FiTEM: Fine-tuning Time-series Foundation Models for Selective Forecasting

Jonas Brusokas[1] (✉), Seshu Tirupathi[2], and Torben Bach Pedersen[1]

[1] Department of Computer Science, Aalborg University {jonasb,tbp}@cs.aau.dk
[2] IBM Research, Ireland seshutir@ie.ibm.com

Abstract. Time-series forecasting is critical for many real-world applications. Recent advances in time-series foundation models have substantially improved forecasting performance across diverse domains. However, most time-series foundation models remain deterministic and produce only point estimates without model confidence quantification, risking costly errors when faced with previously unseen data distributions. So called, *selective* forecasting mitigates this risk by enabling forecasting models to abstain from low-confidence predictions, trading coverage for improved reliability. In this paper, we introduce the Fine-tunable Time-Energy Model for time-series foundation models (FiTEM), a selective forecasting framework that extends pre-trained time-series foundation models, enabling selective forecasting. FiTEM appends a lightweight decoder to a pre-trained time-series foundation model and trains it via self-supervised learning during few-shot fine-tuning to produce confidence scores for each forecast and use those to reject low-confidence forecasts. FiTEM builds on state-of-the-art selective forecasting techniques requiring only a small amount of labeled target data and is trained as part of few-shot fine-tuning of a pre-trained time-series foundation model. We evaluate FiTEM on several time-series forecasting benchmark datasets unused during base model training in two modes: zero-shot, where FiTEM components are trained on limited target data without updating the parameters of the pre-trained foundation model, and few-shot, where the pre-trained model is few-shot fine-tuned on a small fraction of target data before FiTEM components are trained on the same data. Experiments show that FiTEM reduces forecasting error by up to 56.4% at low target coverage and up to 35.4% for target coverage of 50% and above.

Keywords: Time-series foundation models · Selective forecasting · Fine-tuning · Confidence estimation.

1 Introduction

Time-series forecasting is used for many real-world applications, such as energy management, weather forecasting, and traffic prediction [17]. Significant progress has been made in time-series forecasting, with specialized deep learning models improving state-of-the-art performance [17, 15, 18, 14, 12]. Traditionally, time-

series forecasting models are trained and evaluated on a single labeled dataset, which limits their ability to generalize to new domains or datasets [8].

Recently, time-series foundation models trained on a wide range of time-series datasets have emerged, offering a new paradigm for time-series forecasting and analytics [9, 8, 2, 3, 11, 10, 19, 7, 5]. Unlike traditional time-series models trained on a single labeled dataset, foundation models are trained on a wide range of time-series datasets across multiple domains, and hence generalize across diverse distributions, delivering strong zero-shot and few-shot forecasting performance. These models include large language models adapted for time-series (LM-based) [19, 7, 5] and models specifically pre-trained on time-series data (TS-based) [11, 10, 3]. TS-based models are generally smaller with fewer parameters, resulting in less memory usage and faster inference, which is crucial for many real-time applications [8]. Moreover, studies have shown that LM-based time-series models often underperform compared to TS-based models in zero-shot and few-shot settings [8].

A key limitation of current TS-based models is that most of them are deterministic, producing only best-guess point estimates without quantification of uncertainty or model confidence [11, 3, 10]. This is particularly problematic in zero-shot forecasting scenarios, where models are used to forecast unseen data without any fine-tuning. In such cases, the model's performance can vary significantly depending on whether the patterns in the target data align with those observed during pre-training, leading to potentially unreliable forecasts without any indication of confidence. Therefore, having only point forecasts reduces overall model utility, especially in applications where the penalty for erroneous forecasts is high and managing risk becomes essential [1, 16]. So-called *selective forecasting* addresses this limitation by extending deterministic forecasting models with a selection function that allows models to abstain from making predictions when model confidence is low [1, 4]. Selective forecasting enables a trade-off between *selective coverage* (the percentage of forecasts that are selected) and *selective risk* (error of the selected forecasts only), ensuring that the model only forecasts when sufficiently confident. A selective forecasting framework called The Time-Energy Model (TEM) [1] has been proposed to enable selective forecasting for traditional time-series forecasting models, trained on a single labeled dataset. However, there are no selective forecasting methods for large-scale TS-based models, trained on a large corpus of time-series.

In this paper, we introduce Fine-tunable Time-Energy Model (FiTEM), a selective forecasting framework that extends pre-trained time-series foundation models, enabling selective forecasting. FiTEM appends a lightweight decoder to a pre-trained TS-based model and trains it via self-supervised Contrastive Divergence (CD) learning to produce model confidence scores and enable selective forecasting. FiTEM extends state-of-the-art selective forecasting techniques to TS-based models, requiring only a small amount of labeled target data and is trained as part of few-shot fine-tuning of a pre-trained time-series foundation model. Experiments using two TS-based models (Timer and TimerXL) on 3

benchmark datasets show that FiTEM reduces forecasting error by up to 56.4% and up to 35.4% for target coverages of 10% and 50%, respectively.

The paper is organized as follows: Section 2 reviews related work on time-series foundation models and selective forecasting. Section 3 formally defines the forecasting and selective forecasting problems for time-series foundation models. Section 4 presents the proposed FiTEM method. Section 5 describes the experimental evaluation setup. Section 6 reports the experimental results. Finally, Section 7 provides concluding remarks and future directions.

2 Related Work

Deep-learning based time-series forecasting models have advanced significantly in recent years. Recent models like Informer [17], Autoformer [15], FEDformer [18], TimesNet [14], and PatchTST [12] have improved state-of-the-art performance by increasing model accuracy, reducing latency, and enabling multi-task capabilities including short and long-term forecasting, anomaly detection, classification, and imputation. These traditional approaches, often referred to as *time-series specific models* [8], are typically trained and evaluated on a single labeled dataset. While they achieve excellent performance on their specific training datasets, they often cannot generalize to other datasets or distributions. Furthermore, these models generally require substantial amounts of labeled data to train, which is often not available in practice.

Time-series foundation models present an alternative paradigm to time-series specific models. These models are trained on large corpora of datasets across diverse domains, enabling more accurate forecasts across data distributions unseen during training. Foundation models can be categorized into two main types: language-based (LM-based) and time-series-based (TS-based) models. LM-based time-series foundation models leverage large language models pre-trained on extensive text corpora and adapt them for time-series analytics tasks, utilizing their general language understanding and processing capabilities [19, 7, 5]. However, empirical studies indicate that LM-based models often fail to provide meaningfully accurate forecasts, particularly when considering their significant computational requirements [13]. TS-based models are pre-trained exclusively on time-series data and designed for various time-series analytics tasks. These models are trained on a wide range of time-series datasets and are generally smaller and less computationally intensive than LM-based alternatives. Despite their relative size, TS-based models still deliver excellent zero-shot and few-shot performance [10, 11, 3]. The reduced computational footprint makes TS-based models more practical for real-time applications while maintaining strong generalization capabilities across different time-series domains.

One key limitation of TS-based models is that they are typically deterministic, meaning they do not quantify model confidence, which is critical for informed decision-making in real-world applications [1]. Recently, the TEM framework [1] has been proposed to enable so-called *selective forecasting* for deterministic time-series forecasting models, trained on a single labeled dataset. TEM adds

additional neural network layers to encoder-decoder time-series specific models and trains them to produce confidence scores for each forecast. These confidence scores are then used to selectively reject low-confidence forecasts, improving forecasting accuracy. TEM has been shown to work on a wide-range of time-series benchmarks [1], but was developed for time-series specific models, such as Informer [17], Autoformer [15], FEDformer [18], and TimesNet [14], and is trained on a single labeled dataset [1]. At the time of writing, there are no known selective forecasting methods for TS-based models.

In this paper, we propose FiTEM, a fine-tuning-based selective forecasting method for TS-based models. Similarly to state-of-the-art selective forecasting methods, FiTEM introduces a lightweight decoder appended to a pre-trained time-series foundation model to generate confidence scores [1]. FiTEM can be trained in a few hours on a single GPU using only a small amount of labeled target data.

3 Preliminaries and Problem Definition

Deterministic time-series forecasting. Let $\mathbf{X} = (x_{t-m+1}, \ldots, x_t)$ denote the input time series of length m with d features, and $\mathbf{Y} = (y_{t+1}, \ldots, y_{t+h})$ denote the target series of forecasting horizon h. A deterministic time-series forecasting model f_ψ (with parameters θ) produces a point estimate $\hat{\mathbf{Y}} = f_\psi(\mathbf{X})$.

In the context of time-series foundation models, two distinct classes of datasets are considered: *source datasets* $\mathcal{D}_s = \{(\mathbf{X}_s, \mathbf{Y}_s)\}$ and *target dataset* $\mathcal{D}_t = \{(\mathbf{X}_t, \mathbf{Y}_t)\}$. The source datasets $\mathcal{D}_s$ are used for training and validating foundation models. Source datasets generally contain a large number of varied time-series, enabling the model to learn general patterns and dependencies. The target dataset $\mathcal{D}_t$ represents the application-domain data and is used to evaluate the trained model's f_ψ performance in *zero-shot*, *few-shot*, or *full-shot* forecasting scenarios, where a subset of the model parameters θ is fine-tuned on the target data. Source and target datasets do not contain overlapping time-series, i.e., $\mathcal{D}_s \cap \mathcal{D}_t = \emptyset$. In case of few-shot forecasting, a subset of the target dataset $\mathcal{D}'_t \subset \mathcal{D}_t$ is used.

Selective time-series forecasting. Selective time-series forecasting extends deterministic time-series forecasting models f_ψ by introducing a *selection function* g that maps each input $\mathbf{X}$ to $\{0, 1\}$, indicating whether the model should select or reject the forecast. The selective forecasting model (f_ψ, g) produces forecasts as follows:

$$(f_\psi, g)(\mathbf{X}) = \begin{cases} \hat{\mathbf{Y}} = f_\psi(\mathbf{X}), & \text{if } g(\mathbf{X}) = 1 \\ \perp, & \text{if } g(\mathbf{X}) = 0 \end{cases} \tag{1}$$

where the forecast $\hat{\mathbf{Y}} = f_\psi(\mathbf{X})$ is selected if $g(\mathbf{X}) = 1$, and rejected when $g(\mathbf{X}) = 0$. *Selective coverage* $\phi(g)$ quantifies the proportion of inputs selected by the selection function g, and can be viewed as the probability of a forecast being selected.

$$\phi(g) \triangleq \mathbb{E}[g(\mathbf{X})] \equiv \mathbb{P}(g(\mathbf{X}) = 1) \tag{2}$$

Selective risk $R(f_\psi, g, \ell)$ defines the forecast error of the predictive model f_ψ on selected forecasts using the selection function g, where ℓ is a pointwise loss function (such as mean squared error):

$$R(f_\psi, g, \ell) = \frac{\mathbb{E}\big[\ell\big((f_\psi, g)(\mathbf{X}), \mathbf{Y}\big) \cdot g(\mathbf{X})\big]}{\phi(g)} \tag{3}$$

Problem statement. Given a foundation model f_ψ trained on the source datasets $\mathcal{D}_s$, find a selective forecasting model (f_ψ, g) that improves zero-shot and few-shot forecasting accuracy on the target dataset $\mathcal{D}_t$ while maintaining selected target coverage $\phi(g)$.

4 Proposed Method Overview

In this section, we introduce FiTEM (Fine-tunable Time-Energy Model), a selective forecasting framework that extends pre-trained time-series foundation models and enables uncertainty quantification and selective forecasting capabilities. FiTEM addresses the key limitation of current TS-based time-series foundation models by providing a mechanism to reject low-confidence forecasts while maintaining high accuracy on selected forecasts.

FiTEM builds upon the Time-Energy Model (TEM) framework [1] by adapting it for TS-based foundation models, enabling few-shot selective forecasting on target data, previously unseen during training. FiTEM consists of three key components: (1) a lightweight FiTEM architecture that builds a lightweight decoder on top of the pre-trained foundation model, (2) a few-shot training procedure that trains the FiTEM decoder using limited target data, and (3) an inference method that defines the selection function g to enable selective forecasting.

4.1 Architecture

FiTEM extends pre-trained time-series foundation models f_ψ with a lightweight decoder to enable selective forecasting. As shown in Figure 1, the architecture consists of two main components: the pre-trained foundation model f_ψ and the FiTEM decoder E_θ.

The pre-trained foundation model f_ψ is trained on large-scale source datasets $\mathcal{D}_s$ and is used to generate deterministic point forecasts $\hat{\mathbf{Y}} = f_\psi(\mathbf{X})$ for input sequences $\mathbf{X}$. As shown in Figure 1, an arbitrary f_ψ model consists of two components: an encoder ψ_e, which is used to capture and identify patterns in the input data $\mathbf{X}$ and generate an intermediate latent representation $\mathbf{f}_X$, and a decoder ψ_d, which acts as a forecasting head and uses the intermediate latent representation $\mathbf{f}_X$ from the encoder to generate the forecast $\hat{\mathbf{Y}}$. The pre-trained model f_ψ encoder ψ_e can be optionally finetuned on a subset of the target dataset $\mathcal{D}'_t$ to improve the model's forecasting on the target dataset.

The FiTEM decoder E_θ is a lightweight Energy-based Model (EBM) based architecture from TEM [1] that re-uses parameters ψ from the foundation model

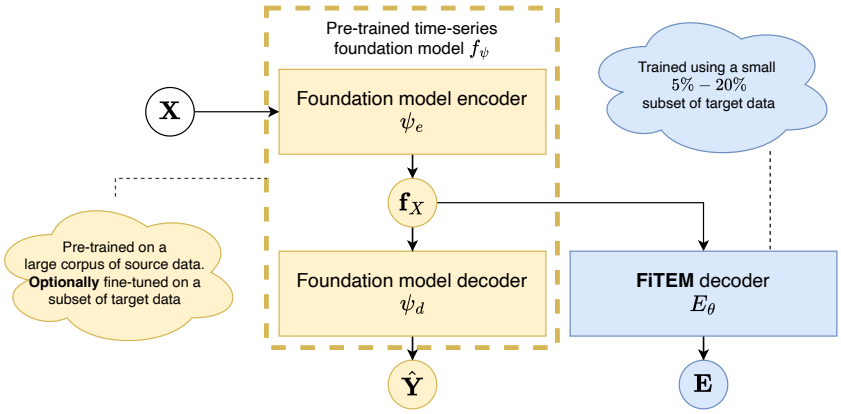

Fig. 1. FiTEM architecture for pre-trained TS-based foundation models

f_ψ. The decoder E_θ learns to re-use the input representations $\mathbf{f}_X$ from the foundation model's f_ψ encoder parameters ψ_e to predict the confidence score $\mathbf{E}$ (called Energy) of the forecast $\mathbf{Y}$. The decoder E_θ is trained using a subset of the target dataset $\mathcal{D}'_t$ (the same dataset as the one used for finetuning the foundation model) to enable selective forecasting on the target dataset. As the E_θ re-uses ψ from the foundation model to capture patterns in the input series $\mathbf{X}$, the E_θ can have significantly less parameters than the original f_ψ model. In FiTEM, we utilize a lightweight MLP-based architecture with a few layers to ensure fast model training and selective forecasting.

4.2 Training Method

FiTEM proposes to train the additional FiTEM decoder E_θ on a subset of the target dataset $\mathcal{D}'_t$ as an additional step to traditional few-shot fine-tuning to enable selective forecasting on the target dataset. Training the E_θ can utilize as few as 5% of the target dataset $\mathcal{D}'_t$, making it suitable for scenarios with limited labeled data.

Few-shot fine-tuning is generally performed by training the pre-trained foundation model f_ψ encoder parameters ψ_e on a limited target data $\mathcal{D}'_t$ using standard supervised learning with the original forecasting loss. This adapts the foundation model f_ψ to target domain characteristics while preserving general time-series patterns learned during pre-training. The fine-tuning uses a reduced learning rate and early stopping to prevent overfitting on the limited data.

In FiTEM, we propose to use the same target dataset $\mathcal{D}'_t$ to train the FiTEM decoder E_θ using self-supervised learning to enable selective forecasting on the target dataset, enabling reducing forecasting error on the target data. We use contrastive divergence (CD) [6] self-supervised learning to train the FiTEM de-

coder E_θ, as used in TEM [1]. The CD training process contrasts positive samples $\mathbf{Y}^{(0)}$ from the training set against negative samples $\mathbf{Y}^{(1)}$ generated through Langevin dynamics:

$$\mathbf{Y}^{(1)} = \mathbf{Y}^{(1)} - \eta \nabla_{\mathbf{Y}^{(1)}} E(\mathbf{X}, \mathbf{Y}^{(1)}) + \omega \tag{4}$$

$$\mathcal{L}_{CD} = (E^+ - E^-) + \lambda_{\text{reg}}((E^+)^2 + (E^-)^2) \tag{5}$$

where $E^+ = E(\mathbf{X}, \mathbf{Y}^{(0)})$ and $E^- = E(\mathbf{X}, \mathbf{Y}^{(1)})$ represent positive and negative sample energies, respectively.

4.3 Inference

Once the FiTEM decoder E_θ is trained, we can use it to perform selective forecasting on the target dataset $\mathcal{D}'_t$. We utilize the Aggregated Energy inference method as proposed in TEM [1] to generate model confidence scores for each forecast $\hat{\mathbf{Y}} = f_\psi(\mathbf{X})$ and then use it to reject low-confidence forecasts based on the user-defined target coverage ϕ.

Aggregated Energy Computation. For each forecast $\hat{\mathbf{Y}} = f_\psi(\mathbf{X})$, FiTEM computes an aggregated energy score that captures the local energy landscape around the prediction:

$$E_{\text{agg}}(\mathbf{X}, \hat{\mathbf{Y}}) = \frac{1}{n} \sum_{i=1}^{n} E(\mathbf{X}, \hat{\mathbf{Y}} + \boldsymbol{\epsilon}_i) - E(\mathbf{X}, \hat{\mathbf{Y}}) \tag{6}$$

where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ are noise samples that probe the energy surface around the forecast. This aggregated energy summarizes how compatible are the forecast $\hat{\mathbf{Y}}$ and the input series $\mathbf{X}$ by sampling energy scores on and around the forecast $\hat{\mathbf{Y}}$.

Coverage Calibration. To enable selective forecasting, FiTEM calibrates aggregated energy scores by partitioning the energy range into aggregated energy intervals and computing the empirical mean forecasting error for each interval. Energy intervals are ranked by forecast error, and cumulative coverage is calculated to determine selection thresholds for desired coverage levels.

Selective Forecasting using Calibrated Aggregated Energy intervals. During inference, FiTEM computes the aggregated energy for each forecast and compares it against pre-calibrated thresholds to make selection decisions:

$$(f_\psi, g)(\mathbf{X}) = \begin{cases} \hat{\mathbf{Y}} = f_\psi(\mathbf{X}), & \text{if } E_{\text{agg}}(\mathbf{X}, \hat{\mathbf{Y}}) \leq \tau(\phi) \\ \perp, & \text{otherwise} \end{cases} \tag{7}$$

where $\tau(\phi)$ is the energy threshold corresponding to desired coverage ϕ. This approach enables real-time selective forecasting with user-defined coverage levels, providing a practical solution for reliable deployment of foundation models in critical applications [1].

5 Experiment Setup

5.1 Foundation Models

To evaluate the performance of FiTEM, we use several state-of-the-art time-series foundation models f_ψ. These models are trained on a wide range of source datasets $\mathcal{D}_s$ from different domains. We use the following models:

- **Timer** [11] – a large pre-trained time series foundation model designed for univariate time-series analytics, including forecasting, imputation, and anomaly detection. Timer is trained on a large openly available corpus of time-series datasets from different domains, including electricity, traffic, weather, and financial time-series. Timer has shown consistently strong performance in both zero-shot and few-shot forecasting scenarios.
- **TimerXL** [10] – is a large decoder-only Transformer model builds on Timer by supporting significantly larger context windows on multivariate time-series. TimerXL has been shown to capture dependencies between correlated time-series and to outperform Timer in zero-shot and few-shot forecasting scenarios.

Both these models are among the best performing TS-based models in literature, have open-source implementations, and are widely used in the time-series community. Furthermore, both of these model's implementations share the same data splitting and loading strategies, as well as sampling strategies for few-shot forecasting, making them directly comparable (which is rarely the case, as shown in [8]).

5.2 Target Datasets

In this paper, we evaluate the performance of FiTEM on three distinct target datasets $\mathcal{D}_t$ from different domains. These datasets are widely used to benchmark state-of-the-art time-series forecasting models and are known to be challenging for traditional forecasting models [17, 15, 18, 14, 12]. Notably, the target datasets $\mathcal{D}_t$ are not used for training any of the foundation models f_ψ (i.e. they are not part of the source datasets $\mathcal{D}_s$ used for pre-training) [11, 10].

- **ETTh1, ETTh2** [17] – Electricity Transformer Temperature datasets containing 2 years of hourly temperature measurements from two electricity transformers in separate Chinese counties, each with 7 sensor features.
- **Exchange Rate** – Daily exchange rates between 8 different currencies against USD from 1990 to 2016, with XRP/USD as the target variable for forecasting.

We use the same sequence length $m = 96$ and prediction horizon $h = 48$ for all experiments across all datasets. All data preprocessing follows the same procedures as outlined in the original Timer and TimerXL papers [11, 10].

5.3 Model Evaluation

We evaluate the performance of the foundation models f_ψ in zero-shot and few-shot forecasting scenarios. In zero-shot forecasting, the foundation model f_ψ is evaluated on the target dataset $\mathcal{D}_t$ without any additional training. In few-shot forecasting, the foundation model f_ψ is finetuned on a subset of the target dataset $\mathcal{D}'_t$ and then evaluated on the target dataset $\mathcal{D}_t$.

In both cases, model performance is evaluated using the test set of the target dataset $\mathcal{D}_t$. The same test set is used, regardless of whether the model is finetuned or not, or how large the finetuning target dataset $\mathcal{D}'_t$ is. Mean Square Error (MSE) is used as the evaluation metric for both deterministic forecasting and as the distance metric for selective risk, following evaluation setup in [1].

5.4 Implementation details

Foundation models. For Timer and TimerXL, we use the official implementations with default hyperparameters as specified in [11, 10]. For zero-shot forecasting, we use the foundation models without any additional training on the target datasets $\mathcal{D}_t$. The models are only trained on their respective source datasets $\mathcal{D}_s$. For few-shot forecasting, we fine-tune the foundation models on subsets of the target datasets $\mathcal{D}'_t$ (5% and 20% of the total target dataset). Fine-tuning is performed for up to 10 epochs using Adam optimizer with a learning rate of 0.0001 and early stopping with a patience parameter of 3. Open source implementations of Timer and TimerXL are available at https://github.com/thuml/OpenLTM.

FiTEM architecture, training, and inference. The FiTEM decoder E_θ uses an MLP-based architecture with 3 fully connected layers and 128 hidden units in each layer. FiTEM is trained using the same contrastive divergence (CD) self-supervised learning procedure as used in the TEM experiments [1] on the same subset of target data $\mathcal{D}'_t$ used for fine-tuning. We use 5% and 20% subsets of the target datasets $\mathcal{D}'_t$ for training FiTEM. The same subsets are used for fine-tuning the foundation models in the few-shot setting. For inference, we use the same configuration of the inference procedure as used in the TEM experiments [1] to enable FiTEM selective forecasting.

Computational resources. All experiments were conducted on a single NVIDIA Quadro RTX 8000 GPU with 48GB VRAM.

6 Results

To quantify the performance of the proposed FiTEM method, we conducted experiments on 3 benchmark time-series forecasting datasets. We built FiTEM on top of the TimerXL foundation model, which is a state-of-the-art transformer-based foundation model for time-series forecasting. We evaluate FiTEM in two distinct settings:

1. **Extending non-finetuned models**: We evaluate the performance of FiTEM on the target dataset $\mathcal{D}_t$, where we only train FiTEM parameters on a select subset of the target dataset $\mathcal{D}'_t$. In this setting, the deterministic foundation model f_ψ is only trained on the original source datasets $\mathcal{D}_s$ and FiTEM improves zero-shot performance on the target dataset $\mathcal{D}_t$.
2. **Extending finetuned models**: We evaluate the performance of FiTEM on the target dataset $\mathcal{D}_t$, where we finetune the parameters of the deterministic foundation model f_ψ on the select subset of the target dataset $\mathcal{D}'_t$ and then train FiTEM parameters on the finetuned model. In this setting, the deterministic foundation model f_ψ is trained on the original source datasets $\mathcal{D}_s$ and then finetuned on a subset of $\mathcal{D}'_t$ and FiTEM improves few-shot performance on the target dataset $\mathcal{D}_t$.

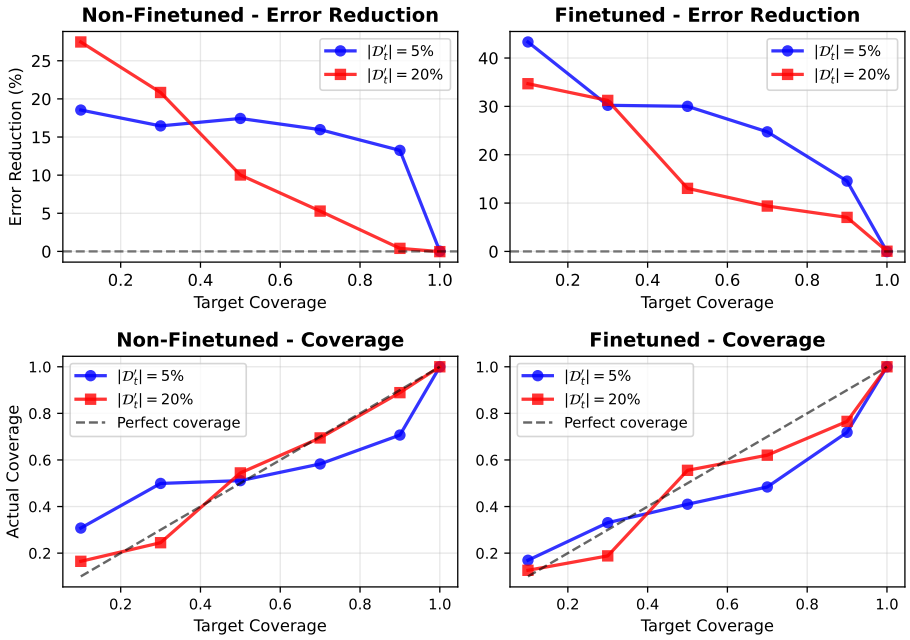

Fig. 2. Selective forecasting results averaged across all target datasets and tested models for non-finetuned and finetuned models, with error reduction percentage (top) and actual coverage expressed as a fraction (bottom) averaged across all target datasets and tested models.

All experiments were conducted using 3 different random seeds to reduce the potential for outliers. Results reported are the second best results out of the 3 runs.

6.1 Selective forecasting performance on non-finetuned models

Table 1. Selective forecasting performance on **non-finetuned** models, where FiTEM was trained on a 5% or 20% subset of the target dataset $\mathcal{D}_t$ to improve zero-shot forecasting performance. Results are shown for target coverages $\phi(g) = 10\%, 30\%, 50\%, 70\%, 90\%$. Each cell contains a pair of the percentage error reduction **in bold** and actual coverage (in parentheses) values for the given target coverage. Percentage error reduction is calculated by comparing the zero-shot forecasting error of the non-finetuned foundation model f_ψ with the selective risk of the FiTEM E_θ trained on top of the non-finetuned model.

Target Coverage	**Timer**		**Timer XL**	
	$\lvert\mathcal{D}'_t\rvert = 5\%$	$\lvert\mathcal{D}'_t\rvert = 20\%$	$\lvert\mathcal{D}'_t\rvert = 5\%$	$\lvert\mathcal{D}'_t\rvert = 20\%$
10%	**15.8%** (34.3%)	**4.2%** (27.9%)	**21.9%** (27.2%)	**56.4%** (5.1%)
30%	**8.8%** (60.4%)	**7.0%** (34.5%)	**26.0%** (39.4%)	**38.1%** (14.4%)
50%	**8.8%** (60.4%)	**3.0%** (70.0%)	**28.1%** (41.8%)	**18.8%** (38.9%)
70%	**7.2%** (70.8%)	**-0.5%** (84.8%)	**26.9%** (45.8%)	**12.5%** (54.1%)
90%	**6.6%** (74.2%)	**-1.6%** (98.0%)	**21.4%** (67.2%)	**2.9%** (79.7%)

As seen in Table 1, FiTEM improves the zero-shot forecasting performance of both the pretrained Timer and TimerXL models on all three datasets when trained on a 5% and 20% subsets of the target dataset $\mathcal{D}_t$. On average across all datasets, FiTEM can reduce forecasting error by up to 15.8% on Timer and 56.4% on TimerXL, when selecting target coverage $\leq 30\%$. When selecting 50% target coverage, FiTEM improves forecasting accuracy by up to 8.8% on Timer and 28.1% on TimerXL, while achieving 60.4% and 41.8% actual coverage respectively. Generally, for non-finetuned Timer and TimerXL models, FiTEM provides a 257% larger improvement in forecasting accuracy for TimerXL when compared to Timer at a cost of lower actual coverage.

We also observe that FiTEM fails to decrease forecasting error for Timer, when trained on 20% of the target dataset $\mathcal{D}_t$, for coverages $\geq 70\%$. This can be explained by Timer's smaller context window, which fails to capture more complex patterns and generates a lower quality latent representation of the time-series data. We also observe that when FiTEM is trained on top of TimerXL with 5% subset of target data $\mathcal{D}'_t$, the forecasting error improvements fluctuate non-monotonically with selected target coverage, while when trained on the 20% subset of target data $\mathcal{D}'_t$, the improvements are more consistent. This is likely due to the very constrained amount of training data available for FiTEM to learn from, and the fact that the model is not able to learn the underlying patterns of the time-series data as well as when trained on the 20% subset of target data $\mathcal{D}'_t$. However, the model can still provide utility by reducing forecasting error, assuming that the user can accept the lower actual coverage as a trade-off. Overall, these results indicate that FiTEM can improve zero-shot forecasting

performance on a target dataset $\mathcal{D}_t$ even when there is not enough data to be used to finetune the deterministic foundation model f_ψ.

6.2 Selective forecasting performance on finetuned models

Table 2. Selective forecasting performance on **finetuned** models, where the deterministic foundation model f_ψ was finetuned on a subset of the target dataset $\mathcal{D}_t$ and FiTEM was trained on a 5% or 20% subset of the target dataset $\mathcal{D}_t$ to improve few-shot forecasting performance. Results are shown for target coverages $\phi(g) = 10\%, 30\%, 50\%, 70\%, 90\%$. Each cell contains a pair of the percentage error reduction **in bold** and actual coverage (in parentheses) values for the given target coverage. Percentage error reduction is calculated by comparing the few-shot forecasting error of the finetuned foundation model f_ψ with the selective risk of the FiTEM E_θ trained on top of the finetuned model.

Target Coverage	**Timer**		**Timer XL**	
	$\lvert\mathcal{D}'_t\rvert = 5\%$	$\lvert\mathcal{D}'_t\rvert = 20\%$	$\lvert\mathcal{D}'_t\rvert = 5\%$	$\lvert\mathcal{D}'_t\rvert = 20\%$
10%	**41.0%** (10.3%)	**44.2%** (20.2%)	**46.2%** (23.6%)	**22.9%** (5.1%)
30%	**25.3%** (29.5%)	**41.3%** (27.0%)	**36.3%** (36.6%)	**18.8%** (10.6%)
50%	**25.7%** (42.2%)	**14.0%** (64.2%)	**35.4%** (39.7%)	**11.8%** (46.9%)
70%	**25.2%** (45.9%)	**8.0%** (70.3%)	**24.2%** (50.9%)	**11.1%** (53.9%)
90%	**9.4%** (74.7%)	**3.6%** (87.0%)	**20.9%** (69.0%)	**11.3%** (66.2%)

As seen in Table 2, FiTEM significantly improves the few-shot forecasting performance of both Timer and TimerXL models when the foundation models are finetuned for 10 epochs. Unlike experiments shown in Section 6.1, the deterministic foundation model f_ψ is now finetuned on subsets of the target dataset $\mathcal{D}_t$ and then FiTEM is trained on top of the finetuned model. The finetuning improves baseline foundation model f_ψ forecasting error and also enables FiTEM to achieve even greater reduction in forecasting error. On average across all settings, FiTEM can reduce forecasting error by up to 44.2% on Timer and 46.2% on TimerXL at low target coverage (10%). At 50% target coverage, FiTEM achieves error reductions of up to 25.7% on Timer and 35.4% on TimerXL, with actual coverage rates of 42.2% and 39.7% respectively. This provides end-users with several potential options for trading off between forecasting error and coverage, depending on the application and the potential penalty for erroneous forecasts.

We observe that FiTEM selective forecasting performance fluctuates depending on the size of the target dataset $\mathcal{D}'_t$ used for training. When trained with 20% of the target dataset $\mathcal{D}'_t$ with Timer, the actual coverage is 40% higher on average (exceeding or nearly matching target coverage) at the expense of 59% decrease in error reduction than when trained with 5% of $\mathcal{D}'_t$. However, this behaviour is generally preferable, since it allows the end-user more direct control over the trade-off between forecasting error and coverage. For FiTEM trained

with TimerXL, the 5% subset of $\mathcal{D}'_t$ shows overall better performance than the 20% subset of $\mathcal{D}'_t$. This was also observed in the non-finetuned experiments, where FiTEM trained with TimerXL and 5% of $\mathcal{D}'_t$ showed better performance than FiTEM trained with TimerXL and 20% of $\mathcal{D}'_t$. This can likely be explained by the fact that TimerXL produces a higher quality latent representation $\mathbf{f}_X$ of the time-series, making the FiTEM decoder converge faster and possibly overfit with the increased number of training samples. However, further experiments will be required to understand the exact conditions that cause this behaviour.

On average, as seen in Figure 2, selective forecasting results on finetuned models are significantly better than the non-finetuned models. The finetuned models are capable of achieving higher error reduction using selective forecasting, while maintaining higher actual coverage. These results demonstrate that FiTEM provides substantial benefits for few-shot forecasting scenarios, with fine-tuning amplifying the selective forecasting improvements across all experimental conditions.

6.3 Model parameters and training time

Table 3. Parameter counts for foundation models f_ψ and FiTEM E_θ.

Model	**Total Parameter count**	**Foundation model** f_ψ **%**	**FiTEM model** E_θ **%**
Timer	69.3M	97.2%	2.8%
TimerXL	68.2M	98.8%	1.2%

As shown in Table 3, FiTEM is very parameter-efficient compared to the foundation models f_ψ it builds on. The FiTEM components constitute only 2.8% and 1.2% of the total model parameters for Timer and TimerXL, respectively. This lightweight FiTEM architecture provides excellent selective forecasting performance with minimal additional computational overhead during fine-tuning or inference, making it suitable for resource-constrained environments and real-time applications.

During experiments, we observed that the training time for FiTEM is comparable to the fine-tuning time for the foundation model f_ψ. As both TS-based models Timer and TimerXL are quite parameter efficient and, as shown in Table 3, the FiTEM components are only a small fraction of the total model parameters, the foundation model fine-tuning and FiTEM training are both computationally inexpensive. Total training time for both fine-tuning f_ψ and training FiTEM is $5-20$ minutes, depending on the size of the subset of the target dataset $\mathcal{D}'_t$ used. In our experiments, no combination of f_ψ finetuning and FiTEM training took longer than 1 hour, when using up to 20% of the target dataset $\mathcal{D}'_t$ for training.

7 Conclusion and Future Work

In this paper, we introduced the Fine-tunable Time-Energy Model (FiTEM), a selective forecasting framework for pre-trained time-series foundation models that can be efficiently trained as an additional step of foundation model finetuning. Our experiments across 3 benchmark datasets demonstrated that FiTEM significantly enhances both zero-shot and few-shot forecasting performance of Timer and TimerXL, reducing errors by up to 56.4% at low coverage levels and up to 35.4% for target coverage of 50% and above. FiTEM's model parameter efficiency and short training times make it practically applicable, not adding substantial computational overhead during both model training and inference. We also provided a detailed analysis of the FiTEM framework, showing that it can be used to improve the reliability of foundation models in a wide range of applications.

For future work, we will extend FiTEM to more TS-based models and add experiments with additional benchmark datasets. Furthermore, we will evaluate more advanced architectures for the FiTEM decoder, based on state-of-the-art time-series analytics models, potentially improving selective forecasting performance further. Finally, we will experiment with using data augmentation strategies to increase the amount of data available for both FiTEM training and foundation model finetuning.

Acknowledgments. This research was partially funded by the 6G-XCEL project under the Horizon Europe programme within the European Union's research and innovation initiatives.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Brusokas, J., Tirupathi, S., Zhang, D., Pedersen, T.B.: The time-energy model: Selective time-series forecasting using energy-based models. Transactions on Machine Learning Research (2025), https://openreview.net/forum?id=iHYCdTAOqF
2. Das, A., Kong, W., Leach, A., Lawrence, M., Martin, A., Sen, R., Yang, Y., Hannachi, S., Kuznetsov, I., Zhou, Y.: A decoder-only foundation model for time-series forecasting. In: Proceedings of the 41st International Conference on Machine Learning (ICML) (2024)
3. Ekambaram, V., Jati, A., Dayama, P., Mukherjee, S., Nguyen, N.H., Gifford, W.M., Reddy, C., Kalagnanam, J.: Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series (2024)
4. Geifman, Y., El-Yaniv, R.: Selectivenet: A deep neural network with an integrated reject option. In: International Conference on Machine Learning (2019)
5. Gruver, N., Finzi, M.A., Qiu, S., Wilson, A.G.: Large Language Models Are Zero-Shot Time Series Forecasters (Nov 2023), https://openreview.net/forum?id=md68e8iZK1

6. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation **14**(8), 1771–1800 (2002). https://doi.org/10.1162/089976602760128018
7. Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J.Y., Shi, X., Chen, P.Y., Liang, Y., Li, Y.F., Pan, S., Wen, Q.: Time-LLM: Time Series Forecasting by Reprogramming Large Language Models (Oct 2023), https://openreview.net/forum?id=Unb5CVPtae
8. Li, Z., Qiu, X., Chen, P., Wang, Y., Cheng, H., Shu, Y., Hu, J., Guo, C., Zhou, A., Wen, Q., Jensen, C.S., Yang, B.: FoundTS: Comprehensive and Unified Benchmarking of Foundation Models for Time Series Forecasting (Nov 2024). https://doi.org/10.48550/arXiv.2410.11802, http://arxiv.org/abs/2410.11802, arXiv:2410.11802 [cs]
9. Liang, Y., Wen, H., Nie, Y., Jiang, Y., Jin, M., Song, D., Pan, S., Wen, Q.: Foundation models for time series analysis: A tutorial and survey. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. p. 6555–6565. KDD '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3637528.3671451, https://doi.org/10.1145/3637528.3671451
10. Liu, Y., Qin, G., Huang, X., Wang, J., Long, M.: Timer-xl: Long-context transformers for unified time series forecasting. In: International Conference on Learning Representations (ICLR) (2025), https://arxiv.org/abs/2410.04803, arXiv:2410.04803
11. Liu, Y., Zhang, H., Li, C., Huang, X., Wang, J., Long, M.: Timer: Generative pre-trained transformers are large time series models. In: Forty-first International Conference on Machine Learning (2024)
12. Nie, Y., Nguyen, N.H., Sinthong, P., Kalagnanam, J.: A time series is worth 64 words: Long-term forecasting with transformers (2023). https://doi.org/10.48550/arXiv.2211.14730, http://arxiv.org/abs/2211.14730
13. Tan, M., Merrill, M.A., Gupta, V., Althoff, T., Hartvigsen, T.: Are language models actually useful for time series forecasting? In: Advances in Neural Information Processing Systems (NeurIPS) (2024), https://arxiv.org/abs/2406.16964, spotlight
14. Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., Long, M.: Timesnet: Temporal 2d-variation modeling for general time series analysis. In: International Conference on Learning Representations (2023), https://arxiv.org/abs/2210.02186
15. Wu, H., Xu, J., Wang, J., Long, M.: Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In: Advances in Neural Information Processing Systems. vol. 34, pp. 22419–22430. Curran Associates, Inc. (2021)
16. Zhang, X.Y., Xie, G.S., Li, X., Mei, T., Liu, C.L.: A survey on learning to reject **111**(2), 185–215 (2023). https://doi.org/10.1109/JPROC.2023.3238024, conference Name: Proceedings of the IEEE
17. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 11106–11115 (2021). https://doi.org/10.1609/aaai.v35i12.17325, https://arxiv.org/abs/2012.07436
18. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In: Proceedings of the 39th International Conference on Machine Learning. pp. 27268–27286. PMLR (Jun 2022)

19. Zhou, T., Niu, P., Wang, X., Sun, L., Jin, R.: One fits all: power general time series analysis by pretrained lm. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2023)