logits) are the  $k$ th elements of these vectors. The parameter  $\epsilon$  is a small positive constant, and  $\text{softplus}(x) = \log(1 + e^x)$ . The use of  $\text{softplus}$  and  $\epsilon$  ensure that the scale  $\tau$  remains positive. Similarly, we add the constraint  $\nu > 2$  to ensure that each component of our mixture has well-defined first and second moments (mean and variance).

The mixture weights  $\pi$  are computed using by applying softmax to the logits:

$$\pi_{t,k} = \text{softmax}(\tilde{\pi}_t, k) = \frac{e^{\tilde{\pi}_{t,k}}}{\sum_{j=1}^K e^{\tilde{\pi}_{t,j}}} \quad (9)$$

An example distribution median and 95th percentile is illustrated in Fig. 6.

#### A.4 Forecasting

When performing inference, we draw  $u$  (for some user specified integer  $u > 0$ ) samples from the mixture distribution at each timestamp, then feed each sample back into the decoder for the next prediction, resulting in  $n$  identically and independently sampled time-series. This allows us to produce prediction intervals at any quantile, limited only by the number of samples. Our exact sampling procedure for several tasks is detailed in Section C.2.

#### A.5 Loss function

TOTO learns the conditional distribution  $p(X_{i+1}|X_{1:i})$ , where  $X_i$  represents the  $i$ -th patch containing multiple time steps.

The  $\mathcal{L}_{\text{NLL}}$  optimizes probabilistic predictions and is defined as:

$$\mathcal{L}_{\text{NLL}}(x, \mu, \tau, \nu) = -\log(p(x_t|X_{1:i})) = -\log\left(\sum_{k=1}^K \pi_{t,k} \mathcal{T}(x_t | \mu_{t,k}, \tau_{t,k}, \nu_{t,k})\right) \quad (10)$$

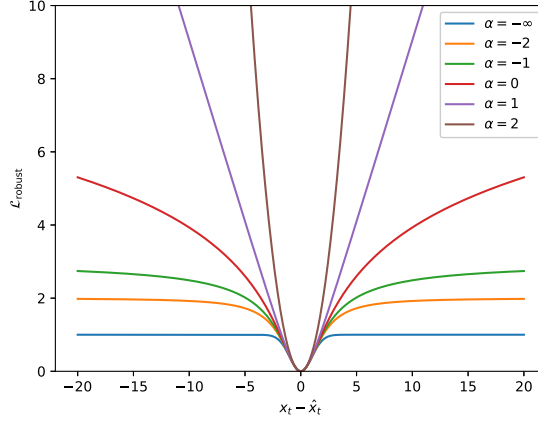


Figure 7: Visualization of generalized robust loss for different values of  $\alpha$ , with  $\delta$  fixed at 1. Changing  $\delta$  scales the horizontal axis.

where  $p(x_t|X_{1:i})$  is the probability density of the ground truth  $x_t$  under the model’s predicted mixture distribution conditioned on all previous patches. The parameters  $\pi_{t,k}$ ,  $\mu_{t,k}$ ,  $\tau_{t,k}$ , and  $\nu_{t,k}$  are the mixture weights and Student-T parameters computed by the model for time step  $t$ .

For a ground truth value  $x_t$  in patch  $i + 1$  and the mean prediction  $\hat{x}_t = \mathbb{E}[p(x_t|X_{1:i})]$ , the robust loss is defined below [61]:

$$\mathcal{L}_{\text{Robust}(\alpha,\delta)}(x_t, \hat{x}_t) = \begin{cases} \frac{1}{2}((x_t - \hat{x}_t)/\delta)^2, & \alpha = 2 \\ \log\left(\frac{1}{2}((x_t - \hat{x}_t)/\delta)^2 + 1\right), & \alpha = 0 \\ 1 - \exp\left(-\frac{1}{2}((x_t - \hat{x}_t)/\delta)^2\right), & \alpha = -\infty \\ \frac{|\alpha-2|}{\alpha} \left[ \left( \frac{((x_t - \hat{x}_t)/\delta)^2}{|\alpha-2|} + 1 \right)^{\alpha/2} - 1 \right], & \text{otherwise} \end{cases} \quad (11)$$

Here,  $\mathcal{L}_{\text{Robust}(\alpha,\delta)}$  serves as a point prediction error measure, where  $\alpha \leq 2$  is a shape parameter that controls the robustness to outlier observations (Fig. 7) and  $\delta > 0$  is a scale parameter that determines the size of the parabolic portion of the loss curve. This loss component directly penalizes point prediction accuracy, and we conjecture this may help steer the mixture model away from degenerate solutions of the type described in [60]. In our ablation study, we find that adding the robust loss component significantly improves point forecasting accuracy without hurting probabilistic predictions (Section E).

$\mathcal{L}$  is applied to each timestep  $t$  in the target patch  $X_{i+1}$ , and the total loss is aggregated across all timesteps during training. By combining the probabilistic  $\mathcal{L}_{NLL}$  loss with the robust point-prediction loss, we achieve both accurate distribution modeling and stable convergence, especially in domains with highly heterogeneous data characteristics. The hyperparameter  $\lambda_{NLL}$  controls the balance between these two loss components and is tuned empirically.

## A.6 Hyperparameter optimization

To determine the optimal architecture and training configuration for Toto, we conducted an extensive hyperparameter sweep using Optuna [88], a Bayesian optimization framework. We employed the Tree-structured Parzen Estimator (TPE) algorithm to efficiently explore the high-dimensional search space.

Our optimization objective was to minimize the validation mean absolute error (MAE) on multi-step forecasting tasks on a random validation split of the observability portion of the pretraining data. We train the model using the AdamW optimizer [89] with a WSD learning rate scheduler [90]. We performed this sweep for 133 iterations over ranges described in Table 5, each for 50,000 training steps.

Category	Values / Ranges
Patch Size	{16, 32, 64}
Variate-wise Attention Frequency	Every {3, 4, 6, 12} layers
Variate-wise Layer First	[True, False]
$\mathcal{T}$ Components	[8, 16, 24, 32]
Loss Function	$\lambda_{\text{NLL}} \in [0.05, 1.0]$
Robust Loss Params	$\alpha \in \{-\infty, -2, 0, 0.5, 1.0\}, \delta \in [0.1, 3.0]$
Warmup Steps	[0, 10,000]
Stable Ratio*	[.1, .9]
Learning Rate	$[10^{-5}, 5 \times 10^{-3}]$
Weight Decay	$[10^{-3}, 10^{-1}]$
Synthetic Data Proportion	[0.0, 0.75]
Shuffling Type	[Normally Distributed, Adjacent, Random, None]
Normally Distributed Shuffling Standard Deviation	[.15, 5000]
Shuffling Frequency	[0.0, 0.3]

Table 5: Summary of hyperparameter search space. \*Stable Ratio defines the proportion of steps that are stable after the warmup phase of the WSD learning rate schedule.

The resulting hyperparameter configuration described in Table 6 obtained the best multistep (average of 96 and 192) MAE on the Datadog validation set.

Hyperparameter	Value
Embedding Dimension	768
MLP Dimension	3072
# Layers	12
# Heads	12
# Variates	32
Spacewise Layer Cadence	12
Patch Size	64
# $\mathcal{T}$ Mixture Model Components	24
Annealing Schedule	WSD
Optimizer	AdamW
$(\beta_1, \beta_2)$	(0.9579, 0.9581)
Weight Decay	0.0014
Initial Learning Rate	0.0005
Warmup Steps	6784
Stable Steps	112,255
Decay Steps	15,962
Batch Size	128
Total Train Steps	135,001
$\mathcal{L}_{\text{Robust}} \alpha$	0.0000
$\mathcal{L}_{\text{Robust}} \delta$	0.1010
$\lambda_{\text{NLL}}$	0.5755
$\kappa$	10

Table 6: Hyperparameters for Toto

In Section E, we perform an ablation study on the impact of various model components. We optimize speed and memory usage by utilizing fused kernel implementations and memory efficient attention operations via xformers [91], (with the FlashAttention-3 kernel [92]).

We ran all experiments, including hyperparameter tuning, final model training, and benchmark evaluation on a GPU cluster consisting of A100s and H100s.

## B Training data preprocessing

### B.1 Observability dataset

Observability metrics are retrieved from a large-scale time series database using a specialized query language supporting filters, group-bys, time aggregation, and various transformations and postprocessing functions (Fig. 8). We consider groups returned from the same query to be related variates in a multivariate time series. After we retrieve the query results, we discard the query strings and group identifiers, keeping only the raw numeric data. As described in Section 3.2, we source metrics defined by user-generated queries. This excludes any customer data and is sourced solely from the internal users and telemetry.

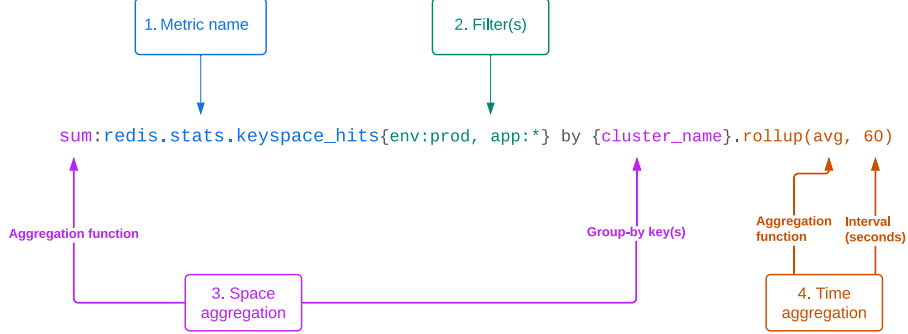


Figure 8: Example metric query in the Datadog platform. The metric name (1) determines which metric is being queried. The filter clause (2) limits which contexts are queried, in this case restricting the query to apps in the prod environment. The space aggregation (3) indicates that the sum of the metric value should be returned for each unique value of the group-by key(s), aggregated across all other keys. The time aggregation (4) indicates that metric values should be aggregated to the average for each 60-second interval. The query results will be a multivariate time series with 1-minute time steps, and with separate individual variates for each unique value of `cluster_name`.

### B.2 Public datasets

We train on a public dataset corpus, which exposes the model to diverse time series behaviors across different domains and sampling frequencies. Our pre-training dataset incorporates a diverse collection of time series from the GIFT-Eval Pretrain collection [26] and non-overlapping Chronos datasets [11]. These datasets include `ercot`, `exchange_rate`, `weatherbench_daily`, `weatherbench_hourly`, `weatherbench_monthly`, `dominick`, `mexico_city_bikes`, `ushcn_daily`, and `wiki_daily_100k`.

### B.3 Synthetic data

We supplement our training with synthetic data to further improve model performance. Our synthetic dataset consists of procedurally generated time series using an approach similar to TimesFM [12], as well as `kernel_synth_1m` from the Chronos dataset [11]. Synthetic data constitutes approximately 33% of our training dataset.

We generate synthetic time series through the composition of components such as piecewise linear trends, ARMA processes, sinusoidal seasonal patterns, and various residual distributions. Our procedural generation randomly combines multiple processes per variate to introduce diverse patterns. The generation includes creating base series with transformations, clipping extreme values, and rescaling to specified ranges.

These synthetic datasets help the model learn robust representations by providing examples with specific characteristics that might be underrepresented in real-world data.

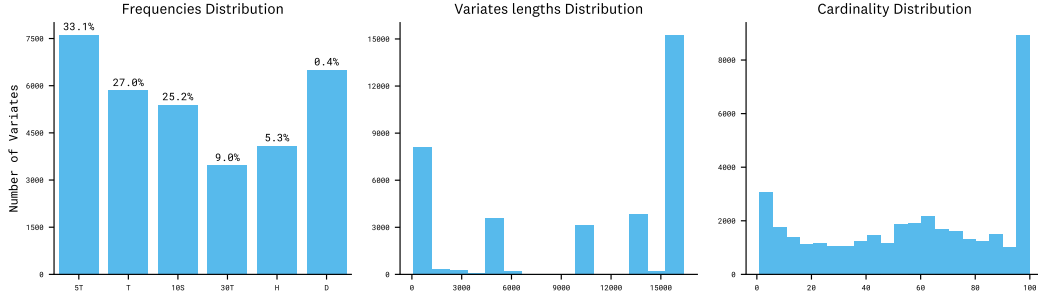


Figure 9: Representative figure showing the metadata breakdown by variate in the dataset: (left) sampling frequency distribution – bar heights show the number of *variates* with each frequency, while the percentages show the fraction of overall *observations*, (middle) series length distribution, and (right) number of variates distribution.

## B.4 Preprocessing

To prepare the raw time series for training, we apply padding and masking techniques to align the series lengths, making them divisible by the patch stride. This involves adding necessary left-padding to both the time series data and the ID mask, ensuring compatibility with the model's requirements.

Next, various data augmentations are employed to enhance the dataset's robustness. We introduce random time offsets to prevent memorization caused by having series always align the same way with the patch grid. After concatenating the Datadog and public datasets for training, we also implement a variate shuffling strategy to maintain diversity and representation. Specifically, we randomly combine variates from either Datadog, open source datasets (GIFT-Eval pretrain and Chronos datasets), and/or synthetic data with a probability of 14%, thus creating new, diverse combinations of data points. We shuffle series with adjacent indices (batched by 32 variates), favoring data points that were closer together in the original datasets. This approach improves the model's ability to generalize across different types of data effectively.

## C BOOM

### C.1 Domain taxonomy

BOOM data is collected using the same query language as described in Section B.1. Each metric query yields a collection of time series—one per unique attribute combination—resulting in multivariate time series with attributes serving as variates. The distribution of aggregation metrics collected is described in Table 7. In Table 8, we categorize the series into 5 major groups based on the domain described within the query. As part of the labeling process, a large language model was used to pre-fill labels from the metric names, which were then manually reviewed by human annotators to ensure consistency and accuracy.

Metric Type	Description	Proportion (%)
<b>Gauge</b>	Last measurement reported within intervals	65.7
<b>Rate</b>	Number of event occurrences per second	26.8
<b>Distribution</b>	Aggregated statistical summaries across sources (e.g., average, percentiles)	5.3
<b>Count</b>	Total number of event occurrences within intervals	2.2

Table 7: Taxonomy of metric types in the benchmark dataset with their relative proportions.

Each time series in the BOOM undergoes a standardized preprocessing pipeline designed to address the noise, irregularities, and high cardinality typical of observability data. First, missing intervals, common in telemetry due to irregular metric emission, are filled using metric-aware strategies: count-based metrics are zero-filled under the assumption that missing values reflect inactivity, while real-valued metrics (e.g., rates or gauges) are linearly interpolated. Following imputation, series are sliced into fixed-length windows of up to 16,384 points. Unlike the training set, which uses random offsets and padding to augment diversity, the benchmark slices are extracted without offset or



System Domain	Description	Proportion (%)
Application Usage	Covers application interactions and user activity (e.g., request rates, API calls)	41.3
Infrastructure	System-level metrics (e.g., CPU usage, memory consumption)	34.4
Database	Focuses on database efficiency (e.g., query latency)	29.3
Networking	Encompasses network behavior, including bandwidth usage or latency	10.0
Security	Relates to authentication, intrusion attempts, or compliance checks	0.3

Table 8: Taxonomy of system domains in the benchmark dataset with their relative proportions. A single time series can belong to multiple domains;

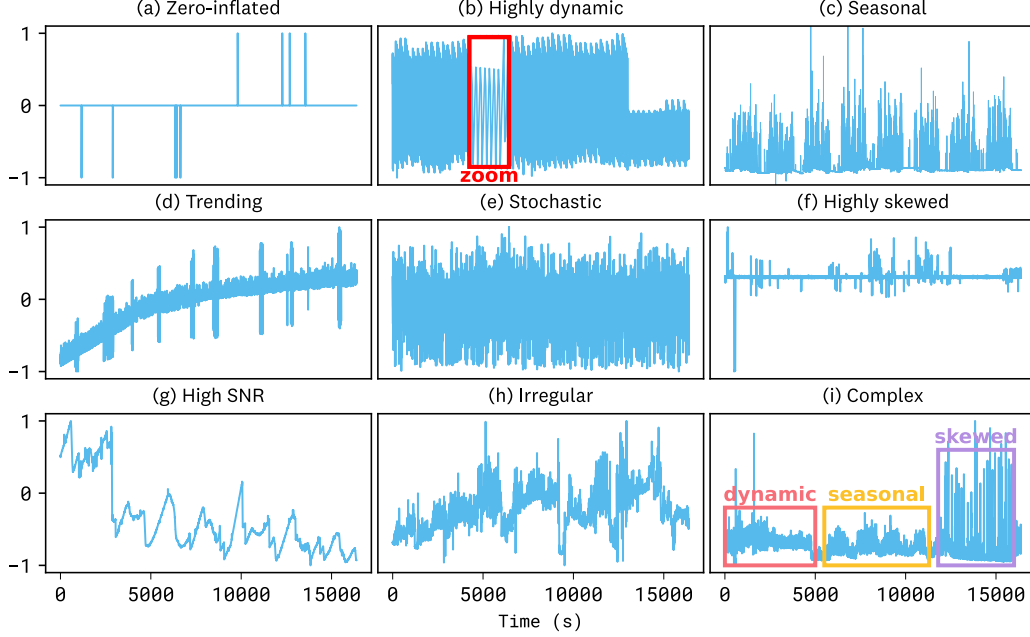


Figure 10: Representative examples from the BOOM, illustrating the unique temporal patterns associated observability data.

padding to ensure quality and comparability. To avoid data leakage in the provided validation split, only one slice is selected per metric, randomly sampled and similar for all the groups. For metrics with more than 100 variates (groups), we randomly subsample 100 to cap input dimensionality. Series are then normalized using the first 90% split of points that will be used for the context windows. We further filter out variates exhibiting abnormal scale change in the test split (final 10% of points) while having constant values in the context window, as these result in degenerate cases that are impossible to forecast meaningfully. This preprocessing results in high-quality, variable-length slices that preserve the structural diversity and challenges of real-world observability time series.

The final benchmark comprises 350 million points across 2,807 metric queries. These series vary widely in sampling frequency, temporal length, and number of variates. Figure 9 illustrates the distribution of series frequencies (left), lengths (middle), and cardinalities (right).

## C.2 Evaluation protocol

### C.2.1 Prediction terms and evaluation windows

Following the GIFT-Eval protocol, we assign a fixed prediction horizon to each time series based on its intrinsic sampling frequency. The specific mapping from frequency to default horizon length is provided in Table 9. This default value defines the *short-term* prediction task.

To define *medium-* and *long-term* prediction tasks, we scale the *short-term* horizon by factors of 10 and 15, respectively. These extended horizons are only applied when they fit entirely within the

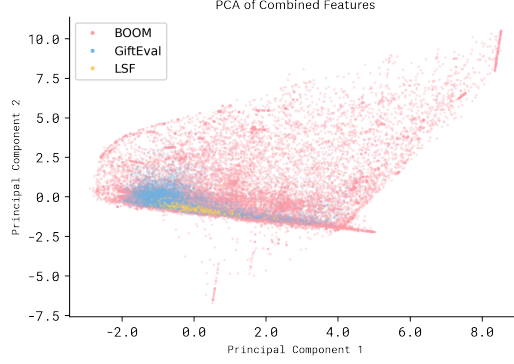


Figure 11: 2D Principal Component Analysis [93] projection of normalized statistical features computed from the Datadog Benchmark and GIFT-Eval datasets. The clear separation between the two distributions highlights a significant shift in underlying time series characteristics.

Frequency	Monthly (M)	Weekly (W)	Daily (D)	Hourly (H)	Minutely (T)	Secondly (S)
Prediction Length (steps)	12	8	30	48	48	60

Table 9: Mapping from time series frequency to default short-term prediction length.

test window, defined as the final 10% of the time series. This strategy enables evaluation across increasingly challenging forecasting ranges.

For each prediction term, series are evaluated using a rolling, non-overlapping window scheme, where each window has a length equal to the corresponding prediction horizon. This ensures full coverage of the test split while avoiding overlapping forecasts.

The evaluation is hierarchical: if a series qualifies for long-term forecasting (i.e., has enough test points to accommodate the longest horizon), it is also evaluated under the medium- and short-term settings. In total, this procedure yields 7,416 evaluation instances across the 2,807 time series in the benchmark. Each instance corresponds to the average performance over multiple rolling windows within a series for a specific prediction term.

### C.2.2 Evaluation metrics

Following GIFT-Eval, our two main metrics of forecasting accuracy are Mean Absolute Scaled Error (MASE) and Continuous Ranked Probability Score (CRPS).

MASE [77], a point-forecasting score, is defined as

$$\text{MASE} = \frac{\text{MAE}_{\text{model}}}{\text{MAE}_{\text{seasonal naive in-sample}}} \quad (12)$$

with the in-sample Seasonal Naive MAE being defined as

$$\text{MAE}_{\text{seasonal naive in-sample}} = \frac{1}{T-m} \sum_{t=m+1}^T Y_t - Y_{t-m} \quad (13)$$

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 (typically defined based upon a lookup table according to the time series frequency).

CRPS [78] is scoring rule for probabilistic forecasts, defined with respect to an observation  $y$

$$\text{CRPS}(D, y) = \int_{\mathbb{R}} (F_D(x) - H(x - y))^2 dx \quad (14)$$

where  $F_D$  is the cumulative distribution function of the forecast distribution  $D$  and  $H$  is the Heaviside step function. We follow the standard practice of taking the mean weighted quantile loss as a discrete approximation, as described by Park et al. [79].

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

### C.2.3 Inference procedures

To evaluate the comparison models on BOOM, we closely follow the evaluation methodology used in the GIFT-Eval implementation. For models not included in GIFT-Eval, we rely on their official implementations and recommended evaluation procedures. All foundation models are evaluated using a unified context length of 2048. This choice is informed by preliminary experiments showing that a shorter context length (512) leads to a general degradation in performance across models. Therefore, we opt for a relatively large context window (2048) to preserve forecast quality, while ensuring feasibility on available hardware.

To evaluate the zero-shot performance of other foundation models on BOOM, we follow the sampling procedures outlined in their respective manuscripts. For Chronos, we generate 20 samples and use the median prediction as the point forecast. For Moirai, we generate 100 samples, again taking the median, and set the patch size to “auto”. For TOTO we generate 256 samples and take the median as the point forecast. TimesFM produces only point forecasts of the mean, which we use directly. In all cases, we compute CRPS with respect to the probabilistic samples and MASE with respect to the point forecast. Since TimesFM and Chronos support only univariate forecasting, we evaluate each variate independently. In contrast, both Moirai and TOTO support joint prediction over groups of related variates.

For the three statistical baselines—AutoARIMA, AutoTheta, and AutoETS—we use the default hyperparameter settings from the statsforecast package, with one exception: for AutoARIMA, we reduce  $max_d$  and  $max_D$  from 2 to 1 due to frequent numerical instability when  $d = D = 2$ . Following GIFT-Eval, we set the maximum input length for all statistical models to 1000.

### C.2.4 Aggregation of results

Common practice for aggregating normalized benchmarking statistics is to use the geometric mean [94]. We adopt this approach, with a slight caveat: due to the presence of constant subsequences in observability time series, it’s possible for models to achieve zero error on a handful of series in BOOM. As zeros cause geometric means to collapse, we instead use the shifted geometric mean, which is stable in such scenarios [80, 81], for aggregating MASE and CRPS. Specifically, for a metric  $m$  across a set of test instances  $N$ , the shifted geometric mean  $\bar{m}_{\text{ShiftedGeom}}$  is defined as

$$\bar{m}_{\text{ShiftedGeom}} = \exp \left( \frac{1}{|N|} \sum_{n \in N} \log(m_n + \varepsilon) \right) + \varepsilon \quad (15)$$

where  $\varepsilon = 1\text{e-}5$  is a small stabilizing constant.

Evaluating forecasts on observability data using these standard evaluation metrics introduces several numerical instabilities which we must also handle:

**NaN values:** The benchmark includes a small number of zero-inflated series, which can result in invalid CRPS values due to division-by-zero errors. To mitigate this, we apply a generic postprocessing strategy where NaN and infinite values are imputed using the mean CRPS across the remaining series.

**Extremely low naive errors:** Certain flat series exhibit extremely low or even null in-sample MAE, leading to instability when normalizing by the seasonal naive MAE, with potentially exploding normalized values. Rather than discarding these special cases — which are informative for evaluating how models handle anomalous flat behavior — we isolate them into a separate data split. This split is evaluated using simple MAE instead of MASE and non-normalized CRPS. To define this subset, we apply an objective criterion: we include all series where either the MAE for the Seasonal Naive forecast on the test set or the in-sample Seasonal Naive MAE [13] are zero. If any prediction window of any variate within a query satisfies the above conditions, we assign the entire query to the second,

Metric	Toto	Moirai <sub>Small</sub>	Moirai <sub>Base</sub>	Moirai <sub>Large</sub>	TimesFM <sub>2.0</sub>	Chronos-Bolt <sub>Small</sub>	Chronos-Bolt <sub>Base</sub>	Timer	Time-MoE	VisionTS
MAE ↓	0.001	0.001	0.000	0.001	0.014	0.003	0.003	0.001	0.001	0.001
CRPS ↓	0.025	0.009	0.003	0.005	0.091	0.022	0.019	0.005	0.005	0.009

Table 10: Performance of Toto and other zero-shot models on the BOOM dataset using the subset of near-constant series.

low-variability set, where metrics are computed without normalization. For evaluation, results in this subset are not normalized and thus are aggregated using a simple arithmetic mean.

The outcomes for this subset are reported in Table 10. All the zero-shot models evaluated seem to handle these cases trivially; they all have extremely low MASE. Notably, TOTO’s CRPS is slightly elevated relative to the other models. We observe that TOTO seems to produce overly wide prediction intervals in at least some of these flat series, and conjecture that this may be due to the frequent presence of large anomalies in its pretraining data. This is an interesting avenue for further study, as this conservatism may in fact be desirable in downstream anomaly detection use cases where overconfident predictions can lead to false positive detections.

**Taxonomy breakdowns.** For the BOOM, we evaluate model performance across stratified groups defined by the dataset’s taxonomy categories. For term (Table 12), metric type (Table 13), and domain (Table 14), we report the shifted geometric mean of the evaluation metric within each group. These results are discussed in detail in Section D.1

### C.3 BOOMLET

To facilitate research in settings where it is computationally infeasible to evaluate on the full BOOM benchmark, we construct a smaller, representative subset denoted as BOOMLET. This subset is created via uniform sampling over metric queries in BOOM, while additionally prioritizing queries with a relatively large number of variates to ensure sufficient training signal. This selection strategy preserves the distributional properties across key taxonomy dimensions, while also maintaining a practical volume of training data per query. BOOMLET comprises 32 metric queries, encompassing 1,627 variates and approximately 23 million observation points.

## D Results

### D.1 BOOM

In Fig. 12, we present qualitative comparisons across three representative forecasting scenarios to highlight the behavioral differences between TOTO, Chronos, and Moirai. In the first example ①, features a highly stochastic signal interwoven with complex seasonal components. While Moirai and Chronos models tend to overfit short-term fluctuations—resulting in jagged forecasts and unstable confidence intervals—TOTO effectively identifies and extrapolates the latent seasonal structure, yielding smoother, more coherent trajectories and uncertainty bands that reflect a deeper structural understanding of the series dynamics. Example ② the target signal exhibits high dynamism with rapidly oscillating structure and sustained amplitude modulations—posing a challenge for long-range temporal modeling. While both Moirai and Chronos models progressively lose phase alignment and dampen their amplitude estimates, TOTO consistently maintains sharp, temporally aligned forecasts with well-calibrated uncertainty, accurately tracking the intricate periodic structure far into the forecast horizon. Finally, example ③, the target series is characterized by sparse, bursty impulses with high variance across events. Here, although TOTO’s mean prediction does not always precisely capture individual peaks, its predictive distribution faithfully mirrors the underlying spikiness of the series, in stark contrast to Chronos, which collapses to an overconfident flat trajectory.

Table 11 reports the results for all versions and sizes of the zero-shot models.

Dataset	Metric	Toto	Moirai <sub>Small</sub>	Moirai <sub>Base</sub>	Moirai <sub>Large</sub>	TimesFM <sub>2.0</sub>	Chronos-Bolt <sub>Small</sub>	Chronos-Bolt <sub>Base</sub>	Timer	Time-MoE <sub>BOOM</sub>	Time-MoE <sub>200M</sub>	VisionTS	DLinear	DeepAR	Naive
BOOM	MASE ↓	0.617	0.729	0.710	0.720	0.725	0.733	0.726	0.796	0.806	0.881	0.988	-	-	1.000
	CRPS ↓	0.375	0.442	0.428	0.436	0.447	0.455	0.451	0.639	0.649	0.643	0.673	-	-	1.000
	Rank ↓	2.369	4.905	4.328	4.561	5.243	5.927	5.576	9.920	9.877	9.843	10.989	-	-	12.631
BOOMLET	MASE ↓	0.617	0.786	0.779	0.767	0.685	0.717	0.711	0.807	0.810	0.793	0.912	0.823	0.883	1.000
	CRPS ↓	0.519	0.631	0.630	0.621	0.603	0.642	0.637	0.793	0.788	0.780	0.885	0.641	0.697	1.000
	Rank ↓	1.300	5.711	5.300	4.967	4.867	6.756	6.511	11.544	11.222	11.189	13.589	6.056	7.900	14.133

Table 11: **BOOM results.** Full results across all models evaluated from Table 2 Key: **Best results**, Second-best results.

To better understand the capabilities and limitations of different forecasting models, we conduct a disaggregated evaluation across four major characteristics that describe time series in the BOOM dataset. This analysis enables us to probe how models respond to structural diversity in real-world time series data.

Across all three categorical axes, the TOTO consistently achieves the lowest CRPS, with strong margins over all baselines.

Real Term	Metric	Toto	Moirai <sub>Small</sub>	Moirai <sub>Base</sub>	Moirai <sub>Large</sub>	TimesFM <sub>2.0</sub>	Chronos-Bolt <sub>Small</sub>	Chronos-Bolt <sub>Base</sub>	Timer	Time-MoE <sub>Base</sub>	Time-MoE <sub>Large</sub>	VisionTS	Naive
Long	MASE ↓	<b>0.688</b>	0.795	0.780	0.799	0.817	0.813	0.798	0.809	0.886	0.950	1.026	1.000
	CRPS ↓	<b>0.424</b>	0.482	0.473	0.491	0.522	0.528	0.519	0.661	0.724	0.694	0.698	1.000
Medium	MASE ↓	<b>0.657</b>	0.771	0.753	0.770	0.780	0.782	0.782	0.804	0.866	0.929	1.011	1.000
	CRPS ↓	<b>0.406</b>	0.476	0.460	0.475	0.499	0.508	0.507	0.671	0.725	0.692	0.698	1.000
Short	MASE ↓	<b>0.535</b>	0.670	0.627	0.626	0.619	0.638	0.632	0.779	0.704	0.794	0.947	1.000
	CRPS ↓	<b>0.318</b>	0.399	0.370	0.369	0.359	0.368	0.365	0.597	0.541	0.570	0.640	1.000

Table 12: Performance comparison of TOTO and other zero-shot models across different **prediction terms**. MASE and CRPS are normalized by the Seasonal Naive forecast and aggregated across tasks using the shifted geometric mean. Key: **Best results**, Second-best results.

Type	Metric	Toto	Moirai <sub>Small</sub>	Moirai <sub>Base</sub>	Moirai <sub>Large</sub>	TimesFM <sub>2.0</sub>	Chronos-Bolt <sub>Small</sub>	Chronos-Bolt <sub>Base</sub>	Timer	Time-MoE <sub>Base</sub>	Time-MoE <sub>Large</sub>	VisionTS	Naive
Count	MASE ↓	0.687	0.814	0.795	0.813	0.919	0.883	0.880	0.663	<b>0.652</b>	1.035	1.220	1.000
	CRPS ↓	<b>0.317</b>	0.370	0.353	0.372	0.403	0.403	0.402	0.662	0.651	0.698	0.603	1.000
Distribution	MASE ↓	<b>0.658</b>	0.741	0.724	0.729	0.745	0.759	0.753	0.890	0.878	0.877	1.034	1.000
	CRPS ↓	<b>0.382</b>	0.434	0.422	0.428	0.440	0.452	0.446	0.608	0.604	0.596	0.674	1.000
Gauge	MASE ↓	<b>0.583</b>	0.720	0.686	0.700	0.706	0.706	0.696	0.721	0.760	0.890	0.922	1.000
	CRPS ↓	<b>0.382</b>	0.471	0.444	0.456	0.466	0.469	0.463	0.658	0.694	0.703	0.672	1.000
Rate	MASE ↓	<b>0.634</b>	0.753	0.728	0.733	0.726	0.742	0.739	0.864	0.846	0.862	1.041	1.000
	CRPS ↓	<b>0.369</b>	0.433	0.418	0.422	0.431	0.445	0.443	0.630	0.619	0.596	0.687	1.000

Table 13: Performance comparison of TOTO and other zero-shot models across different **metric types**. MASE and CRPS are normalized by the Seasonal Naive forecast and aggregated across tasks using the shifted geometric mean. Key: **Best results**, Second-best results.

Domain	Metric	Toto	Moirai <sub>Small</sub>	Moirai <sub>Base</sub>	Moirai <sub>Large</sub>	TimesFM <sub>2.0</sub>	Chronos-Bolt <sub>Small</sub>	Chronos-Bolt <sub>Base</sub>	Timer	Time-MoE <sub>Base</sub>	Time-MoE <sub>Large</sub>	VisionTS	Naive
Application usage	MASE ↓	<b>0.639</b>	0.747	0.721	0.730	0.736	0.748	0.748	0.871	0.863	0.884	1.042	1.000
	CRPS ↓	<b>0.378</b>	0.440	0.422	0.430	0.441	0.452	0.451	0.636	0.633	0.611	0.691	1.000
Database	MASE ↓	<b>0.635</b>	0.751	0.738	0.743	0.765	0.761	0.757	0.716	0.714	0.903	1.017	1.000
	CRPS ↓	<b>0.362</b>	0.429	0.414	0.418	0.440	0.444	0.441	0.619	0.618	0.633	0.647	1.000
Infrastructure	MASE ↓	<b>0.568</b>	0.692	0.650	0.670	0.679	0.678	0.663	0.728	0.791	0.847	0.863	1.000
	CRPS ↓	<b>0.391</b>	0.476	0.446	0.462	0.471	0.474	0.466	0.655	0.713	0.710	0.666	1.000
Networking	MASE ↓	<b>0.635</b>	0.795	0.786	0.773	0.765	0.779	0.757	0.871	0.856	0.933	1.035	1.000
	CRPS ↓	<b>0.400</b>	0.493	0.484	0.484	0.493	0.506	0.489	0.725	0.721	0.739	0.734	1.000
Security	MASE ↓	<b>0.682</b>	0.741	0.739	0.736	0.717	0.734	0.729	0.828	0.770	0.776	0.924	1.000
	CRPS ↓	<b>0.476</b>	0.505	0.504	0.504	0.525	0.539	0.535	0.664	0.625	0.629	0.735	1.000

Table 14: Performance comparison of TOTO and other zero-shot models across different **metric domains**. MASE and CRPS are normalized by the Seasonal Naive forecast and aggregated across tasks using the shifted geometric mean. Key: **Best results**, Second-best results.

### D.1.1 BOOMLET

We present results on the BOOMLET subset in Table [11](#).

## D.2 GIFT-Eval

To provide a comprehensive evaluation of Toto’s forecasting capabilities, we benchmarked our model on the GIFT-Eval benchmark [\[26\]](#). GIFT-Eval is a collection of diverse time series datasets that covers a wide range of domains and characteristics, including:

- Various frequencies (hourly, daily, weekly, monthly, yearly)
- Different domains (energy, traffic, retail, economics, etc.)
- Varying series lengths and forecasting horizons
- Single and multiple seasonality patterns

The benchmark evaluates models using multiple metrics, with particular emphasis on:

- Mean Absolute Scaled Error (MASE): Measures point forecast accuracy relative to a naive forecast
- Continuous Ranked Probability Score (CRPS): Evaluates the quality of probabilistic forecasts
- Overall Rank: Aggregates performance across all datasets

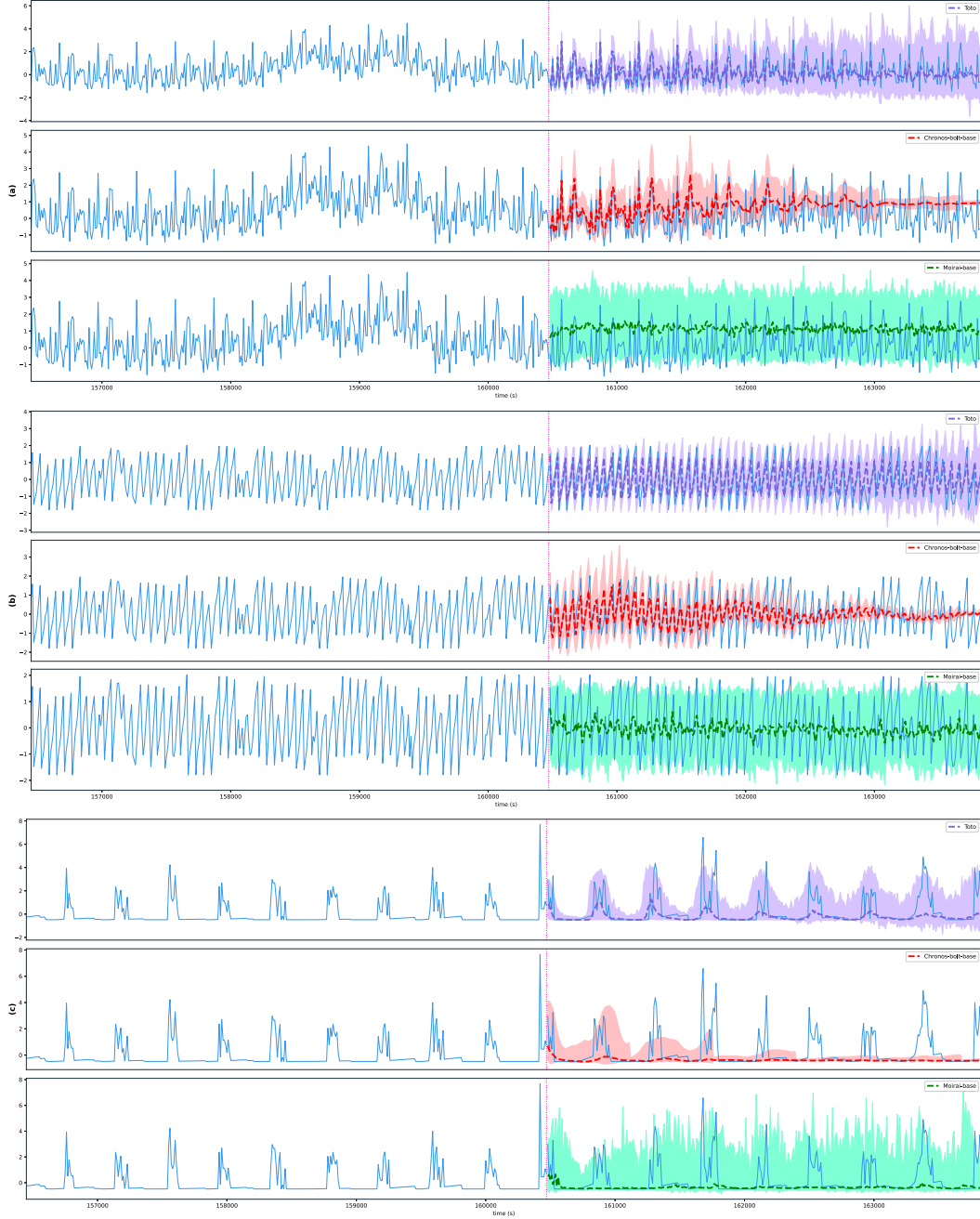


Figure 12: Example of 336-step zero-shot comparative forecasts on the Boom, showing multivariate probabilistic predictions. Solid lines represent ground truth, dashed lines represent median point forecasts, and shaded regions represent 95% prediction intervals.

For models other than TOTO, we report the published numbers from the public leaderboard [3]. For TOTO, we use the same inference settings described in Section C.2.3 with one modification: as GIFT-Eval does impose a maximum context length, we select a length of 4096 (the native context length used in TOTO’s pretraining, as described in Table 6).

### D.3 LSF

In addition to our primary evaluations, we also assess the model’s performance on the Long Sequence Forecasting (LSF) benchmark datasets—ETTh1, ETTh2, ETTm1, ETTm2, Electricity, and Weather [2]. As noted by

Aksu et al. [1], these datasets are limited in size and diversity, and recent findings [29] suggest that strong supervised baselines can already perform near the upper bound on such benchmarks. This may indicate a saturation point where further gains from foundation models are difficult to observe, rather than a fundamental limitation of the models themselves. Nevertheless, as it remains a widely used legacy benchmark in the literature, we report zero-shot results of TOTO on it to maintain consistency with established practices in the field.

Furthermore we leverage its small scale and constrained use-cases to examine TOTO’s capacity to transfer to new datasets and specialized domains by conducting fine-tuning experiments on the training splits of its datasets.

Following standard practice for the LSF benchmark, we report normalized Mean Absolute Error (MAE) and Mean Squared Error (MSE), in order to be able to compare performance across different datasets. We evaluate using forecast lengths of 96, 192, 336, and 720 time steps. Predictions are generated using sliding windows with a stride of 1. For the Electricity dataset, however, we use a stride equal to the prediction length to reduce computational resource requirements. The results are then averaged. We compare TOTO’s performance with results reported by recent state-of-the-art time series foundation models, including Moirai [13], VisionTS [24], TimesFM [12], Time-MoE [14], TimeLLM [95], GPT4TS [96], xLSTMTIME [97] and other models evaluated in Woo et al. [13] and Das et al. [12]. We display zero-shot and full-shot TOTO results in Table 4 and Table 15 respectively. We also provide additional per prediction length results in Table 16 and Table 17.

Table 4 shows that TOTO consistently delivers the best overall performance across all datasets, achieving the lowest average MAE and MSE, and outperforming other zero-shot baselines on 8 out of 12 evaluation metrics. Its performance is especially strong on ETTm2, Electricity, and Weather, where it continues to excel even in zero-shot scenarios.

Zero Shot										Full Shot									
Dataset	Metric	Total	Total <sub>rr</sub>	TimeLLM	GPT4TS	VisionTS <sub>rr</sub>	Time-MoE <sub>BaseFT</sub>	Time-MoE <sub>LargeFT</sub>	Time-MoE <sub>UltraFT</sub>	TimesFM*	xLSTMTime	tTransformer	TimesNet	PatchTST	Crossformer	TIDE	DLinear	SCINet	FEDformer
ETTh1	MAE ↓	0.413	0.409	0.423	0.426	0.409	0.406	<b>0.404</b>	0.406	0.426	0.428	0.448	0.450	0.455	0.522	0.507	0.452	0.647	0.460
	MSE ↓	0.435	0.415	0.408	0.427	0.395	0.379	0.375	<b>0.373</b>	-	0.408	0.454	0.458	0.469	0.529	0.541	0.456	0.747	0.440
ETT2	MAE ↓	0.363	<b>0.363</b>	0.383	0.394	0.382	0.386	0.386	<b>0.380</b>	0.410	0.386	0.407	0.407	0.407	0.684	0.550	0.515	0.723	0.449
	MSE ↓	0.340	0.359	<b>0.334</b>	0.354	0.336	0.346	0.361	<b>0.334</b>	-	0.346	0.383	0.414	0.387	0.942	0.611	0.559	0.954	0.437
ETTm1	MAE ↓	0.378	<b>0.357</b>	0.372	0.383	0.367	0.381	0.371	0.373	0.388	0.371	0.410	0.400	0.400	0.495	0.419	0.407	0.481	0.452
	MSE ↓	0.396	0.349	0.329	0.352	0.338	0.345	<b>0.322</b>	0.329	-	0.347	0.407	0.400	0.387	0.513	0.419	0.403	0.486	0.448
ETTm2	MAE ↓	0.303	<b>0.291</b>	0.313	0.326	0.319	0.335	0.332	0.334	0.334	0.310	0.332	0.333	0.326	0.611	0.404	0.401	0.537	0.349
	MSE ↓	0.267	<b>0.244</b>	0.251	0.266	0.261	0.271	0.284	0.277	-	0.254	0.288	0.291	0.281	0.757	0.358	0.350	0.571	0.305
Electricity	MAE ↓	0.243	<b>0.234</b>	0.252	0.263	0.249	-	-	-	-	0.250	0.270	0.295	0.304	0.334	0.344	0.300	0.365	0.327
	MSE ↓	0.161	<b>0.152</b>	0.158	0.167	0.156	-	-	-	-	0.157	0.178	0.193	0.216	0.244	0.252	0.212	0.268	0.214
Weather	MAE ↓	0.245	<b>0.233</b>	0.257	0.270	0.262	0.275	0.273	0.280	-	0.255	0.278	0.287	0.281	0.315	0.320	0.317	0.363	0.360
	MSE ↓	0.224	<b>0.206</b>	0.225	0.237	0.227	0.236	0.234	0.250	-	0.222	0.258	0.259	0.259	0.259	0.271	0.265	0.292	0.309
Mean	MAE ↓	0.324	<b>0.314</b>	0.333	0.344	0.331	-	-	-	-	0.333	0.358	0.378	0.362	0.494	0.424	0.399	0.519	0.400
	MSE ↓	0.304	<b>0.284</b>	<b>0.284</b>	0.300	0.286	-	-	-	-	0.289	0.328	0.333	0.333	0.541	0.409	0.374	0.553	0.359
Best Count		8	1	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0

Table 15: Full-Shot comparison of models on the LSF benchmark, with TOTO’s Zero-Shot result in the first data column.\*TimesFM only reports values for MAE on ETTh1, ETTh2, ETTm1, and ETTm2 after fine-tuning.

Key: **Best results**, Second-best results. “Best Count” row reports the number of times each model attains the best result for a given dataset-metric pair.



Furthermore, Table 15 shows that even when starting from a strong SOTA baseline, TOTO’s performance improves with fine-tuning, showing it can achieve full-shot SOTA results and adapt to new domains with limited data. This highlights TOTO’s robustness and versatility as a foundation model for a wide range of time-series forecasting tasks.

**Full-shot results on LSF benchmarks** We conduct fine-tuning experiments on Toto following similar procedure delineated by [98] and [13]. The full-shot results for each dataset, comparing fine-tuned and zero-shot performance, are reported in Table 15.

**Results** Our experimental results demonstrate that when finetuned, denoted as TOTO<sub>FT</sub>, achieves state-of-the-art performance on 3 out of 6 datasets in the LSF benchmark—specifically, ETTm2, Electricity, and Weather—where it outperforms all other models on both MAE and MSE metrics. Additionally, TOTO<sub>FT</sub> achieves the best MAE score on ETTm1 and ETTh2, although it does not lead on MSE for those datasets. Compared to its zero-shot counterpart, TOTO<sub>FT</sub> consistently improves both MAE and MSE metrics across most datasets, with particularly notable gains in ETTm1 (MAE: 0.378  $\rightarrow$  0.357, MSE: 0.396  $\rightarrow$  0.349) and ETTm2 (MAE: 0.303  $\rightarrow$  0.291, MSE: 0.267  $\rightarrow$  0.244). Overall, TOTO<sub>FT</sub> ranks first in 8 out of 12 metric-dataset pairs, outperforming all other models, including both zero-shot and full-shot baselines. Notably, it also delivers the best overall performance on the benchmark, achieving the lowest average MAE (0.314) and MSE (0.284). These results underscore the effectiveness of fine-tuning in enhancing Toto’s predictive performance, establishing TOTO<sub>FT</sub> as the new SOTA model on the LSF benchmark. In addition, this demonstrates that Toto is a robust foundation model, adaptable to a wide range of downstream datasets, including those from entirely new domains, making it a versatile choice for time-series forecasting tasks.

A closer examination of the results reveals that while Toto<sub>FT</sub> achieves state-of-the-art performance on most datasets, the effectiveness of fine-tuning varies across them. Fine-tuning proves especially beneficial on ETTm1, ETTm2, and Weather, where it significantly enhances model predictions. In contrast, the improvements on ETTh1 are more modest, and for ETTh2, fine-tuning yields no notable gains—potentially due to the relatively small size of these datasets. Moreover, even though fine-tuning generally improves performance over the original TOTO model, TOTO<sub>FT</sub> does not outperform other full-shot models on ETTh1.

Additional details on zero-shot and full-shot results per prediction length are displayed in Table 17.

Zero Shot										
Dataset	Prediction Length	Metric	Toto	Moirai <sub>Small</sub>	Moirai <sub>Base</sub>	Moirai <sub>Large</sub>	VisionTS	TIME-MoE <sub>Base</sub>	TIME-MoE <sub>Large</sub>	TIME-MoE <sub>Ultra</sub>
ETTh1	96	MAE ↓	<u>0.381</u>	0.402	0.402	0.398	0.383	<u>0.381</u>	0.382	<b>0.379</b>
		MSE ↓	0.382	0.375	0.384	0.380	0.353	0.357	<u>0.350</u>	<b>0.349</b>
	192	MAE ↓	<u>0.408</u>	0.419	0.429	0.434	0.410	<b>0.404</b>	0.412	0.413
		MSE ↓	0.428	0.399	0.425	0.440	0.392	<b>0.384</b>	<u>0.388</u>	0.395
	336	MAE ↓	<b>0.422</b>	0.429	0.450	0.474	<u>0.423</u>	0.434	0.430	0.453
		MSE ↓	0.457	0.412	0.456	0.514	<b>0.407</b>	<u>0.411</u>	<u>0.411</u>	0.447
	720	MAE ↓	<b>0.440</b>	0.444	0.473	0.568	<u>0.441</u>	0.477	0.455	0.462
		MSE ↓	0.472	<u>0.413</u>	0.470	0.705	<b>0.406</b>	0.449	0.427	0.457
ETTh2	96	MAE ↓	<b>0.310</b>	0.334	0.327	<u>0.325</u>	0.328	0.359	0.354	0.352
		MSE ↓	<u>0.273</u>	0.281	0.277	<u>0.287</u>	<b>0.271</b>	0.305	0.302	0.292
	192	MAE ↓	<b>0.356</b>	0.373	0.374	<u>0.367</u>	<u>0.367</u>	0.386	0.385	0.379
		MSE ↓	<u>0.339</u>	0.340	0.340	0.347	<b>0.328</b>	0.351	0.364	0.347
	336	MAE ↓	<u>0.387</u>	0.393	0.401	0.393	<b>0.381</b>	0.418	0.425	0.419
		MSE ↓	0.374	<u>0.362</u>	0.371	0.377	<b>0.345</b>	0.391	0.417	0.406
	720	MAE ↓	<b>0.400</b>	<u>0.416</u>	0.426	0.421	0.422	0.454	0.496	0.447
		MSE ↓	<b>0.375</b>	<u>0.380</u>	0.394	0.404	0.388	0.419	0.537	0.439
ETTm1	96	MAE ↓	<b>0.333</b>	0.383	0.360	0.363	0.347	0.368	0.357	<u>0.341</u>
		MSE ↓	0.320	0.404	0.335	0.353	0.341	0.338	<u>0.309</u>	<b>0.281</b>
	192	MAE ↓	0.364	0.402	0.379	0.380	<u>0.360</u>	0.388	0.381	<b>0.358</b>
		MSE ↓	0.371	0.435	0.366	0.376	0.360	0.353	<u>0.346</u>	<b>0.305</b>
	336	MAE ↓	<u>0.388</u>	0.416	0.394	0.395	<b>0.374</b>	0.413	0.408	0.395
		MSE ↓	0.408	0.462	0.391	0.399	0.377	0.381	<u>0.373</u>	<b>0.369</b>
	720	MAE ↓	0.426	0.437	0.419	<u>0.417</u>	<b>0.405</b>	0.493	0.477	0.472
		MSE ↓	0.485	0.490	0.434	<u>0.432</u>	<b>0.416</b>	0.504	0.475	0.469
ETTm2	96	MAE ↓	<b>0.237</b>	0.282	0.269	<u>0.260</u>	0.282	0.291	0.286	0.288
		MSE ↓	<b>0.172</b>	0.205	0.195	<u>0.189</u>	0.228	0.201	0.197	0.198
	192	MAE ↓	<b>0.280</b>	0.318	0.303	<u>0.300</u>	0.305	0.334	0.322	0.312
		MSE ↓	<b>0.232</b>	0.261	0.247	0.247	0.262	0.258	0.250	<u>0.235</u>
	336	MAE ↓	<b>0.320</b>	0.355	0.333	0.334	<u>0.328</u>	0.373	0.375	0.348
		MSE ↓	<b>0.290</b>	0.319	<u>0.291</u>	0.295	0.293	0.324	0.337	0.293
	720	MAE ↓	<u>0.375</u>	0.410	0.377	0.386	<b>0.370</b>	0.464	0.461	0.428
		MSE ↓	0.372	0.415	<u>0.355</u>	0.372	<b>0.343</b>	0.488	0.480	0.427
Electricity	96	MAE ↓	<b>0.213</b>	0.299	0.248	<u>0.242</u>	0.266	-	-	-
		MSE ↓	<b>0.129</b>	0.205	0.158	<u>0.152</u>	0.177	-	-	-
	192	MAE ↓	<b>0.229</b>	0.310	0.263	<u>0.259</u>	0.277	-	-	-
		MSE ↓	<b>0.145</b>	0.220	0.174	<u>0.171</u>	0.188	-	-	-
	336	MAE ↓	<b>0.247</b>	0.323	<u>0.278</u>	<u>0.278</u>	0.296	-	-	-
		MSE ↓	<b>0.163</b>	0.236	<u>0.191</u>	0.192	0.207	-	-	-
	720	MAE ↓	<b>0.282</b>	0.347	<u>0.307</u>	0.313	0.337	-	-	-
		MSE ↓	<b>0.206</b>	0.270	<u>0.229</u>	0.236	0.256	-	-	-
Weather	96	MAE ↓	<b>0.179</b>	0.212	<u>0.203</u>	0.208	0.257	0.214	0.213	0.211
		MSE ↓	<b>0.149</b>	0.173	0.167	0.177	0.220	0.160	0.159	<u>0.157</u>
	192	MAE ↓	<b>0.223</b>	0.250	<u>0.241</u>	0.249	0.275	0.260	0.266	0.256
		MSE ↓	<b>0.192</b>	0.216	0.209	0.219	0.244	0.210	0.215	<u>0.208</u>
	336	MAE ↓	<b>0.265</b>	0.282	<u>0.276</u>	0.292	0.299	0.309	0.322	0.290
		MSE ↓	<b>0.245</b>	0.260	<u>0.256</u>	0.277	0.280	0.274	0.291	<u>0.255</u>
	720	MAE ↓	<b>0.312</b>	<u>0.322</u>	0.323	0.350	0.337	0.405	0.400	0.397
		MSE ↓	<b>0.310</b>	<u>0.320</u>	0.321	0.365	0.330	0.418	0.415	0.405
Best Count			29	0	0	0	11	2	0	6

Table 16: Zero-Shot Comparison of different models with TOTO on the LSF benchmark datasets for each prediction length. Non-TOTO values are reproduced from published tables.

Key: **Best results**, Second-best results. “Best Count” row reports the number of times each model attains the best result for a given metric.

Dataset	Prediction Length	Metric	Zero Shot										Full Shot									
			TotalT	TimeLM	GPT4TS	VisionTS <sub>T</sub>	Time-MoE <sub>BaseFT</sub>	Time-MoE <sub>LangFT</sub>	Time-MoE <sub>UnitFT</sub>	TimesFM*	xLSTMTIME	Transformer	TimesNet	PatchTST	Crossformer	TiDE	DLinear	SCINet	FEDformer			
ETTm1	96	MAE↓	0.381	0.374	0.392	0.397	0.376	0.373	0.371	0.373	0.398	0.395	0.405	0.402	0.419	0.448	0.464	0.400	0.599	0.419		
		MSE↓	0.382	0.364	0.362	0.376	0.347	0.345	0.335	0.363	0.363	0.368	0.386	0.384	0.414	0.423	0.479	0.386	0.654	0.376		
	192	MAE↓	0.408	0.402	0.418	0.418	0.400	0.396	0.400	0.396	0.424	0.416	0.436	0.429	0.445	0.474	0.492	0.432	0.631	0.448		
	336	MAE↓	0.428	0.409	0.398	0.416	0.385	0.372	0.374	0.372	0.359	0.401	0.441	0.436	0.460	0.471	0.525	0.437	0.719	0.420		
ETTm2	96	MAE↓	0.422	0.418	0.427	0.433	0.415	0.412	0.412	0.418	0.436	0.437	0.458	0.469	0.466	0.546	0.515	0.459	0.659	0.465		
		MSE↓	0.457	0.436	0.430	0.442	0.407	0.389	0.390	0.388	0.481	0.422	0.487	0.491	0.501	0.570	0.565	0.481	0.778	0.459		
	192	MAE↓	0.440	0.440	0.457	0.456	0.443	0.443	0.443	0.443	0.445	0.465	0.491	0.500	0.488	0.621	0.558	0.516	0.699	0.507		
	720	MAE↓	0.472	0.454	0.442	0.477	0.439	0.410	0.402	0.425	0.445	0.441	0.503	0.521	0.500	0.653	0.594	0.519	0.856	0.506		
ETTm2	96	MAE↓	0.310	0.309	0.328	0.342	0.328	0.340	0.335	0.338	0.356	0.333	0.349	0.374	0.348	0.584	0.440	0.387	0.621	0.397		
		MSE↓	0.273	0.272	0.268	0.285	0.269	0.276	0.278	0.274	0.274	0.273	0.297	0.340	0.302	0.745	0.400	0.333	0.707	0.358		
	192	MAE↓	0.356	0.355	0.375	0.389	0.374	0.371	0.373	0.370	0.400	0.378	0.400	0.414	0.400	0.656	0.509	0.476	0.689	0.439		
	336	MAE↓	0.387	0.338	0.329	0.354	0.332	0.331	0.345	0.330	0.428	0.340	0.380	0.402	0.388	0.877	0.528	0.477	0.860	0.429		
ETTm1	96	MAE↓	0.374	0.372	0.368	0.373	0.351	0.373	0.384	0.362	0.428	0.373	0.428	0.452	0.426	1.043	0.643	0.594	1.000	0.496		
		MSE↓	0.434	0.432	0.368	0.373	0.361	0.364	0.364	0.362	0.457	0.375	0.428	0.426	0.426	1.043	0.643	0.657	0.838	0.474		
	192	MAE↓	0.400	0.400	0.430	0.431	0.411	0.411	0.431	0.411	0.457	0.430	0.457	0.457	0.456	1.043	0.643	0.657	0.838	0.474		
	720	MAE↓	0.430	0.430	0.430	0.430	0.430	0.430	0.430	0.430	0.430	0.430	0.430	0.430	0.430	1.043	0.643	0.657	0.838	0.474		
ETTm2	96	MAE↓	0.333	0.313	0.334	0.346	0.322	0.334	0.325	0.323	0.345	0.335	0.368	0.375	0.367	0.426	0.387	0.372	0.438	0.419		
		MSE↓	0.320	0.278	0.272	0.292	0.281	0.286	0.264	0.256	0.374	0.286	0.334	0.338	0.329	0.404	0.364	0.345	0.438	0.379		
	192	MAE↓	0.364	0.345	0.358	0.372	0.353	0.358	0.350	0.343	0.374	0.361	0.391	0.387	0.385	0.451	0.404	0.389	0.450	0.441		
	336	MAE↓	0.371	0.328	0.310	0.332	0.322	0.327	0.295	0.307	0.374	0.329	0.377	0.374	0.367	0.450	0.398	0.380	0.439	0.426		
ETTm2	96	MAE↓	0.388	0.368	0.384	0.394	0.389	0.390	0.371	0.376	0.397	0.379	0.420	0.411	0.410	0.515	0.425	0.413	0.485	0.459		
		MSE↓	0.408	0.364	0.352	0.366	0.356	0.354	0.323	0.326	0.436	0.358	0.426	0.410	0.399	0.532	0.428	0.413	0.490	0.445		
	192	MAE↓	0.426	0.403	0.411	0.421	0.413	0.445	0.435	0.435	0.436	0.411	0.459	0.450	0.439	0.589	0.461	0.453	0.550	0.490		
	720	MAE↓	0.456	0.426	0.383	0.417	0.391	0.433	0.409	0.454	0.478	0.416	0.491	0.478	0.454	0.666	0.487	0.474	0.595	0.543		
ETTm2	96	MAE↓	0.237	0.215	0.253	0.263	0.256	0.265	0.259	0.273	0.263	0.250	0.264	0.267	0.259	0.366	0.305	0.292	0.377	0.287		
		MSE↓	0.172	0.158	0.195	0.173	0.169	0.172	0.169	0.183	0.203	0.184	0.186	0.187	0.175	0.286	0.206	0.192	0.266	0.203		
	192	MAE↓	0.280	0.269	0.295	0.301	0.294	0.306	0.295	0.301	0.309	0.288	0.309	0.309	0.302	0.492	0.364	0.362	0.445	0.328		
	336	MAE↓	0.320	0.312	0.319	0.329	0.325	0.328	0.323	0.323	0.349	0.318	0.340	0.343	0.343	0.542	0.422	0.427	0.591	0.269		
Electricity	96	MAE↓	0.290	0.306	0.329	0.341	0.334	0.345	0.341	0.339	0.349	0.322	0.348	0.321	0.305	0.597	0.377	0.369	0.637	0.325		
		MSE↓	0.275	0.263	0.271	0.286	0.278	0.281	0.293	0.278	0.415	0.271	0.311	0.321	0.305	1.042	0.524	0.522	0.735	0.415		
	192	MAE↓	0.375	0.362	0.379	0.401	0.392	0.424	0.433	0.424	0.415	0.380	0.407	0.403	0.400	1.042	0.524	0.522	0.735	0.415		
	720	MAE↓	0.372	0.344	0.352	0.378	0.372	0.403	0.451	0.425	-	0.361	0.412	0.408	0.402	1.730	0.558	0.554	0.960	0.421		
Electricity	96	MAE↓	0.213	0.207	0.224	0.238	0.218	-	-	-	0.221	0.221	0.240	0.272	0.285	0.314	0.329	0.282	0.345	0.308		
		MSE↓	0.129	0.123	0.131	0.139	0.126	-	-	-	-	0.128	0.148	0.168	0.195	0.219	0.237	0.197	0.247	0.193		
	192	MAE↓	0.229	0.224	0.241	0.251	0.237	-	-	-	-	0.243	0.253	0.289	0.289	0.322	0.330	0.285	0.355	0.315		
	336	MAE↓	0.145	0.142	0.152	0.153	0.144	-	-	-	-	0.150	0.162	0.184	0.199	0.231	0.236	0.196	0.257	0.201		
Weather	96	MAE↓	0.247	0.239	0.248	0.266	0.256	-	-	-	0.259	0.259	0.269	0.300	0.305	0.337	0.344	0.301	0.369	0.329		
		MSE↓	0.163	0.155	0.160	0.169	0.162	-	-	-	-	0.166	0.178	0.198	0.215	0.246	0.249	0.209	0.269	0.214		
	192	MAE↓	0.282	0.266	0.298	0.297	0.286	-	-	-	-	0.276	0.317	0.320	0.337	0.363	0.373	0.333	0.390	0.355		
	720	MAE↓	0.206	0.187	0.192	0.206	0.192	-	-	-	-	0.185	0.225	0.220	0.256	0.280	0.284	0.245	0.299	0.246		
Weather	96	MAE↓	0.179	0.165	0.201	0.212	0.192	0.203	0.201	0.208	-	0.187	0.214	0.220	0.218	0.230	0.261	0.255	0.306	0.296		
		MSE↓	0.149	0.134	0.147	0.162	0.142	0.151	0.149	0.154	-	0.144	0.174	0.172	0.177	0.158	0.202	0.196	0.221	0.217		
	192	MAE↓	0.223	0.211	0.234	0.248	0.238	0.246	0.244	0.251	-	0.236	0.254	0.261	0.259	0.277	0.298	0.296	0.340	0.336		
	336	MAE↓	0.192	0.177	0.189	0.204	0.191	0.195	0.192	0.202	-	0.192	0.221	0.219	0.225	0.206	0.242	0.237	0.261	0.276		
Weather	96	MAE↓	0.265	0.253	0.279	0.286	0.282	0.288	0.285	0.297	-	0.272	0.296	0.306	0.297	0.335	0.335	0.335	0.378	0.380		
		MSE↓	0.245	0.225	0.262	0.254	0.246	0.247	0.245	0.252	-	0.237	0.278	0.280	0.278	0.272	0.287	0.283	0.309	0.339		
	192	MAE↓	0.312	0.302	0.316	0.337	0.337	0.366	0.365	0.376	-	0.326	0.349	0.359	0.348	0.418	0.386	0.381	0.428	0.428		
	720	MAE↓	0.310	0.288	0.304	0.326	0.328	0.352	0.352	0.392	-	0.313	0.358	0.365	0.354	0.398	0.351	0.345	0.377	0.403		

Table 17: Full-Shot Comparison of different models with TOTO on the LSF benchmark datasets for each prediction length, with TOTO’s Zero-Shot result in the first data column. Key: **Best results**, **Second-best results**. “Best Count” row reports the number of times each model attains the best result for a given metric.

## D.4 Computational Efficiency

We have conducted an empirical study to evaluate the computational efficiency of recent time series foundation models, which we present in Table [18](#). All experiments were performed on a single NVIDIA A100 (40 GB) GPU using synthetic multivariate time series. Each model was profiled under identical settings, with a context length of 2,048 and a prediction horizon of 480 for variable number of variates. We include comparisons against both multivariate (Moirai-Base) and univariate models (Time-MoE-50M, Chronos-Bolt-Base, and TimesFM-500M).

Table 18: Model Benchmark Results

# Variates	Model	Wall Time (ms)	CUDA Time (ms)	Peak Memory (MB)	Total FLOPs (GFLOPs)
10	TOTO	652.8 $\pm$ 12.6	37.0 $\pm$ 0.0	721.0 $\pm$ 0.0	<b>121.6 <math>\pm</math> 0.0</b>
	TOTO (no KV cache)	585.0 $\pm$ 7.9	45.8 $\pm$ 4.1	733.6 $\pm$ 0.1	885.2 $\pm$ 0.0
	Moirai	<b>148.1 <math>\pm</math> 2.4</b>	<b>19.1 <math>\pm</math> 1.7</b>	<b>489.8 <math>\pm</math> 0.0</b>	167.6 $\pm$ 0.0
	Time-MoE	1569.0 $\pm$ 150.1	361.2 $\pm$ 4.1	5341.2 $\pm$ 0.0	3449.9 $\pm$ 0.0
	Chronos	831.6 $\pm$ 25.2	93.7 $\pm$ 0.0	925.5 $\pm$ 0.0	2162.5 $\pm$ 0.0
	TimesFM	579.5 $\pm$ 44.9	109.6 $\pm$ 0.1	2023.6 $\pm$ 0.0	5188.1 $\pm$ 0.0
30	TOTO	627.2 $\pm$ 10.4	43.8 $\pm$ 0.4	934.9 $\pm$ 0.0	<b>364.7 <math>\pm</math> 0.0</b>
	TOTO (no KV cache)	628.2 $\pm$ 55.5	75.6 $\pm$ 5.5	971.9 $\pm$ 0.3	2655.5 $\pm$ 0.0
	Moirai	<b>284.1 <math>\pm</math> 204.7</b>	<b>86.4 <math>\pm</math> 4.3</b>	<b>1221.6 <math>\pm</math> 0.0</b>	642.1 $\pm$ 0.0
	Time-MoE	1924.7 $\pm$ 22.6	970.4 $\pm$ 5.5	15036.3 $\pm$ 0.0	10349.8 $\pm$ 0.0
	Chronos	925.8 $\pm$ 22.1	179.7 $\pm$ 8.4	1135.4 $\pm$ 0.0	6487.5 $\pm$ 0.0
	TimesFM	698.6 $\pm$ 18.1	280.3 $\pm$ 8.6	2200.9 $\pm$ 0.0	15564.3 $\pm$ 0.0
50	TOTO	687.5 $\pm$ 72.6	<b>50.7 <math>\pm</math> 2.9</b>	1148.3 $\pm$ 0.0	<b>607.8 <math>\pm</math> 0.0</b>
	TOTO (no KV cache)	627.1 $\pm$ 55.4	99.9 $\pm$ 7.7	1209.3 $\pm$ 0.0	4425.8 $\pm$ 0.0
	Moirai	<b>303.1 <math>\pm</math> 23.6</b>	186.4 $\pm$ 1.5	2642.0 $\pm$ 0.0	1302.2 $\pm$ 0.0
	Time-MoE	3239.9 $\pm$ 252.8	1572.0 $\pm$ 0.7	24730.3 $\pm$ 0.0	17249.6 $\pm$ 0.0
	Chronos	1029.6 $\pm$ 14.8	266.5 $\pm$ 18.2	1335.4 $\pm$ 0.5	10812.5 $\pm$ 0.0
	TimesFM	662.0 $\pm$ 9.2	441.4 $\pm$ 11.6	<b>2383.2 <math>\pm</math> 0.0</b>	25940.5 $\pm$ 0.0
70	TOTO	<b>667.1 <math>\pm</math> 12.4</b>	<b>57.6 <math>\pm</math> 4.5</b>	1361.6 $\pm$ 0.0	<b>850.9 <math>\pm</math> 0.0</b>
	TOTO (no KV cache)	669.8 $\pm$ 55.9	126.5 $\pm$ 2.6	1447.4 $\pm$ 0.0	6196.1 $\pm$ 0.0
	Moirai	454.8 $\pm$ 16.1	346.0 $\pm$ 2.9	4763.1 $\pm$ 0.0	2148.0 $\pm$ 0.0
	Time-MoE	8612.0 $\pm$ 248.8	3464.1 $\pm$ 168.3	30394.6 $\pm$ 423.5	37882.8 $\pm$ 1968.5
	Chronos	1193.4 $\pm$ 44.0	333.4 $\pm$ 10.7	<b>1543.3 <math>\pm</math> 0.0</b>	15137.6 $\pm$ 0.0
	TimesFM	842.0 $\pm$ 15.2	606.6 $\pm$ 7.7	2555.2 $\pm$ 0.0	36316.6 $\pm$ 0.0
90	TOTO	<b>712.1 <math>\pm</math> 48.6</b>	<b>66.9 <math>\pm</math> 0.0</b>	1575.9 $\pm$ 0.0	<b>1094.0 <math>\pm</math> 0.0</b>
	TOTO (no KV cache)	714.8 $\pm$ 91.3	156.8 $\pm$ 4.5	1685.5 $\pm$ 0.0	7966.5 $\pm$ 0.0
	Moirai	667.2 $\pm$ 5.5	562.1 $\pm$ 1.0	7577.0 $\pm$ 0.0	3179.4 $\pm$ 0.0
	Time-MoE	8985.5 $\pm$ 729.0	3315.5 $\pm$ 157.5	36581.7 $\pm$ 2532.6	36010.0 $\pm$ 1572.7
	Chronos	1209.9 $\pm$ 69.8	427.9 $\pm$ 7.8	<b>1750.2 <math>\pm</math> 0.0</b>	19462.6 $\pm$ 0.0
	TimesFM	1124.7 $\pm$ 36.1	786.3 $\pm$ 8.5	2727.2 $\pm$ 0.0	46692.8 $\pm$ 0.0
150	TOTO	<b>708.3 <math>\pm</math> 8.8</b>	<b>89.8 <math>\pm</math> 0.8</b>	<b>2223.2 <math>\pm</math> 0.0</b>	<b>1823.3 <math>\pm</math> 0.0</b>
	TOTO (no KV cache)	714.6 $\pm$ 31.5	257.5 $\pm$ 5.2	2406.4 $\pm$ 0.0	13277.4 $\pm$ 0.0
	Moirai	1830.3 $\pm$ 122.1	1488.5 $\pm$ 0.7	20193.6 $\pm$ 0.0	7387.2 $\pm$ 0.0
	Time-MoE	11903.6 $\pm$ 677.6	5109.4 $\pm$ 271.8	33639.9 $\pm$ 2419.7	56227.9 $\pm$ 3128.8
	Chronos	1435.2 $\pm$ 25.1	670.4 $\pm$ 4.3	2375.2 $\pm$ 0.0	32437.6 $\pm$ 0.0
	TimesFM	1699.8 $\pm$ 7.7	1239.3 $\pm$ 5.1	3256.4 $\pm$ 0.0	77821.4 $\pm$ 0.0
200	TOTO	<b>710.2 <math>\pm</math> 16.1</b>	<b>100.9 <math>\pm</math> 4.7</b>	<b>2757.5 <math>\pm</math> 0.0</b>	<b>2431.1 <math>\pm</math> 0.0</b>
	TOTO (no KV cache)	747.7 $\pm$ 71.8	340.7 $\pm$ 6.1	2998.2 $\pm$ 0.0	17703.2 $\pm$ 0.0
	Moirai	3097.1 $\pm$ 380.8	2316.1 $\pm$ 1.3	35487.6 $\pm$ 0.0	12170.0 $\pm$ 0.0
	Time-MoE	15668.0 $\pm$ 152.2	7828.7 $\pm$ 3.0	30352.4 $\pm$ 0.0	86490.8 $\pm$ 0.0
	Chronos	1799.4 $\pm$ 23.9	890.3 $\pm$ 7.0	2897.4 $\pm$ 0.0	43250.2 $\pm$ 0.0
	TimesFM	2284.4 $\pm$ 24.1	1665.3 $\pm$ 3.7	3698.9 $\pm$ 0.0	103761.8 $\pm$ 0.0
300	TOTO	<b>775.2 <math>\pm</math> 15.4</b>	<b>132.8 <math>\pm</math> 3.6</b>	<b>3821.7 <math>\pm</math> 0.0</b>	<b>3646.7 <math>\pm</math> 0.0</b>
	TOTO (no KV cache)	805.1 $\pm$ 25.3	505.8 $\pm$ 5.3	4187.8 $\pm$ 0.0	26554.9 $\pm$ 0.0
	Moirai	OOM	OOM	OOM	OOM
	Time-MoE	16590.2 $\pm$ 254.8	9698.9 $\pm$ 0.2	31517.0 $\pm$ 0.0	108459.7 $\pm$ 0.0
	Chronos	2583.3 $\pm$ 23.6	1290.2 $\pm$ 4.5	3935.1 $\pm$ 0.0	64875.2 $\pm$ 0.0
	TimesFM	3302.4 $\pm$ 22.0	2445.4 $\pm$ 5.9	4569.9 $\pm$ 0.0	155642.8 $\pm$ 0.0

To isolate the fundamental computational cost of each architecture, we evaluate simple forward passes generating a single output sample. This configuration reflects the core operation dominating training and fine-tuning, where repeated forward evaluations occur at scale, and thus provides a standardized measurement of training efficiency. Many models, including TOTO, increase the number of samples at inference time to boost predictive accuracy; this analysis does not address such test-time scaling.

TOTO demonstrates consistently strong computational efficiency and scalability compared to other models, with the lowest FLOPs across all variate counts. It achieves the lowest or second-lowest wall times across all variate counts, maintaining stable performance even as the input dimensionality increases from 10 to 300 variates. Starting from 150 variates onward, TOTO outperforms Moirai on all metrics, including wall time and CUDA time; the gap grows significantly with more variates. Even against univariate competitors, where cross-variate

interaction is ignored, TOTO provides better performance, demonstrating its highly optimized architecture and efficient memory use. We note that TOTO uses KV caching, while other models do not appear to implement this. For additional transparency, we also share TOTO statistics with KV caching turned off and note that the general trend remains the same.

## E Ablations

We evaluate the contribution of various architectural components of the TOTO model by systematically disabling one component at a time and measuring the relative performance degradation. The full TOTO model serves as the control, and each variant’s performance is presented relative to this baseline in Table 19. All models in the ablation study, including the control, were trained for 75,000 steps (a subset of the full-length training of the TOTO base model).

Model	Best NLL Loss (% increase) ↓
<b>Control</b>	<b>0.0%</b>
No Variate-wise Attention	1.6%
No Robust Loss	11.1%
No Student-T Mixture	27.2%
No Causal Scaling	27.3%

Table 19: Relative change in NLL on held-out observability pretraining data when removing key design features of the TOTO architecture.

To compare performance between the different arms of the experiment, we look at NLL loss on a held-out validation split of the observability portion of the pretraining data. This summarizes the output distribution and gives us a single performance metric to compare both point forecasting and probabilistic forecasting. For each model, we pick the checkpoint with lowest NLL throughout the training run (evaluating on the validation set every 5,000 steps).

The results reveal that removing key modeling elements significantly impacts performance. Disabling Causal Scaling leads to the largest degradation, with an increase of 27.3% in NLL when we replace the causal scaler with a naive global scaler. Replacing the Student-T mixture model with a single Student-T output causes a similar NLL increase of 27.2%. Interestingly, removing the robust loss component and optimizing NLL alone actually leads to a *worse* overall NLL, with an 11.1% increase; we speculate this is because the robust loss stabilizes the training, as discussed in Section 3.1. Finally, removing the variate-wise attention (i.e. making all the attention layers time-wise while holding the parameter count constant) leads to a more modest increase in NLL of 1.6%.

## F Impact statement

In developing TOTO, we followed a structured approach to ensure responsible development, focusing on identifying, assessing, and mitigating potential risks associated with the use of our model. Given that TOTO specifically generates time series forecasts, the potential harms are considerably lower compared to language, image, or other more general-purpose models. Our primary focus was ensuring the accuracy and reliability of the forecasts generated by TOTO, which are crucial for maintaining and optimizing infrastructure and application performance.

The BOOM benchmark provides numerical time series generated from observability metrics, which we view as a valuable resource to the broader time series research community. Each series has an associate high-level application label and no other metadata and contains no PII.