Towards Efficient Foundation Model: A Novel Time Series Embedding

Jessy Xinyi Han
Massachusetts Institute of Technology
xyhan@mit.edu

Arth Dharaskar
Ikigai Labs
adharask@ikigailabs.io

Nathaniel Lanier
Ikigai Labs
nathaniel@ikigailabs.io

Abdullah Omar Alomar
Ikigai Labs
abdullah@ikigailabs.io

Aditya Agrawal
Ikigai Labs
aagrawal@ikigailabs.io

Angela Yuan
Massachusetts Institute of Technology
yangela@mit.edu

Jocelyn Hsieh
Ikigai Labs
jhsieh@ikigailabs.io

Ishan Shah
Ikigai Labs
ishah@ikigailabs.io

Muhammad Jehangir Amjad
Ikigai Labs
jehangir@ikigailabs.io

Devavrat Shah
Massachusetts Institute of Technology
devavrat@mit.edu

Abstract

Time Series Foundation Model (TSFM) learns appropriate embeddings from pre-training data and uses them to embed the input time series for in-context learning to produce forecast. TSFM requires rich pre-training dataset and large computational resources to learn effective embeddings. In contrast, traditional time series modeling paradigm generates forecast for a given time series by fitting one of many pre-determined models and using the best of them to produce a forecast. Though resource efficient, it suffers from inability to utilize the pre-training data along with the challenges involved in the best model selection. In this work, we are motivated to bring the best of both worlds together to enable resource efficient TSFM approach. Towards that, we introduce a novel embedding of time series of any length and scale by mapping them to unit square (i.e $[0, 1]^2$) or equivalently a 2D image. To evaluate its efficacy compared to embedding from a TSFM, we consider the task of model identification or classification for dataset where each time series is generated from one of many pre-determined model class. We find that the performance of the proposed embeddings is comparable to that of embeddings from a pre-trained TSFM, but at a fraction of resource requirement. This suggests an alternative architectural possibility for a compute efficient TSFM.

NeurIPS 2025 Workshop on MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.

1 Introduction

Time Series Foundation Model (TSFM). In the simplest terms, a TSFM is a universal forecaster: when time series data of any length, with potentially missing values, is provided as an input, it can generate prediction or forecast for the future time step(s). Loosely speaking, a TSFM achieves this task by pre-training on a massive data repository and in the process, learning (encoding) and extracting (decoding) patterns from the dataset. Operationally, learning and extracting pattern corresponds to TSFM's ability to embed any time series into appropriate embedding space. Indeed, in an encoder-decoder model, for in-context learning, TSFM encoder embeds the input time series which in turn is decoded to produce forecast. In that sense, it is this time series embedding (encoder) of the TSFM that allows it to be an effective predictor or forecaster. Examples of TSFM include Timesfm [2], Moirai [4] and Chronos [1].

A Traditional Time Series Model. In the traditional paradigm, given an input time series, an appropriate model class with associated parameter(s) is chosen from a collection of them. This selection is typically done through an experimental framework where multiple models are trained on the same time series and the best model is chosen based on their out-of-sample performance. Such approaches are typically resource efficient, especially compared to TSFM. Examples of such an approach includes Prophet [3].

Key Challenges. The TSFM comes with the challenge of (a) resource intensive training, inference, (b) requirement of *sufficiently rich* pre-training dataset. The traditional approaches come with the challenge of (c) experimentation being ineffective especially for short time series, (d) unable to incorporate information available in the pre-training data.

Goal. Is it possible to overcome challenges (a)-(d) simultaneously? Specifically, it is possible to have a resource efficiency of traditional approach while ability to incorporate information from pre-training data for forecasting a given time series and avoid the need of experimentation.

1.1 Contributions. As the main contribution of this work, we introduce novel, universal time series embedding. For any given time series of any length and scale along with potentially missing values, it maps the time series to a unit square, viewed equivalently as mapping to a 2D image of fixed size. A typical TSFM uses embeddings along with pre-learned model transformation in the training phase to produce a forecast. In the traditional approach, the goal is to learn which is the best model class amongst a collection of them. In effect, both approaches try to utilize the input time series to determine what sorts of pattern it corresponds to. Therefore, to evaluate efficacy of the embedding proposed in this work, we consider the following model identification problem: given a collection of time series generated from one of the many model classes, identify the true generating model class based on the obesrved time series only. For this model identification or classification problem, we compare the performance of our proposed time series embedding and that of TimesFM [2]. Specifically, Table 1 provides a summary of our findings. As can be deduced from it, we find that the performance of our embedding is nearly identical to that of embeddings of the TSFM; and our embedding takes a fraction of computational resource compared to that of TSFM.

1.2 Organization. The rest of the manuscript is organized as follows. In Section 2, we introduce the novel time series embeddings. In Section 3, we introduce the model classification task and evaluate performance of the introduced embeddings. We compare its performance with that of embeddings from a pre-trained TSFM. We provide brief conclusion in Section 4.

2 Embeddings of Time Series

Let $x_{1:T} = x_1, \ldots, x_T \in \mathbb{R} \cup \{\star\}$ denote a time series of length T; $\star =$ missing observation.

2.1 Image-Based Embeddings of Time Series

We describe three embeddings that transform $x_{1:T}$ into structured image-like representations along the time axis. Each method partitions the time horizon into K intervals and optionally discretizes values into J bins. We first normalize the series to $[0,1]$: $y_t = \begin{cases} \star, & x_t = \star, \\ \frac{x_t - x_{\min}}{x_{\max} - x_{\min}}, & \text{otherwise,} \end{cases}$ with $x_{\min} = \min_{t:x_t \neq \star} x_t$ and $x_{\max} = \max_{t:x_t \neq \star} x_t$, and $0/0 = 0$.

Table 1: Performances of Different Embeddings. p is the probability that a time point is NaN. We report embedding dimension (Dim), CV accuracy/F1, strict Test accuracy (Overall/Pure/Mixed), and Test Relaxed Overlap accuracy (Overall/Pure/Mixed). Best performing is **bolded**.

	Dim	CV Acc / F1	Test Strict Acc Overall / Pure / Mixed	Test Overlap Acc Overall / Pure / Mixed
Missingness $p = 0.0$				
RGBA (K=8)	32	.745 / .741	–	–
BW (J=40, K=64)	1280	.755 / .749	–	–
Heat (J=40, K=32, blur=1)	1280	**.766 / .761**	.775 / .807 / **.754**	.988 / .979 / .994
TimesFM	1280	–	**.781 / .843** / .740	**.997 / .995 / .998**
Missingness $p = 0.1$				
RGBA (K=8)	32	.719 / .712	–	–
BW (J=40, K=32)	1280	.746 / .740	–	–
Heat (J=40, K=32, blur=1)	1280	**.749 / .743**	**.759** / .811 / **.725**	.983 / .969 / .993
TimesFM	1280	–	.748 / **.824** / .697	**.996 / .991 / .998**
Missingness $p = 0.2$				
RGBA (K=8)	32	.707 / .698	–	–
BW (J=40, K=32)	1280	.735 / .727	–	–
Heat (J=40, K=32, blur=1)	1280	**.739 / .732**	**.738** / .800 / **.696**	.983 / .968 / .993
TimesFM	1280	–	.721 / **.805** / .665	**.992 / .989 / .994**

A. Black-White Grid. This embedding aims to capture *whether* certain value ranges are ever visited by the time series. Let $b_t = \lfloor (J-1)\, y_t \rceil$ be the value bin index for $y_t \neq \star$, where y_t is the normalized series on $[0, 1]$ and $\lfloor \cdot \rceil$ denotes rounding to the nearest integer. We form a binary matrix $\mathcal{B} \in \{0, 1\}^{J \times K}$: $B_{j,k} = \mathbf{1}\Big\{\exists\, t \in \mathcal{I}_k : y_t \neq \star \wedge b_t = j\Big\}$ where $\mathcal{I}_k$ denotes the k-th interval of time. Thus, $B_{j,k} = 1$ if bin j is hit during interval k, and 0 otherwise. This visualizes which amplitudes appear in each time slice and yields a black-and-white bitmap of support of the time series.

B. Density Heatmap. To capture not just whether values occur but also their *relative frequency*, we construct a density heatmap. This embedding retains information about concentration and shape of the distribution over time. Using the same binning as Black-White Grid, we count the number of hits in each bin: $H_{j,k} = \big|\{\, t \in \mathcal{I}_k : y_t \neq \star \wedge b_t = j \,\}\big|, \quad \mathcal{H} \in \mathbb{R}_{\geq 0}^{J \times K}$. The result $\mathcal{H}$ is a vertical density heatmap of size $J \times K$. Optionally, a Gaussian blur can be applied for smoothing.

C. Segmented Quantile RGBA Embedding. In settings where robustness to outliers is beneficial, quantiles often provide a compact and interpretable summary. For each interval $\mathcal{I}_k$, we compute: (1) Missingness fraction $m_k = \frac{1}{|\mathcal{I}_k|} \sum_{t \in \mathcal{I}_k} \mathbf{1}\{y_t = \star\}$; (2) Quantiles $Q_k^{0.25}, Q_k^{0.50}, Q_k^{0.75}$ of the non-missing values.

We then encode into a pixel $(R_k, G_k, B_k, A_k) \in [0, 255]^4$:

$$R_k = \lfloor 255 \cdot Q_k^{0.25} \rfloor, \quad G_k = \lfloor 255 \cdot Q_k^{0.50} \rfloor, \quad B_k = \lfloor 255 \cdot Q_k^{0.75} \rfloor, \quad A_k = \lfloor 255 \cdot (1 - m_k) \rfloor.$$

Stacking across $k = 1, \ldots, K$ yields an array in $[0, 255]^{K \times 4}$. This embedding forms a colored strip encoding distributional shape and data presence per segment.

We have included the visualizations for these 3 different embeddings mentioned above in Fig. 1.

2.2 TimesFM Emebedding

In contrast to the image-based encodings above, we also evaluate a learned representation derived from a TSFM TimesFM [2]. We instantiate the public `google/timesfm-2.0-500m-pytorch` checkpoint, configured with 50 Transformer layers, and a context length of 2048. Given an input series $x_{1:T}$, the encoder produces a sequence of patch embeddings after which mean pooling is applied across the patch dimension to obtain a single fixed–dimensional embedding vector for each time series. The resulting embedding captures long–range temporal patterns and higher–order dependencies learned during pretraining. In case there are NaN values, we first fill in the missing values with linear interpolation before using the model to get embeddings.

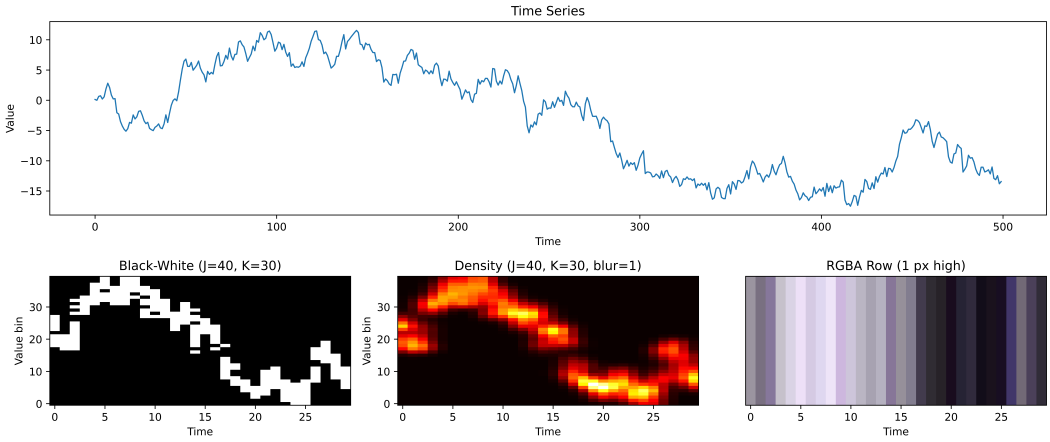

Figure 1: Visualizations of a time series and its corresponding three image-based embeddings.

3 Experiments, Evaluation

Setup. We construct a collection of synthetic time series from several classical models (SARIMA, Linear, Bernoulli, Harmonic). To test the performance on compositional time series, we also generate mixtures of two "pure" classes by combining them with random weights. The dataset is stratified into 80/20 train–test splits across pure and mixed classes. We compare three proposed image-based encodings (binary grid, heatmap, quantile RGBA) against the TimesFM embeddings. For each image-based embedding, hyperparameters (e.g., temporal resolution, discretization, blur) are tuned by cross-validation. To evaluate robustness, we further inject random missingness ($p = 0.1, 0.2$). All embeddings are processed via a unified pipeline: standardization, PCA, and a random forest classifier. On the test set, we report strict accuracy, macro-F1, and confusion matrices. Due to the inclusion of mixture classes, we additionally compute a relaxed "overlap accuracy," counting predictions correct if at least one constituent class is recovered. Results are reported overall and separately for pure vs. mixed classes. Full details of the experimental setup are deferred to the Appendix A.

Results. Across all settings, the heatmap embedding with Gaussian blur (J=40, K=32, blur=1) and TimesFM consistently emerged as the strongest performers. As shown in Table 1, both methods maintain high discriminative power under increasing missingness. At $p = 0$, TimesFM achieves the best strict accuracy (.781 overall, .843 for pure classes), slightly outperforming the heatmap embedding (.775 overall). However, as missingness increases, the heatmap embedding achieves better strict accuracy, with TimesFM slightly lower but higher on pure classes. TimesFM also retains superior overlap accuracy across different levels of missingness. Comparing accuracies of pure and mixed classes, pure classes remain stable ($\simeq .80$), while mixed classes drop more sharply (from .754 to .696 for heatmaps; .740 to .665 for TimesFM).

4 Conclusion

We introduce a lightweight image-based embedding alternative to pre-trained foundation model embeddings. Through a synthetic model identification benchmark, we show that these image-based embeddings achieve performance comparable to TimesFM while requiring only a fraction of the computational resources. Our empirical analysis demonstrates complementary strengths that heatmap embeddings with Gaussian blur excel in strict classification under moderate missingness, whereas TimesFM provides slightly better overlap accuracy, particularly for mixed classes. Both approaches demonstrate robustness to data corruption up to 20%.

These findings suggest that simple, resource-efficient encoders can rival foundation models on structured tasks, and may serve as building blocks for compute-efficient TSFMs. Future work includes extending the evaluation to real-world datasets, exploring hybrid encoders that combine image-based features with learned representations, and integrating the proposed image-based embeddings with pre-trained vision transformers.

References

[1] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024.

[2] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting, 2024.

[3] Sean J. Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

[4] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers, 2024.

A Details of Experimental Evaluations

Dataset. We begin with a library of simulated time series generated from several types of classical time series models with randomly sampled parameters, including SARIMA, Linear, Bernoulli, and Harmonic. These serve as "pure" classes in our synthetic data. To test whether representations can capture compositional structure, we also created "mixed" classes by combining two pure series. Given standardized sequences a and b, we draw $r \sim \mathcal{U}(0.3, 0.7)$ and form the mixed sequence

$$y = \sqrt{r}\, a + \sqrt{1-r}\, b.$$

which preserves unit variance of the mixture by allocating "energy" in proportions r and $1-r$, in contrast to a convex combination $ra + (1-r)b$ that would systematically reduce variance. Repeating this procedure produces 1000 mixtures per unordered class pair. The final corpus thus contains both pure and mixed examples, labeled accordingly.

Train–test. The full dataset is split 80/20 into training and test partitions, stratified by label. This ensures both pure and mixed categories are proportionally represented.

Encoding. Each series is transformed into one of the three image-based embeddings defined in Section 2: the binary black–white grid (E1), density heatmap (E2), segmented quantile RGBA strip (E3), or the foundation model TimesFM embeddings (E4). For each embedding we sweep natural hyperparameters such as temporal resolution K, value discretization J, and blur strength. Flattened feature vectors are padded to a common length when necessary.

To assess robustness under incomplete data, we inject missing values at random (MCAR) with probabilities $p \in \{0.1, 0.2\}$, replacing each time point independently with $\star$. The encoders were NaN-safe: RGBA explicitly encodes missingness in its alpha channel, while BW and Heatmap embeddings ignore NaNs during binning. We repeated the same cross-validation and held-out test evaluation protocol as in the complete-data setting.

Classification. All embeddings are processed through a unified supervised pipeline:

1. *Standardization.* Each feature dimension is centered and rescaled to unit variance.
2. *Dimensionality reduction.* Principal component analysis (PCA) is applied, retaining at most 128 components if original embedding is of a higher dimension.
3. *Classifier.* A random forest classifer with 200 decision trees is trained.

Model selection. For the image-based encodings, we evaluate candidate encoders and their hyperparameters using five-fold stratified cross-validation on the training set, scoring by accuracy and macro-F1 (averaged harmonic mean of precision and recall). The configuration with the best macro-F1 is retrained on the full training data. We also run the TimesFM embedding for comparison against the best performing image-based embedding.

Evaluation metrics. On the held-out test set, we report (i) strict classification accuracy, (ii) F1, and (iii) confusion matrices. Because mixed classes consist of two base labels, we additionally report a relaxed "overlap accuracy" in which a prediction is considered correct if it recovers at least one constituent class. Results are also broken out by pure and mixed subsets.