From Pre-training to Post-training: A Survey on Time Series Foundation Models

Zhen Liu, Boyuan Li, Hao Huang, Yanru Sun, Yucheng Wang, Min Wu, and Qianli Ma

***Abstract*—Deep learning models have achieved remarkable progress in time series analysis. However, most models require retraining when encountering new domain data, which limits their generalization and cross-domain transferability. While prior efforts have explored adapting large pre-trained language or vision models to time series tasks, modality gaps hinder their ability to capture complex temporal dynamics. Time Series Foundation Models (TSFMs) trained from scratch have been developed to overcome these limitations, enabling effective knowledge transfer through domain-specific fine-tuning while preserving intrinsic time series characteristics. To this end, this survey offers a comprehensive review of TSFMs trained from scratch, following a unique perspective from pre-training to post-training. Specifically, we present a taxonomy of TSFMs across three dimensions: (1) datasets, discussing the construction of large-scale source domain datasets and the impact of target domains on model evaluation; (2) pre-training, covering task-agnostic and task-specific training paradigms; and (3) post-training, encompassing optimization via supervised, collaborative, and reinforcement-based fine-tuning. Finally, we highlight potential future research directions, including sample quality evaluation, pre-training paradigm design, and agent-based post-training, to advance the development and practical deployment of TSFMs. Our code is available at https://github.com/ZLiu21/awesome-tsfms-from-pre-training-to-post-training.**

***Index Terms*—time series, foundation models, pre-training, post-training, reinforcement learning**

I. Introduction

Time series are sequences of temporally ordered numerical observations, serving as fundamental data forms in pattern recognition and data mining [1], [2]. Many dynamic real-world processes, including weather forecasting [3], medical diagnosis [4], and industrial monitoring [5], can be naturally expressed as time series. Advances in deep learning have driven significant progress in time series forecasting [6], classification [7], anomaly detection [8], and imputation [9], largely by reducing reliance on handcrafted features. However, these models often require training for specific domains and depend on abundant high-quality data, limiting their generalization and adaptability. Developing a unified framework that enables cross-domain knowledge sharing remains a critical challenge.

To develop unified models applicable across diverse scenarios, scholars have introduced Time Series Foundation Models (TSFMs) for downstream time series tasks [10], [11]. Existing TSFMs fall into two main categories: those adapted from large pre-trained language or vision models through fine-tuning, and those trained from scratch on large-scale time series source datasets [12], [13]. Compared with traditional supervised deep learning models, TSFMs demonstrate stronger transferability and improved generalization across heterogeneous domains. However, time series data are characterized by complex temporal dependencies [6], periodic patterns [2], inter-variable correlations [14], and irregular sampling behaviors [15]. Effectively capturing these intrinsic patterns and integrating general knowledge with domain-specific features during fine-tuning are key considerations in advancing TSFMs.

Adapting large pre-trained language and vision models for time series analysis [16], [17] has attracted increasing attention. Some methods discretize continuous temporal signals into tokens for language models [18], while others convert time series into images for visual encoders [19]. Although these approaches have achieved notable results in certain tasks, they often exhibit limited interpretability and weak adaptation to temporal structures. The core issue lies in the distributional discrepancy between pre-training data from textual or visual domains and the intrinsic characteristics of temporal signals, which hampers the models' capacity to capture complex temporal dynamics [20], [21]. Furthermore, converting time series into other modalities easily disrupts their inherent temporal dependencies and may introduce modality-specific noise, thereby degrading representation quality. Given that real-world time series are frequently non-stationary and irregularly sampled, language- or vision-based models remain inadequate for modeling such dynamic and heterogeneous patterns.

Consequently, recent studies have focused on developing from-scratch TSFMs [22], [23], pre-trained directly on large-scale source time series datasets. Such models can capture shared dynamic principles across domains while preserving domain-specific temporal characteristics [11]. Unlike using pre-trained language or vision models, from-scratch TSFMs retain the inherent temporal structure without requiring data-type conversion, thereby enabling more efficient and faithful knowledge transfer [24]. Furthermore, since time series are typically one-dimensional numerical vectors, these models typically require fewer parameters and lower computational resources than those used in vision or language tasks [25], [26]. In high-stakes domains such as healthcare and predictive

Zhen Liu, Boyuan Li, Hao Huang, and Qianli Ma are with the School of Computer Science and Engineering, South China University of Technology, China.

Zhen Liu, Yanru Sun, Yuchen Wang, and Min Wu are with the Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore.

Yanru Sun is with the Tianjin Key Lab of Machine Learning, College of Intelligence and Computing, Tianjin University, China.

Zhen Liu and Boyuan Li are co-first authors.

Min Wu and Qianli Ma are co-corresponding authors. E-mail: wumin@i2r.a-star.edu.sg; qianlima@scut.edu.cn.

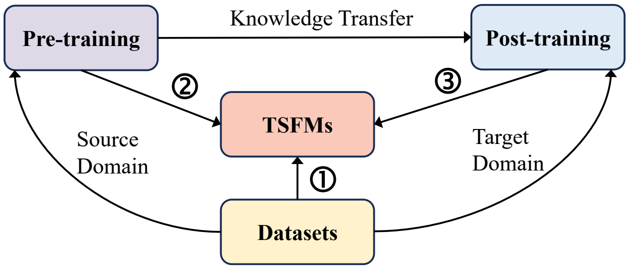


Fig. 1. TSFMs trained from scratch for time series modeling.

TABLE I
COMPARISON OF OUR WORK WITH OTHER RELATED SURVEYS.

Survey	Focus Topic			Data Types	
	Datasets	Pre-training	Post-training	Regular	Irregular
Ma et al. [12]	✗	✓	✗	✓	✗
Miller et al. [31]	✗	✓	✗	✓	✗
Jin et al. [13]	✗	✓	✗	✓	✗
Liang et al. [32]	✗	✓	✗	✓	✗
Ye et al. [33]	✗	✓	✗	✓	✗
Kottapalli et al. [34]	✗	✓	✗	✓	✗
Liu et al. [35]	✓	✗	✗	✓	✗
Our Survey	✓	✓	✓	✓	✓

maintenance [4], [27], from-scratch TSFMs not only improve downstream task performance but also enhance reliability, facilitating human-in-the-loop decision support under privacy and safety constraints. Moreover, they are better equipped to handle irregular time series [28].

This paper presents a comprehensive review of TSFMs, covering the full range from pre-training to post-training. We focus on TSFMs trained from scratch and analyze them along three key dimensions (refer to Figure 1): (1) the construction of large-scale source datasets and evaluation protocols for target domains; (2) the design and optimization of pre-training strategies using these source datasets; and (3) post-training mechanisms for knowledge transfer on target datasets. While some TSFMs have largely emphasized model architectural innovations and pre-training techniques [29], [30], the impact of source dataset selection, sample quality, and post-training optimization warrants full attention. We argue that effective TSFMs require a well-designed source dataset combined with coordinated pre-training and post-training strategies to integrate general knowledge with domain-specific features, thereby improving both generalization and practical applicability.

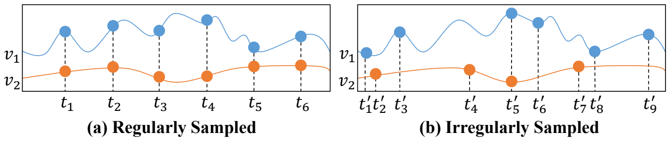
Although several surveys have examined TSFM development, most emphasize pre-training while overlooking the roles of datasets, post-training, and handling irregular time series (see Table II). For example, some studies [12], [31] analyze pre-training strategies without discussing the integration of large-scale, multi-domain datasets. Other surveys [13], [32], [33], [34] primarily adapt the adaptation of pre-trained language and vision models or emphasize pre-training approaches for time series, offering limited exploration of post-training. The work [35] investigates pre-training with synthetic time series yet rarely considers real-world time series data. Unlike previous surveys, this paper adopts a vertical perspective from pre-training to post-training, systematically reviewing from-scratch TSFMs and examining strategies involving both real-world and synthetic source datasets as well as evaluation on target domains. For a detailed discussion of the differences between the aforementioned reviews and the discrepancies with our review, please refer to Appendix A.

The main contributions of this survey are threefold:

- We present the first comprehensive survey of TSFMs from an integrated pre-training to post-training perspective. This review systematically analyzes the modeling characteristics and developmental trends of from-scratch TSFMs across both regular and irregular time series.
- We propose a taxonomy of from-scratch TSFMs based on three core dimensions: *Dataset–Pre-training–Post-training*. It systematically analyzes large-scale source dataset construction, target dataset evaluation, task-agnostic and task-specific pre-training, and post-training optimization via supervised, collaborative, and reinforcement learning.
- We discuss potential future directions in detail, focusing on (i) sampling quality evaluation and domain-specific dataset construction, (ii) pre-training architecture and paradigm design, and (iii) incremental and agent-based post-training.

II. BACKGROUND

A. Time Series Data Types



(a) Regularly Sampled

(b) Irregularly Sampled

Fig. 2. Illustration of time series data types, where v_1 and v_2 are two different variables.

1) Regularly Sampled Time Series

A regularly sampled time series consists of observations collected at uniform time intervals [7], [36], as shown in Figure 2a. For a multivariate time series with V variables, it can be represented as:

$$\mathbf{X}_{\text{regular}} = \{\mathbf{x}_i\}_{i=1}^{T}, \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^V$ denotes the observation vector at time step i, and T represents the total number of time steps.

2) Irregularly Sampled Time Series

An irregularly sampled time series features non-uniform time intervals between observations [17], [15], as shown in Figure 2b. In multivariate settings, differing sampling rates across variables cause temporal misalignment. It can be represented as:

$$\mathbf{X}_{\text{irregular}} = \{[(t_i^v, x_i^v)]_{i=1}^{T_v}\}_{v=1}^{V}, \tag{2}$$

where the v-th variable contains T_v observations, and each observation comprises the timestamp $t_i^v \in \mathbb{R}$ and corresponding value $x_i^v \in \mathbb{R}$.

B. Time Series Properties Modeling

1) Temporal Dependencies

Time series data consists of ordered numerical observations that exhibit intrinsic temporal dependencies, commonly classified as short-term or long-term. Recurrent neural networks [9] and convolutional neural networks [7] effectively model short-term dependencies, whereas Transformer-based models [6] and Mamba architectures [37] better capture long-term patterns. Additionally, multi-scale modeling approaches [38] have been widely employed to jointly learn temporal dependencies across multiple time horizons.

2) Frequency Modeling

Frequency-domain information is essential for many downstream time series tasks [2]. Fourier and wavelet transforms [39], [40] are commonly applied to mitigate sampling noise from sensor errors and to model seasonal or trend dynamics that are difficult to capture in the time domain [41]. Recent studies [42] have further explored frequency-domain modeling for irregular time series.

3) Variable Relationships

In practice, most time series are multivariate, comprising multiple interdependent channels. Nie et al. [20] show that modeling each channel independently rather than jointly can improve forecasting accuracy. Further studies [14], [43] demonstrate that channel clustering or graph neural networks effectively capture inter-variable dependencies, enhancing downstream performance. Thus, effective multivariate modeling requires balancing channel-independent and channel-dependent strategies for the target task.

4) Time Series Shapelets

Shapelets are discriminative subsequences that best characterize the class of a time series instance [44]. They have been widely used to improve the interpretability of classification and clustering models [45], [46]. Recent patch-based modeling methods [20], [11], which treat subsequences as input tokens, demonstrate strong performance in time series modeling. Selecting discriminative patches as shapelets offers a principled approach to improving interpretability while maintaining downstream task performance.

C. Time Series Downstream Tasks

Practical downstream tasks for time series include forecasting [6], classification [7], anomaly detection [8], imputation [47], clustering [45], extrinsic regression [48], change point detection [49], segmentation [50], and retrieval [51]. Existing TSFMs [12], [13], [32] primarily focus on the first four tasks, which are discussed in Appendix B.

D. Why Build TSFMs from Scratch?

Deep learning has achieved notable success in diverse time series tasks. However, most models are trained on small, single-domain datasets, leading to two main limitations: (1) reduced robustness to real-world complexities, including irregular sampling, missing or anomalous values, and noisy or scarce labels [52], [53], [54]; and (2) limited parameter and knowledge transfer, necessitating independent retraining for each target scenario. Alternative approaches adapt pre-trained language and vision Foundation Models (FMs) to time series modeling [18], [19], but they often lack interpretability and may raise security concerns in safety-critical applications. Therefore, building TSFMs from scratch addresses these issues in two ways:

- It more effectively captures the intrinsic properties of time series and enhances interpretability compared with fine-tuning pre-trained language and vision FMs for downstream time series tasks.
- It supports effective post-training adaptation, allowing knowledge learned in the pre-training stage to be efficiently transferred to downstream tasks, which reduces the need for training new models from scratch.

III. Taxonomy of TSFMs

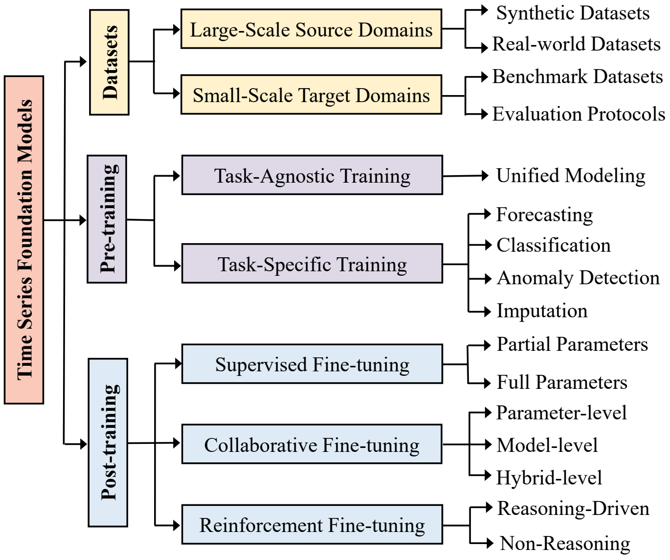


Fig. 3. The taxonomy of TSFMs trained from scratch.

Fig. 3 presents a taxonomy of TSFMs trained from scratch. It comprises three main components: datasets (Section IV), pre-training strategies (Section V), and post-training strategies (Section VI). Notably, the review emphasizes pre-training on large-scale time series datasets (typically over one million samples) rather than transfer learning on small time series datasets, and focuses on post-training strategies for TSFMs trained from scratch.

IV. Time Series Datasets for TSFMs

A. Large-Scale Source Domains

Comprehensive details of source and target datasets are summarized in Tables III and V in Appendix C.

1) Synthetic Datasets

Recent studies [55] show that pre-trained TSFMs exhibit scaling behavior, with performance improving as dataset size and model capacity increase. Beyond architectural design, the quality and scale of pretraining data strongly affect generalization. Synthetic data offers a promising alternative, providing diverse and well-controlled sequences for pretraining. Existing methods generally fall into two categories: mixup-based and intrinsic characteristics-based strategies.

Mixup-based Strategies: Mixup [56], originally introduced for image classification through synthetic sample generation, has been effectively applied to time series downstream tasks [57]. Formally, given two training examples (x_i, y_i) and (x_j, y_j), mixup constructs synthetic samples as:

$$\tilde{x} = \lambda x_i + (1-\lambda) x_j, \quad \tilde{y} = \lambda y_i + (1-\lambda) y_j, \quad \lambda \in [0,1]. \tag{3}$$

This formulation extends the training distribution by enforcing that linear interpolations in the feature space correspond to linear interpolations in the label space. For time series applications, Fatir et al. [22] introduced TSMixup, which generalizes this idea by sampling k subsequences of length l, scaling them, and constructing convex combinations. By ensuring equal contribution of sequences with different magnitudes, TSMixup produces diverse synthetic sequences. Empirical results indicate that such synthetic corpora enhance the generalization of TSFM pretraining.

Intrinsic Characteristics-based Strategies: The intrinsic properties of time series, such as trends, seasonality, nonlinear interactions, and anomalies, have been leveraged for synthetic data generation. A common approach employs Gaussian Processes (GPs) defined via a covariance kernel $\kappa(t, t')$:

$$f(t) \sim \mathcal{GP}(\mu(t), \kappa(t, t')). \tag{4}$$

Kernel composition enables the generation of synthetic sequences that capture seasonality, global trends, and stochastic noise. For instance, Fatir et al. [22] employ GP kernel compositions to create diverse sequences, mitigating the limitations of small forecasting datasets. Xie et al. [58] extend this by integrating GP kernels with structural causal models to produce causally consistent synthetic data for classification pretraining. Taga et al. [59] further propose a GP-based multivariate synthesis method combining kernel compositions with linear coregionalization.

Beyond GPs, synthetic time series data have proven valuable for TSFM pretraining. Rotem et al. [60] created a large-scale univariate dataset of 15 million series with diverse structures, demonstrating transfer learning gains for classification. Dooley et al. [61] generated series with multi-scale seasonality and exponential trends, achieving competitive zero-shot forecasting performance. Emami et al. [62] introduced a short-term load forecasting benchmark combining 900,000 simulated buildings from seven public datasets. Wang et al. [63] proposed a dual-modality series-symbol generation approach, while Lan et al. [64] developed a large anomaly-rich corpus with token-level labels for anomaly detection pretraining.

Summary: Synthetic data has proven effective for TSFM pretraining, enabling scalable model development by increasing the size of source datasets [65], [55]. Key challenges include adapting synthetic corpora to domain-specific contexts (e.g., ECG), and evaluating data quality within principled pretraining paradigms. Additionally, synthetic data shows promise for post-training applications [66]. These studies highlight synthetic data as a central enabler of scalable, versatile, and generalizable TSFM development.

2) Real-world Datasets

Compared with large-scale text corpora like WikiText-103 [67] or image datasets such as ImageNet [68], real-world time series datasets are limited in scale and quality. Unlike text or images with standardized formats, time series vary widely in sampling frequency, dimensionality, and inter-channel relationships across domains [11], [21], complicating the construction of unified corpora for TSFM pretraining. Recent studies [69], [25], [29] have applied advanced pre-processing to aggregate public datasets, producing large-scale sources exceeding one million samples or one billion time points. These datasets typically fall into general-purpose and domain-specific categories.

General-Purpose: Recent efforts to construct large-scale source datasets for TSFMs generally adopt two strategies: (i) hybrid real and synthetic datasets, and (ii) unified real-world datasets. For hybrid datasets, studies expand real data with synthetic sequences to increase scale. For example, Das et al. [70] combine Google Trends, Wikipedia pageviews, and synthetic signals; Fatir et al. [22] integrate 13 public datasets with synthetic samples to form over 11 million series; and Shi et al. [29] introduce Time-300B, comprising over 300 billion time points including real-world and synthetic datasets across nine domains. In addition, for unified datasets, researchers aggregate and standardize real data from multiple domains. Rasul et al. [71] collate 27 datasets spanning diverse sectors; Liu et al. [25] develop UTSD-12G with 1 billion time points across seven domains; Lin et al. [21] merge UCR/UEA and other sources for 1.89 million samples; and Goswami et al. [11] compile the Time Series Pile with 13M samples. Larger repositories include LOTSA (27B observations) [24], extended by Zhang et al. [72] to 260B points. Also, Dempster et al. [73] present MONSTER, which provides 29 datasets spanning six domains and up to 59 million series. Time-IMM [52] explores the use of multi-modality data with irregular time series.

Domain-Specific: Several domain-specific datasets have been developed for TSFM pretraining in healthcare, cloud computing, and electromagnetic signal analysis. Woo et al. [74] introduce large-scale CloudOps datasets containing billions of observations to study TSFM scalability in operational environments. In healthcare, Gow et al. [75] curate the MIMIC-IV-ECG dataset with 800K ECGs, while McKeen et al. [76] compile 1.5M ECGs from multiple clinical sources. Li et al. [28] further aggregate 454B time points from intensive care, operating rooms, long-term monitoring, and epidemiological systems. Beyond healthcare, Luo et al. [27] present EMind, the largest electromagnetic signal dataset, comprising radar, communication, and interference data preserved in raw in-phase and quadrature formats.

Summary: Despite recent progress, constructing a large-scale multivariate time series source dataset remains challenging, as integrating inter-channel dependencies is essential for improving downstream performance on multi-channel tasks. Another key issue is balancing sample proporti. Another key issue is balancing sample proportions across heterogeneous domains, as imbalanced data can bias pre-training and hinder fine-tuning generalization. Furthermore, aligning observations across datasets can be challenging for irregular time series.

B. Small-Scale Target Domains

1) Benchmark Datasets

TSFMs aim to achieve generalizable performance across diverse downstream tasks through pretraining. However, the credibility of their evaluation depends heavily on the selection of benchmarks. To ensure fair assessment, comprehensive and representative benchmarks are essential. This section systematically reviews benchmark datasets commonly used for four major downstream tasks: forecasting, classification, anomaly detection, and imputation.

Time Series Forecasting: Benchmarking in forecasting typically distinguishes between regularly and irregularly sampled datasets. For regular time series, TSFMs are usually evaluated on long-horizon prediction tasks [6]. Representative benchmark datasets include ETT [6], Electricity, Traffic, Weather, Exchange, ILI [77], Solar-Energy, and PEMS [78]. For irregularly sampled forecasting, commonly adopted datasets include MIMIC-III, MIMIC-IV, PhysioNet'12 [79], Human Activity [80], and USHCN [81], which span domains such as clinical monitoring and environmental time series with nonuniform sampling intervals.

Time Series Classification: Classification benchmarks are typically categorized into regular and irregular settings. For regularly sampled data, the UCR archive (128 datasets) is the primary benchmark for univariate classification [82], while the UEA archive (30 datasets) serves as the standard for multivariate classification[83]. However, many TSFM studies evaluate on only a small subset of UEA datasets (often 10), overlooking the full archives and the UCR benchmark. Such selective evaluation risks cherry-picking and compromises fair comparison [82], particularly against advanced models such as InceptionTime [7] and MultiRocket [84]. For irregularly sampled classification, datasets such as MIMIC-III [85], PhysioNet'12 [79], PhysioNet'19 [86], and PAM [87] are commonly used.

Time Series Anomaly Detection: Anomaly detection benchmarks are divided into univariate and multivariate tasks. The UCR anomaly detection archive [54], containing 250 datasets, is the most comprehensive resource for univariate evaluation. Multivariate benchmarks include datasets such as MSL, SMAP, PSM, SMD, SWaT, NIPS-TS-SWAN, and NIPS-TS-GECCO [88], [8]. In contrast, irregularly sampled datasets are rare, likely due to the sparsity of observations that makes anomalies less identifiable.

Time Series Imputation: Imputation benchmarks are generally categorized into regular and irregular sampling settings. For regularly sampled data, datasets such as BeijingAir [89], PeMS [90], ETT, and Traffic [6], as well as classification datasets from the UCR and UEA archives [82], [83], are commonly used. These datasets are typically modified by artificially masking values under MCAR, MAR, or MNAR settings to create synthetic incomplete datasets for imputation evaluation [47]. For irregularly sampled time series, PhysioNet'12 [79] remains the widely used benchmark.

Summary: This section underscores the importance of rigorous dataset selection when evaluating TSFMs. Evaluations based on a limited subset of datasets risk cherry-picking and overestimating generalization [82]. Therefore, task-aligned benchmark selection is essential to ensure fair and credible assessment of TSFMs across diverse downstream tasks.

2) Evaluation Strategy

Fair evaluation of TSFMs requires careful selection of metrics, robust baselines, and consistent dataset preprocessing. Many existing studies, however, adopt prior experimental settings without critically assessing evaluation protocols, which can yield biased or inconsistent results relative to strong task-specific baselines. To address this, this section systematically reviews evaluation strategies for forecasting, classification, anomaly detection, and imputation tasks.

Time Series Forecasting: Evaluating time-series forecasting (TSF) models involves several key design choices that affect the reliability and reproducibility of results. *(i) Window construction and data partitioning.* TSF benchmarks commonly use a sliding-window approach to form input–output pairs $(\mathbf{X}_{t-L+1:t}, \mathbf{X}_{t+1:t+H})$, where L and H denote input and forecasting horizons. Datasets are typically split chronologically into training, validation, and testing subsets (e.g., 6:2:2 or 7:1:2) to prevent future information leakage into model training. *(ii) Evaluation metrics.* TSF Performance is mainly assessed with Mean Absolute Error (MAE) and Mean Squared Error (MSE), sometimes complemented by MAPE [91]. To capture temporal structure, recent works adopt shape-aware metrics such as Shape-Aware Temporal Loss [92] and Patch-wise Structural Loss [93]. Probabilistic metrics like QICE and CRPS [94] are used for uncertainty-aware forecasting. *(iii) Comparative baselines.* Many TSFM studies evaluate only deep-learning models [18], omitting classical statistical and machine learning methods such as ARIMA, ETS, Prophet, and Random Forest [95], which remain competitive in short-horizon or low-variance settings. *(iv) Experimental protocols.* Differences in normalization, random seeds, and batch handling can bias results of TSF. The Drop-last policy requires caution, as inconsistent handling of incomplete batches may artificially improve results or introduce inconsistency [96]. Unified toolkits like TimesNet Benchmark [77] and TFB [96] promote standardized preprocessing and metric computation for TSF evaluation.

Time Series Classification: Existing studies have extensively analyzed TSFMs in forecasting tasks, but evaluation strategies for classification tasks remain refined. Key issues include: *(i) Evaluation metrics.* Most works report average test accuracy on multiple UCR and UEA datasets, which may be biased by the outlier accuracy of one dataset. To address this limitation, statistical metrics such as average rank, p-value [97], and critical difference diagrams [1] should be employed. For imbalanced classification tasks, F1-score, AUROC, and Balanced Accuracy [1] are more suitable. *(ii) Data partitioning.* The UCR and UEA archives include only training and test sets without predefined validation splits. Many TSFMs improperly use the test set for validation [98], leading to unfair comparisons with methods that do not access test data for training. A common approach in existing studies [99] is to evaluate models at the epoch with the lowest training loss. A more appropriate approach is to reserve 10% of the training data for validation [100] or merge the default splits and re-divide them (e.g., 6:2:2) [82], [12] to produce training, validation, and testing sets. *(iii) Baseline selection.* TSFM evaluations often use forecasting-oriented models [11] instead of classification-specific baselines, weakening the effectiveness in classification tasks. For fair comparison, classification-specific baselines should include both deep learning–based methods (e.g., InceptionTime [7]) and convolutional kernel approaches (e.g., MultiRocket [84]).

Time Series Anomaly Detection: TSFMs have achieved remarkable progress in time series anomaly detection (TSAD), yet fairness issues persist in evaluation metrics and baseline selection [54], [8]. *(i) Evaluation metrics.* Most TSFMs rely solely on F1-Score, Precision, and Recall for evaluation. However, Kim et al. [101] reveal that many TSAD methods compute F1 under a "point adjustment" protocol, which may overestimate performance, while Huet et al. [102] demonstrate that classical Precision and Recall can be easily manipulated under weak assumptions. Thus, these metrics fail to capture model reliability. Future TSFM evaluation should incorporate more comprehensive measures, including point-adjusted F1 [101], affiliated Precision/Recall pairs, surface-volume scores [102], AUC-ROC, and VUS-ROC [103]. *(ii) Baseline selection.* Many TSFM studies evaluate TSAD using only deep learning methods, overlooking classical approaches that perform competitively in subsequence and trend anomaly detection [8]. Future benchmarks should include diverse baselines such as HBOS [104] and OCSVM [105] to ensure fair and comprehensive evaluation.

Time Series Imputation: Imputation is one of the key downstream evaluation tasks for TSFMs [106], with two major aspects requiring attention: *(i) Missing mechanisms.* Most existing studies generate artificial missing values on complete datasets to evaluate model performance, typically under the missing completely at random (MCAR) assumption. However, in real-world scenarios, missing at random (MAR) and missing not at random (MNAR) mechanisms are more prevalent [47], and future TSFMs should account for these more complex patterns. *(ii) Evaluation metrics.* Most existing studies assess imputation accuracy primarily using Mean Squared Error (MSE) and Mean Absolute Error (MAE). However, the objective of time series imputation extends beyond restoring missing values, thus aiming to generate complete sequences that enhance the performance of downstream tasks such as forecasting, classification, and anomaly detection. Thus, future imputation methods should be evaluated not only by reconstruction accuracy but also by their ability to improve downstream task performance (e.g., forecasting) [107], ensuring stronger alignment with real-world application needs.

Summary: Evaluation strategies for TSFMs across downstream tasks remain insufficiently standardized and lack methodological rigor. In forecasting, inconsistencies in preprocessing, baseline selection, and batch handling compromise comparison fairness. In classification, using test sets for validation and relying on average accuracy reduce result reliability. In anomaly detection, overreliance on F1-score, Precision, and Recall can inflate performance. In imputation, neglecting non-random missing mechanisms (MAR and MNAR) and insufficient assessment of imputed sequences for downstream tasks limits real-world applicability.

V. Time Series Pre-training for TSFMs

Table IV in Appendix D summarizes task-agnostic and task-specific TSFMs trained from scratch.

A. Task-Agnostic Training

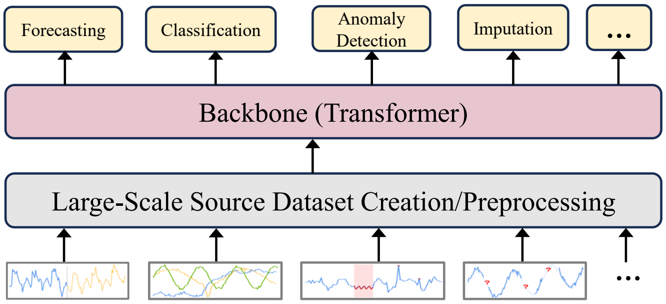


Fig. 4. Illustrative of TSFMs for task-agnostic pre-training.

Recent work [11], [25] pays attention to task-agnostic Time Series Foundation Models (TSFMs), which use massive source datasets fed into backbones (typically Transformers) for pretraining. The pretrained models are subsequently fine-tuned for diverse downstream tasks, including forecasting, classification, anomaly detection, and imputation, as illustrated in Figure 4. This paradigm shift from single-domain pipelines highlights the move toward general-purpose modeling. Task-agnostic TSFMs can be broadly classified into general-purpose and domain-specific designs.

General Scenarios: General-purpose TSFMs aim to exploit large-scale source datasets for pretraining, enabling the learning of transferable representations across diverse downstream tasks. Goswami et al. [11] introduce MOMENT, a family of open-source Transformer encoders pretrained via masked time-series modeling to minimize reconstruction error. Liu et al. [25] propose a decoder-only Transformer pretrained on datasets exceeding one billion time points using next-token prediction. Lin et al. [21] develop NuTime, a Transformer encoder pretrained on over one million sequences from heterogeneous sources using self-supervised objectives. Gao et al. [69] present UniTS, a unified framework that tokenizes predictive and generative tasks within a single Transformer-based model, facilitating transfer across domains with distinct temporal dynamics and resolutions. Similarly, Zhang et al. [72] propose TimesBERT, trained on 260 billion time points to capture temporal and variable dependencies, excelling across major downstream tasks. Chen et al. [108] explore federated TSFM training to address data heterogeneity while maintaining performance. Also, Ekambaram et al. [100] design ultra-compact models with only one million parameters using dual-space masked reconstruction, achieving strong results in classification, anomaly detection, imputation, and retrieval.

Domain-specific: Several TSFMs are designed for domain-specific pretraining. Lunelli et al. [109] develop an xLSTM-based self-supervised model for ECG analysis that surpasses existing state-of-the-art methods. Luo et al. [27] propose a foundation model for electromagnetic signal analysis, integrating large-scale pretraining with modality-specific structures to enable broad generalization across applications. Yang et al. [110] introduce WirelessGPT, a foundation model for multi-task learning in wireless communication and sensing, achieving substantial improvements in channel estimation, prediction, and human activity recognition. Hung et al. [111] present D-BETA, a contrastive masked autoencoder that jointly pretrains ECG signals and textual data, demonstrating superior performance in arrhythmia classification and patient identification tasks.

Summary: Task-agnostic TSFMs demonstrate strong potential in learning unified representations transferable across diverse downstream tasks, representing both a major challenge and an important direction in time series modeling. In general-purpose settings, although several TSFMs achieve competitive results across tasks, studies [112], [113], [114] indicate that their advantages often concentrate on specific tasks (e.g., forecasting), with limited comparison against leading task-specific supervised baselines in other domains. This gap remains a key issue in evaluating model generality. Domain-specific TSFMs, such as those for ECG, electromagnetic signals, or financial data [115], show that pretraining within specialized modalities enhances fine-tuning effectiveness. Integrating general-purpose modeling with domain-aware pretraining provides a promising route toward more universally robust TSFMs.

B. *Task-Specific Training*

Compared with task-agnostic TSFMs, task-specific training paradigms can better exploit multi-source datasets to capture intrinsic temporal features that align with the objectives of downstream tasks. As a result, task-specific TSFMs often demonstrate stronger generalization and adaptability in target applications. Therefore, this section focuses on common task-specific pre-training paradigms developed for four representative downstream tasks in TSFMs.

1) *Forecasting*

Forecasting remains the most extensively studied pre-training objective for TSFMs [24], [116], [23]. Since forecasting benchmarks often exhibit complex temporal dynamics and inter-variable correlations, TSFM pre-training aims to capture dependencies both over time and across variables. However, when scaling a single large model across diverse datasets, effectively modeling temporal dynamics and variable interactions becomes increasingly challenging.

Temporal Modeling: The formulation of time series forecasting naturally aligns with the next-token prediction paradigm, as shown in Figure 5 (a), where temporal modeling is performed by sequentially predicting future observations from past inputs. Given the shared sequential nature of textual and temporal data, early works reprogrammed pre-trained large language models (LLMs) for forecasting by leveraging their next-token prediction capability [18], [117]. Recent studies pre-train native time-series foundation models directly on large-scale datasets, retaining the autoregressive formulation while optimizing all parameters end-to-end for temporal dependency modeling [29], [70], [118]. For *regularly sampled* data, recent TSFMs largely follow the decoder-only autoregressive paradigm but differ in how they scale context length, model capacity, and output distribution. Liu et al. [116] extends next-token prediction to multivariate settings and introduces a causal TimeAttention to capture long-range dependencies across dimensions. Building on this, Shi et al. [29] scales forecasting to billion-scale data through a sparse mixture-of-experts design, activating only a few experts per step to balance capacity and efficiency. Zhang et al. [119] propose TimeRAF, which further enhances zero-shot forecasting by integrating retrieval augmentation, where a learnable retriever injects external time-series knowledge through channel prompting. While these models focus on scalability and generalization, Liu et al. [30] introduce Moirai-MoE, which shifts toward distributional diversity by introducing token-level sparse routing within the Transformer to model heterogeneous temporal patterns. Finally, Liu et al. [23] propose Sundial, which unifies forecasting and generative pre-training through a flow-matching objective, enabling continuous next-patch modeling without discrete tokenization. For *irregularly sampled* data, foundation models extend autoregressive forecasting to handle non-uniform temporal dynamics. Chowdhury et al. [120] pre-trains on irregular multivariate series using a constant time-masking scheme combined with reconstruction and contrastive objectives, improving temporal alignment under uneven sampling. Xiao et al. [121] further learns a continuous latent space over patchified inputs, providing a unified representation framework that generalizes across both regular and irregular temporal patterns.

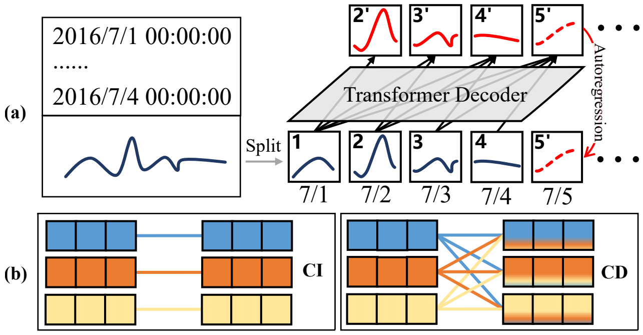


Fig. 5. An illustration of TSFMs pre-training for forecasting. (a) Next-token prediction with causal attention. (b) Channel Independence (CI) and Channel Dependence (CD) strategies.

Variable Modeling: Another major challenge in pre-training TSFMs lies in handling differences in the number of variables across datasets. To ensure dimensional consistency across heterogeneous datasets, most existing TSFMs adopt a channel-independent design that typically flattens multivariate inputs into a single sequence and shares encoder parameters across channels [70], [118], [122], as illustrated in Figure 5 (b). While this strategy simplifies large-scale pre-training, it removes inter-variable structures, rendering explicit dependency modeling techniques less applicable. Recently, several studies have begun to incorporate channel interactions during pre-training. Liu et al. [25], [116] introduces a variable dependency matrix that captures cross-channel relations even within flattened sequences, enabling the model to jointly encode temporal and variable dependencies. Zhang et al. [123] adopts a two-stage paradigm: univariate pre-training for scalability followed by multivariate fine-tuning to reintroduce variable-specific adaptation. Ansari et al. [124] employs a group attention mechanism to facilitate dependency learning across multiple time series.

Summary: For temporal modeling, the next-token prediction paradigm have proven effective for capturing dependencies across datasets with varying temporal characteristics. Nevertheless, research on pre-training for irregularly sampled or missing-value time series remains limited, constraining the applicability of current TSFMs in real-world scenarios. Recent advances have integrated temporal and variable modeling within unified attention mechanisms, which improves efficiency but may limit specialized representation learning. While unified modeling improves scalability, architectures that balance or selectively decouple temporal and variable dependencies are essential to achieving both efficiency and specialization.

2) Classification

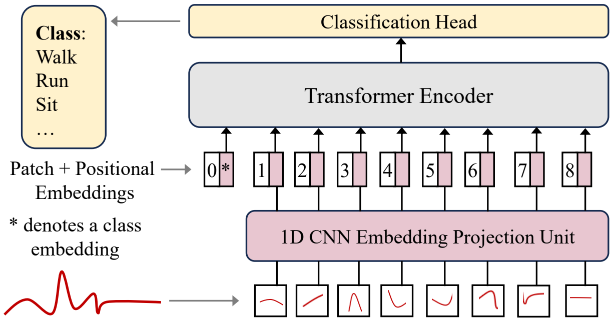

Class:
Walk
Run
Sit
…
Classification Head
Transformer Encoder
Patch + Positional Embeddings
0 *
1
2
3
4
5
6
7
8
* denotes a class embedding
1D CNN Embedding Projection Unit

Fig. 6. Illustrative of TSFMs pre-training for classification task.

Time series classification differs fundamentally from forecasting, as it focuses on extracting discriminative temporal patterns from entire sequences for categorical judgment. A commonly adopted TSFM architecture for classification tasks is inspired by the Vision Transformer (ViT) [125], as shown in Figure 6. In this design, sequences are segmented into fixed-length subsequences, embedded through 1D convolution and positional encoding, and processed by a Transformer encoder with a prepended class token, whose output is passed to a classification head. Yeh et al. [126] demonstrate that this pretraining approach, combined with the Transformer, yields superior performance. Lin et al. [21] and Feofanov et al. [26] further validate its effectiveness on the UCR and UEA benchmarks [82], [83], confirming its robustness as a pretraining backbone. Existing TSFMs for classification can be broadly categorized into general-purpose and domain-specific models.

General Scenarios: Early studies on time series classification primarily employed RNN- and CNN-based approaches. Malhotra et al. [127] proposed unsupervised reconstruction pretraining using multilayer RNNs on 18 UCR datasets, while Kashiparekh et al. [128] utilized CNNs on the same datasets, training independent classification layers to enable transfer learning. Ismail Fawaz et al. [129] conducted a large-scale evaluation on 85 UCR datasets, showing that transfer learning can either enhance or degrade performance depending on the choice of source dataset, thus emphasizing the importance of dataset selection. Rotem et al. [60] developed a scalable, architecture-agnostic transfer learning framework trained on 15 million synthetically generated univariate time series, facilitating adaptation to diverse CNN-based models. With the emergence of Transformers, Pu et al. [112] compared RNN-, CNN-, and Transformer-based pretraining across 150 datasets, demonstrating that pretraining mainly accelerates optimization but is still limited by dataset scale. To mitigate this issue, Ismail-Fawaz et al. [130] aggregated UCR datasets into domains and applied InceptionTime [7] for cross-domain pretraining, achieving notable gains over training from scratch. Building on these advances, Lin et al. [21], Feofanov et al. [26], and Liu et al. [131] combined UCR, UEA, and eight additional datasets [82], [83] into a large-scale corpus of 1.89 million sequences, validating the effectiveness of ViT-based Transformer pretraining for classification. Similarly, Zhang et al. [72] introduced TimesBERT, further confirming the suitability of Transformer encoder-only architectures for classification pretraining.

Domain-specific: Several studies have investigated domain-specific TSFMs for classification tasks. Huang et al. [132] proposed a TSFM for medical time series that disentangles domain-specific medical knowledge from task-specific components to mitigate data heterogeneity. Zhang et al. [133] introduced a contrastive learning framework for human activity time series, aligning temporal data with text descriptions generated by large language models to enable multi-modal pretraining. Bickmann et al. [134] developed CardX, an open-source ECG foundation model trained from scratch on over one million recordings, which outperforms existing foundation models while using fewer parameters and lower computational resources. Similarly, Zhang et al. [4] presented an ECG foundation model pretrained on more than one million clinical recordings, enhancing representation learning for ECG time series classification.

Summary: Classification-based TSFMs show clear advantages through pretraining on large-scale, multi-domain datasets, outperforming models trained from scratch on small domain-specific data. However, most existing approaches focus on univariate inputs, leaving inter-variable dependency modeling insufficiently addressed. In addition, classification pretraining under irregular sampling and missing values remains underexplored, highlighting the need for more robust pretraining strategies and multi-variable representation learning.

3) Anomaly Detection

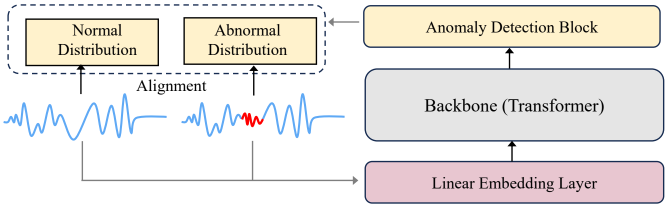


Fig. 7. Illustrative of TSFMs pre-training for the anomaly detection task.

Time series anomaly detection focuses on training models capable of distinguishing normal from abnormal value distributions within sequential data. As illustrated in Figure 7, recent studies [135], [114] employ Transformer-based architectures to capture the statistical characteristics of both normal and anomalous patterns. Evidence from existing work [135], [136] indicates that encoder-only Transformer models effectively represent these distributions, establishing a good foundation for developing TSFMs.

Specifically, Yan et al. [137] provide a comprehensive survey of deep transfer learning methods for time-series anomaly detection, highlighting applications in manufacturing, predictive maintenance, energy, and infrastructure monitoring. Their study underscores the strong potential of foundation models trained from scratch for anomaly-related predictive tasks. Beyond transfer learning, TSFMs pretrained on general time-series tasks have been evaluated. Shyalika et al. [113] show that scratch-trained models (TimeGPT [138], MOIRAI [24], Time-MOE [29], Chronous [22]) often underperform specialized anomaly detection methods. Hela et al. [114] report that MOMENT [11] surpasses traditional baselines on unsupervised smart meter anomaly detection. Maru and Sato [139] combine Time-MoE [29] and MOMENT [11] into a retrieval-augmented TSFM, achieving strong, domain-independent performance on the UCR anomaly detection archive [54]. González et al. [140] introduce foundation auto-encoders integrating variational auto-encoders with dilated CNNs trained on approximately 750,000 time-series samples, showing effective zero-shot anomaly detection capabilities. Shentu et al. [141] propose a unified framework pre-trained on heterogeneous time series datasets from multiple domains, enabling a single model to generalize across diverse anomaly detection tasks without domain-specific training.

Summary: Current research on TSFMs for anomaly detection primarily extends task-agnostic TSFMs, which were originally developed for forecasting or classification tasks. Promising directions, such as retrieval-augmented architectures, have been explored; however, the design of TSFMs specifically optimized for anomaly detection remains at an early stage. Progress in this area relies on the availability of large-scale benchmark datasets tailored for anomaly detection and on the refinement of pretraining strategies to effectively capture complex multivariate dependencies, thereby enhancing the performance of anomaly-focused TSFMs.

4) Imputation

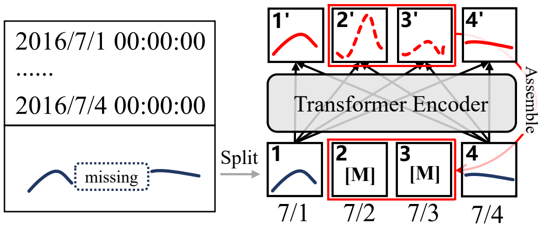


Fig. 8. An illustration of TSFMs pre-training for imputation via masked full attention, where [M] denotes masked values.

Existing TSFMs for imputation are often adapted from those designed for forecasting, where models should distinguish imputation targets within time series inputs. An illustration of TSFMs pre-training for imputation is available in Figure 8. The challenges of temporal and variable modeling have been discussed in Section V-B1, and we focus on handling imputation targets here.

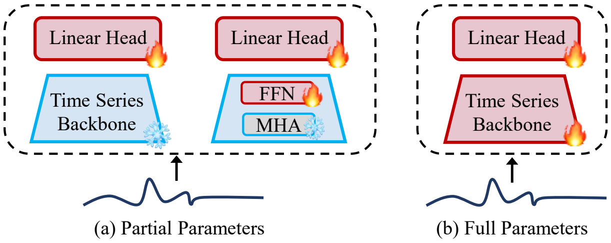
During pre-training, most models employed patch masking as imputation targets for input time series, using reconstruction as the training objective. For regular time series inputs, masked attention offers a straightforward strategy for handling imputation targets by suppressing attention scores associated with these targets. For example, Zhou et al. [16] employ GPT-2 as the backbone transformer, leveraging its attention mechanism to model imputation tasks. Goswami et al. [11] implement masking in the embedding space, replacing the corresponding patch embeddings with dedicated mask embeddings. Liu et al. [25] adopt causal masking to constrain predictions to past observations, a strategy primarily designed for forecasting rather than imputation. For irregular time series inputs, Chowdhury et al. [120] applied a constant time masking technique to preserve the original sampling rate information.

Summary: Reconstruction learning with masked imputation targets is the predominant pre-training approach for imputation tasks. However, some TSFMs inadequately address the differences between forecasting and imputation. For instance, Timer [25] employs causal masking to restrict attention to past observations, a design suitable for forecasting but suboptimal for imputation. Beyond masking strategies, pre-training objectives play a critical role. Mean squared error is commonly used as the training loss, yet it does not capture the temporal fluctuations of time series. Loss functions that account for the dynamics of both observed and imputed values can enhance the effectiveness of TSFM pre-training for imputation tasks.

VI. Time Series Post-training for TSFMs

Table VI in Appendix E summarizes the differences among supervised, collaborative, and reinforcement fine-tuning strategies for TSFM post-training.

A. *Supervised Fine-tuning*



(a) Partial Parameters

(b) Full Parameters

Fig. 9. Illustrative of TSFMs post-training based on supervised fine-tuning. MHA denotes multi-head attention, and FFN represents feed-forward network.

TSFMs transfer knowledge learned from large-scale source datasets to downstream tasks with limited target-domain data through fine-tuning. A common paradigm is supervised fine-tuning, where model parameters are updated using labeled target-domain data. Two primary approaches are typically adopted: partial-parameter and full-parameter fine-tuning, as illustrated in Figure 9. In the partial-parameter setting, only a subset of model parameters is updated. This can be achieved by fine-tuning a lightweight linear head or selected linear layers within the backbone, thereby reducing training cost [142], [143]. In contrast, full-parameter fine-tuning updates all model parameters, generally achieving stronger downstream performance at the expense of higher computational cost [21], [111]. The following sections discuss existing methods for both partial- and full-parameter fine-tuning in detail.

1) Partial Parameters

Partial-parameter fine-tuning adapts TSFMs to target tasks by updating only a small subset of parameters while keeping most pretrained weights frozen. This is typically achieved by adding a lightweight trainable head or selectively updating specific layers within the backbone. For instance, Kashiparekh et al. [128] investigated fine-tuning strategies for a four-layer CNN pretrained on diverse univariate time series classification datasets. They proposed four variants that froze different combinations of convolutional layers and employed a linear classification head, demonstrating that updating the final convolutional layer was crucial for strong classification performance. More recently, Qiao et al. [38] proposed a multiscale fine-tuning framework for TSFMs, which freezes the backbone while integrating scale-specific, parameter-efficient modules into the input projection and attention layers. This design enabled efficient adaptation and yielded substantial improvements in long time series and probabilistic forecasting tasks.

Summary: Partial-parameter fine-tuning provides computational efficiency and preserves most pre-trained knowledge but may underperform when extensive task-specific parameters are required. Research on TSFMs has primarily focused on multiscale modeling and selective fine-tuning of specific layers. Ongoing efforts can emphasize parameter-efficient strategies that explicitly exploit time-series intrinsic properties, such as frequency-domain periodicity, discriminative shapelet patterns, and inter-variable dependencies, to enhance fine-tuning effectiveness.

2) Full Parameters

Full-parameter fine-tuning involves updating all learnable weights of a TSFM during adaptation to downstream tasks. This strategy fully exploits the representational capacity of the pretrained model without restricting adaptation to specific modules. Gao et al. [69] compared full-parameter fine-tuning with lightweight prompt-based fine-tuning across forecasting, classification, and anomaly detection under few-shot settings, finding that full fine-tuning generally achieved superior downstream performance. Similarly, Lin et al. [21] reported that, for time series classification, updating all Transformer parameters significantly outperformed methods that only fine-tuned the linear classification head. Extending these results, Feofanov et al. [26] conducted a systematic evaluation on the UCR and UEA classification archives [82], [83], showing that full-parameter fine-tuning improved average accuracy by approximately 7% compared with linear-head tuning.

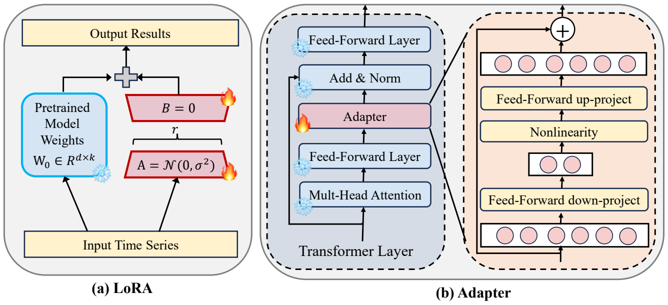
Summary: Full-parameter fine-tuning provides strong empirical advantages by allowing models to fully utilize pre-trained knowledge for task-specific adaptation. However, this approach is computationally demanding, requiring significant resources for deployment, which constrains its applicability in resource-limited settings. A central challenge is developing fine-tuning strategies that align pretrained modules with the statistical characteristics of downstream time series data while preserving computational efficiency.

B. Collaborative Fine-tuning

Collaborative fine-tuning aims to enhance pre-trained TSFMs by incorporating external modules or auxiliary models during post-training adaptation. Based on the level of collaboration and degree of external dependency, these methods can be categorized into three hierarchical levels: Parameter-level Collaboration (PLC), Model-level Collaboration (MLC), and Hybrid-level Collaboration (HLC). Specifically, PLC focuses on parameter-efficient adaptation of pre-trained TSFMs through lightweight external modules; MLC emphasizes multimodal collaboration with large language or vision foundation models; whereas HLC lies between the two, leveraging medium-scale models to achieve efficient pre-trained knowledge transfer.

1) Parameter-level Collaboration

PLC introduces lightweight trainable modules to efficiently fine-tune pretrained TSFMs. LoRA and Adapter have been proven effective parameter-efficient tuning techniques in NLP and CV: LoRA adapts tasks via low-rank matrices within Transformer layers, while Adapter inserts bottleneck modules to freeze backbone weights and enable task-specific adaptation. Following this principle, both methods can achieve parameter-efficient fine-tuning for TSFMs, as illustrated in Figure 10. This section outlines their implementation and applicability to TSFMs post-training.



(a) LoRA (b) Adapter

Fig. 10. Illustration of TSFM post-training frameworks based on (a) Low-Rank Adaptation (LoRA) and (b) Adapter modules.

LoRA: Low-Rank Adaptation (LoRA) enables efficient task adaptation by inserting trainable low-rank matrices into Transformer layers while keeping pretrained weights frozen, significantly reducing post-training computational cost [144]. Specifically, for a pre-trained model weight matrix $W_0 \in \mathbb{R}^{d \times k}$, the update is reparameterized as

$$W_0 + \Delta W = W_0 + BA, \tag{5}$$

where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and $r \ll \min(d, k)$. During fine-tuning, W_0 remains fixed while A and B are trainable, achieving parameter-efficient adaptation. As illustrated in Figure 10a, this mechanism efficiently adapts dense layers without altering pretrained weights.

Recent studies have applied LoRA to time series modeling. Nie et al. [145] proposed a channel-aware low-rank adaptation that leverages identity-conditioned components to capture inter-variable dependencies while preserving training efficiency. Further, LoRA has been applied to fine-tune TSFMs trained from scratch. Gupta et al. [146] showed that LoRA-adapted TSFMs, including Lag-Llama [71], MOIRAI [24], and Chronos [22], exhibit strong zero-shot performance on forecasting tasks. Wu et al. [147] proposed an uncertainty-aware fine-tuning framework that combines LoRA with resource-efficient mixture-of-expert modules, improving robustness across TSFMs, particularly for anomaly detection. Pan et al. [148] developed a two-stage mixture-of-LoRA approach, pre-training a model for one-step prediction and adapting it with step-specific LoRA modules. Zhou et al. [149] introduced a holistic framework integrating mixed-order optimization, LoRA, and layer-dependent updates, enhancing computational efficiency for ECG-based cardiovascular disease detection.

Adapter: Early studies [150] on text and image data demonstrated that adapters can match the performance of full fine-tuning while using two orders of magnitude fewer trainable parameters. Unlike supervised fine-tuning, adapter tuning freezes the pretrained backbone and incorporates lightweight task-specific modules, allowing extensive parameter sharing across tasks. Formally, for a pretrained model $\phi_w(x)$ with parameters w, adapter tuning defines a modified mapping:

$$\psi_{w,v}(x) = \phi_w(x) + A_v(\phi_w(x)), \tag{6}$$

where $A_v(\cdot)$ is the adapter module with trainable parameters v, while w remains fixed. Adapters typically use a bottleneck structure, projecting d-dimensional features to a lower-dimensional space m, applying a nonlinearity, and projecting back to d. Each adapter layer contains $2md+d+m$ parameters, substantially fewer than $|w|$ when $m \ll d$. Figure 10b illustrates the integration within a Transformer layer [150], with the adapter inserted after the projection in both the attention and feed-forward layers.

Building on this foundation, recent studies examine how adapters can be tailored for time series modeling. Early work integrates adapters into LLMs for time series tasks. For example, Niu et al. [151] incorporate temporal, channel, frequency, and anomaly adapters into each Transformer block of a pretrained LLM, enabling efficient fine-tuning for temporal dynamics. For adapters applied to TSFMs trained from scratch, Ilbert et al. [152] explore both classical dimension-reduction strategies (e.g., random projection, variance-based selection) and neural network-based adapters for multivariate time series classification. Although many of these methods are non-trainable, they perform a similar role to adapters by enabling parameter-efficient adaptation of TSFMs to downstream tasks. Benechehab et al. [153] introduce probabilistic adapters to extend univariate TSFMs to multivariate forecasting, applying stochastic feature transformations in a latent space while keeping the pretrained backbone frozen, thereby supporting probabilistic modeling with minimal additional cost.

Summary: LoRA and Adapter offer parameter-efficient fine-tuning frameworks that allow TSFMs to adapt to downstream time series tasks with substantially fewer trainable parameters. LoRA-based methods have been shown to enhance modeling of inter-variable dependencies and improve forecasting accuracy, particularly in zero-shot scenarios. Adapter-based approaches primarily focus on extending univariate TSFMs to multivariate settings, reducing fine-tuning costs and guiding architectural choices for multivariate pretraining. However, existing LoRA and adapter designs seldom leverage intrinsic time series features (i.e., shapelets), and strategies for efficiently fine-tuning irregular time series are limited. Addressing these gaps is critical for developing more effective and efficient LoRA- and adapter-based post-training strategies for TSFMs.

2) *Model-level Collaboration*

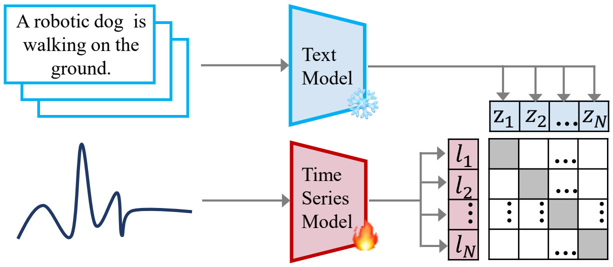


Fig. 11. Illustrative of post-training based on multimodal learning.

Model-level collaboration focuses on integrating time series models with large language or vision foundation models to enhance downstream performance. Prior studies [154] demonstrate that using natural language as auxiliary supervision in image–text representation learning, as in the *CLIP* paradigm, improves generalization and enables zero-shot transfer by aligning paired images and texts in a shared embedding space via contrastive learning [155]. This paradigm can be directly applied to TSFMs (refer to Fig. 11), where a time series encoder replaces the image encoder, a frozen LLM generates textual embeddings, and alignment is achieved by projecting both modalities into a shared normalized embedding space:

$$Z_X = \text{Norm}\big(f_\theta(X)W_x\big), \quad Z_T = \text{Norm}\big(g_\phi(T)W_t\big), \tag{7}$$

where f_θ and g_ϕ denote the time series and text encoders, and W_x, W_t are projection matrices. The training objective typically employs a symmetric contrastive loss [155], encouraging paired samples to be close in the embedding space while separating mismatched pairs.

Recent studies show that integrating textual and visual modalities with time series can improve downstream task performance [156], [157]. Liu et al. [158] proposed a dual-modality forecasting framework combining a dedicated encoder for disentangled time series representations with a language-guided branch generating prompt-based embeddings. Chen et al. [157] developed a series–image contrastive framework that leverages visual representations to augment structural information and enhance classification generalization. Recently, some studies have explored multimodal strategies for TSFMs to enhance post-training effectiveness [159]. Zhang et al. [133] proposed a unified pre-training framework for motion time series, aligning temporal data with LLM-enhanced text via contrastive learning, achieving strong classification with limited labels. Wang et al. [160] treated time series as a "foreign language", developing a multimodal TSFM that enables efficient fine-tuning across forecasting tasks. Wu et al. [161] introduced a cross-modal attention mechanism to integrate visual, textual, and physiological signals for anomaly detection, promoting robust generalization in medical applications. Liu et al. [162] presented the CALF framework, mitigating distributional discrepancies between textual and temporal modalities to improve multivariate forecasting. Chen et al. [163] proposed a lightweight multimodal retriever that aligns time series and text embeddings for task-specific adaptation.

Summary: Current work demonstrates that integrating textual and visual modalities with time series data effectively leverages pretrained language and vision models to enhance time series representations, thereby improving performance across diverse downstream tasks. However, existing multimodal post-training approaches rarely exploit intrinsic time series properties, such as multiscale patterns or frequency-domain features. Furthermore, the development of unified multimodal TSFMs that serve multiple downstream tasks remains a key research direction.

3) *Hybrid-level Collaboration*

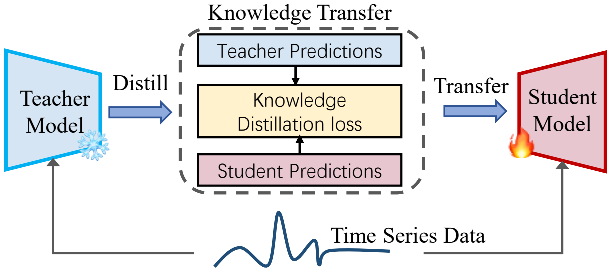


Fig. 12. Illustrative of post-training based on knowledge distillation.

Hybrid-level collaboration (HLC) bridges parameter-level and model-level approaches by leveraging medium-scale external models for TSFMs post-training. A natural realization of HLC is knowledge distillation (KD) [164], which transfers knowledge from a large teacher model to a smaller student model, facilitating deployment on resource-constrained devices without sacrificing performance. As shown in Fig. 12, the pre-trained TSFM serves as a frozen teacher, and the student model is optimized a carefully designed distillation loss. A common approach is feature-based knowledge transfer, defined as:

$$L_{\text{FeaD}}(f_t(x), f_s(x)) = L_F(\Phi_t(f_t(x)), \Phi_s(f_s(x))), \quad (8)$$

where $f_t(x)$ and $f_s(x)$ denote intermediate feature representations extracted from the teacher and student models, respectively. As teacher and student feature maps may differ in dimensionality, transformation functions $\Phi_t(\cdot)$ and $\Phi_s(\cdot)$ are applied to achieve alignment. The similarity function $L_F(\cdot)$ quantifies the discrepancy between the transformed features, guiding the student to inherit the teacher's representational capabilities. This formulation underpins feature-based KD for TSFMs.

Recent studies have demonstrated the effectiveness of KD in time series modeling across diverse downstream tasks. For example, Xing et al. [165] proposed an efficient federated distillation framework for multitask time series classification, combining feature-based student–teacher learning with distance-based weight alignment. KD has also been applied directly to TSFMs trained from scratch. Yang et al. [166] developed a two-stage framework transferring knowledge from multiple speech-based TSFMs into a unified neural transducer for automatic speech recognition. Abbaspourazad et al. [167] distilled representations from photoplethysmogram encoders into wearable accelerometry encoders. To reduce computational costs in ECG-based disease detection, Zhou et al. [149] transferred the representational capacity of pre-trained TSFMs into lightweight student models. Chan et al. [168] further proposed a degradation-aware fine-tuning approach, distilling knowledge from TSFMs into compact expert models.

Summary: KD has proven effective for enhancing downstream time-series tasks, enabling TSFMs to reduce training costs during post-training. However, current KD applications in TSFMs primarily rely on generic distillation strategies and rarely incorporate time series representational learning in the design of distillation losses. Promising directions include developing feature-aware KD tailored to temporal structures and extending KD to multimodal settings, where rich representations from heterogeneous modalities such as sensor, speech, or physiological signals can be transferred to improve student models.

C. *Reinforcement Fine-tuning*

Reinforcement fine-tuning integrates the principles of Reinforcement Learning (RL) into the post-training phase of TSFMs [169], enabling optimization through interaction with feedback from the environment rather than explicit supervision signals. In contrast to supervised and collaborative fine-tuning, which depends on gradient-based optimization guided by differentiable losses such as MSE or cross-entropy, RL adopts a reward-driven optimization process based on task-specific or expert-defined feedback [170]. This paradigm allows TSFMs to incorporate environmental information and human preferences without the need for differentiable losses during post-training [171].

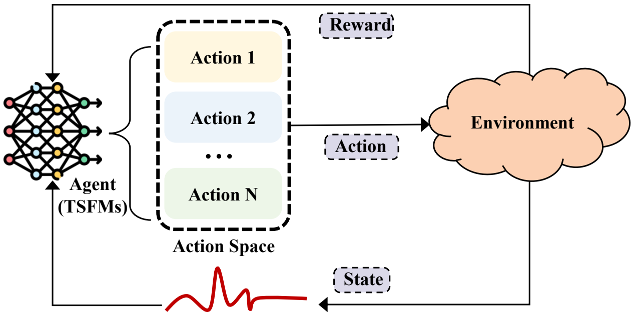


Fig. 13. Overview of reinforcement learning for time series modeling.

As shown in Fig. 13, reinforcement learning (RL) models learning as a closed-loop interaction between an agent and its environment [172]. The agent observes the current state, takes an action, receives a (possibly delayed) reward, and updates its policy (pre-trained TSFM) to maximize cumulative returns. This framework is well-suited for post-training TSFMs on preference-based objectives beyond conventional numerical losses. In practice, a pre-trained TSFM serves as the initial policy, while reinforcement objectives guide partial or full parameter updates through policy optimization. Reinforcement fine-tuning for TSFMs can be categorized into reasoning-driven and non-reasoning approaches. Reasoning-driven methods incorporate mechanisms such as chain-of-thought generation or multimodal alignment within the RL framework to enhance inference capabilities for time series modeling [173]. In contrast, non-reasoning methods directly optimize rewards based on task-specific or human-preference signals without explicit reasoning processes [169]. Both approaches highlight the integration of reinforcement optimization with post-training, advancing TSFMs from supervised fine-tuning toward intelligent, feedback-driven adaptation.

1) *Reasoning-Driven Fine-tuning*

Time series reasoning seeks to uncover causal structures in sequential data by capturing long-term dependencies, periodic patterns, and dynamic uncertainties [174], enabling human-like logical inference for complex tasks such as multi-step forecasting. In RL, an agent observes a state, selects an action, receives a reward, and updates its policy to maximize cumulative returns, forming a dynamic "perception–feedback–adaptation" loop [175]. The distinction lies in that reasoning emphasizes internal logical consistency, whereas RL can strengthen reasoning through external feedback in the loop. In TSFM post-training, reasoning drives decision generation, while RL can provide feedback signals to refine and stabilize the reasoning process. Moreover, explicitly structured reasoning steps can improve the interpretability of TSFMs by revealing the intermediate decision processes underlying the final predictions.

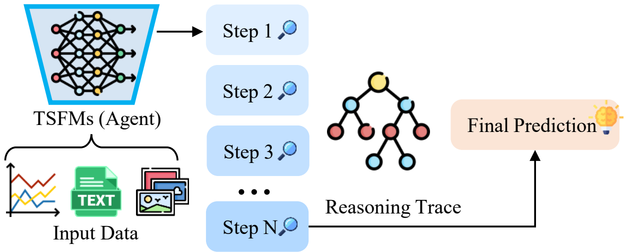


Fig. 14. Reasoning-driven reinforcement fine-tuning for TSFMs.

Reasoning-driven reinforcement post-training shifts optimization from numerical fitting to reasoning-oriented refinement by integrating the model's reasoning outputs with external feedback. As shown in Fig. 14, given a time series and corresponding textual or visual inputs, the model first generates reasoning-based outputs using LLMs [176], which are then assessed with domain-specific or human-aligned reward signals. These rewards guide fine-tuning through reinforcement learning policy optimization. Among such strategies, *Group Relative Policy Optimization* (GRPO) has emerged as a prominent approach [177]. GRPO introduces a relative advantage function over grouped samples to stabilize policy updates. For each input query q, G candidate outputs $\{o_i\}_{i=1}^{G}$ are sampled from the old policy, each assigned a scalar reward $\{r_i\}$. Rewards are normalized within the group to compute relative advantages $\hat{A}_{i,t}$, and the pre-trained TSFM policy parameters are updated as follows:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\big(r_{i,t}(\theta),\, 1-\epsilon,\, 1+\epsilon\big)\hat{A}_{i,t}\Big)\Bigg] - \beta\, D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}), \tag{9}$$

where the clipping function is defined as $\text{clip}(x, a, b) = \max(\min(x, b), a)$, $r_{i,t}(\theta)$ is the likelihood ratio between the new and reference policies, and D_{KL} denotes KL-divergence regularization. This mechanism aligns reasoning outputs with human or domain feedback, shifting optimization from "prediction accuracy" toward "reasoning rationality".

Reasoning-driven fine-tuning has achieved remarkable success in NLP [175], [177] and is now being extended to time series modeling. Recent work leverages the logical generation and multimodal reasoning capabilities of LLMs to improve TSFM performance on downstream tasks. For instance, Liu et al. [178] proposed a staged RL framework that employs dynamic reward mechanisms to capture temporal dependencies and overcome knowledge truncation, significantly improving forecasting accuracy. Luo et al. [173] integrated supervised fine-tuning with GRPO to jointly optimize pointwise errors (e.g., MSE, MAE) and structural consistency (e.g., trends, seasonality, extrema), achieving superior multi-step forecasting performance. Zhang et al. [176] developed a multimodal model, *TimeMaster*, which transforms time series into image-text representations and introduces a tri-output structure (reasoning, classification, and expansion). Using token-level GRPO optimization and LLM-based discriminators, it evaluates open-ended reasoning quality and structural accuracy. In financial scenarios, Xiao et al. [179] decoupled reasoning chains from trading recommendations and adopted volatility-normalized returns as rewards, leading to enhanced risk awareness and evidence-based decision-making. These studies collectively demonstrate the feasibility and growing potential of reasoning-driven RL in multimodal and interpretable time series modeling.

Summary: Reasoning-driven fine-tuning provides a promising framework for integrating human-aligned reasoning, domain-specific feedback, and reinforcement learning principles into TSFM post-training. While its current applications remain in their infancy compared with NLP domains, the paradigm offers significant potential to bridge reasoning and adaptive optimization. Continued advancements in reward modeling and multimodal feedback alignment are expected to further enhance the stability and generalization of reasoning-driven TSFMs. Also, subsequent approaches require deeper integration of irregular time series and frequency-domain periodic variations into reasoning modeling.

2) *Non-Reasoning Fine-tuning*

Non-reasoning post-training methods focus exclusively on aligning the final model outputs with external feedback signals, such as domain metrics or expert preferences, without explicitly supervising or rewarding intermediate reasoning trajectories. In contrast, reasoning-driven fine-tuning optimizes not only the final prediction but also the structure and quality of the reasoning process, encouraging longer and more explicit step-by-step inference. In practice, the non-reasoning approach directly enables TSFMs to align predictions with domain-specific operational goals. For instance, in industrial anomaly detection, rewards can be defined as weighted combinations of false alarm and miss rates [180], replacing traditional label-based supervision. In energy optimization or financial forecasting [181], task-specific indicators, such as yield, cost, or stability, can serve as reward functions, guiding post-training model optimization.

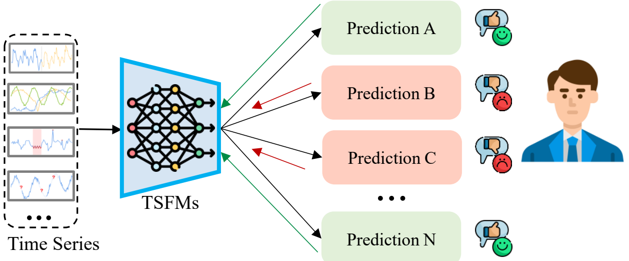


Fig. 15. Non-Reasoning fine-tuning for TSFMs based on RL.

As shown in Fig. 15, non-reasoning reinforcement post-training fine-tunes a pre-trained TSFM using external feedback instead of fixed supervision signals. The process typically involves three stages. First, external feedback is collected from the target domain, including measurable indicators (e.g., stability) and human preference scores (e.g., expert evaluations of prediction curves). Second, these signals are used to construct a reward or preference model that quantifies output quality. Finally, the pre-trained TSFM is treated as a policy model and optimized via reinforcement learning algorithms, most notably Proximal Policy Optimization (PPO) [170] or Direct Preference Optimization (DPO) [182]. PPO is a canonical and widely adopted method that relies on explicit reward modeling and on-policy optimization, making it broadly applicable across diverse post-training scenarios. By contrast, DPO [182] can be viewed as a simplified variant that directly leverages pairwise preference data to reduce training overhead.

Recent studies have advanced the non-reasoning fine-tuning paradigm in time series modeling to improve the adaptability of TSFMs. Qi et al. [169] proposed the *Time-series Policy Optimization (TPO)*, a reinforcement-learning-from-feedback approach designed for time series downstream tasks. TPO constructs contrastive preference pairs curated by domain experts to align TSFMs with task-specific objectives. Experimental results in [169] show that TPO yields consistent performance improvements from zero-shot to supervised fine-tuning. In particular, for long-tail product forecasting, TPO achieves an additional *+20.05* percentage point increase in online accuracy, demonstrating the practical advantage of reinforcement-style post-training in TSFMs. Similarly, Niu *et al.* [183] introduced *LangTime*, a language-guided unified TSFM that combines cross-domain pre-training with non-reasoning fine-tuning. LangTime incorporates a language model backbone enhanced with temporal encoders and domain-specific prompts, and optimizes multi-dimensional reward mechanisms to enable stable long-horizon autoregressive forecasting across diverse benchmarks. Overall, *LangTime* underscores the synergy between language-based representation learning and reinforcement-style optimization in advancing TSFMs.

Summary: Non-reasoning reinforcement fine-tuning elevates human and environmental feedback into an explicit optimization objective during the post-training stage of TSFMs. This paradigm enables TSFMs to align with performance-oriented metrics, such as cost efficiency and expert preferences, rather than relying solely on minimizing numerical loss functions. However, the practical adoption of this approach remains limited by the dependence on high-quality preference data and the need for specialized critic or reward models, both of which increase training cost and implementation complexity. Recent advances in lightweight optimization and policy distillation provide effective means to alleviate these constraints [182]. Moreover, non-reasoning reinforcement fine-tuning generally depends on reliable probabilistic decision-making, which many existing TSFMs do not adequately support [153]. Consequently, equipping TSFMs with stable probabilistic decision capabilities is essential for advancing research in non-reasoning reinforcement fine-tuning.

VII. Future Directions

A. Large-Scale Time Series Datasets

1) Sample Quality Evaluation

The effectiveness of TSFMs largely depends on the availability of large-scale high-quality source datasets for pre-training. Prior studies [55] show that increasing data size and diversity generally improves downstream performance. However, recent work [26] indicates that enlarging the source dataset does not always yield better results due to distributional mismatches between pre-training and target domains. When a significant portion of the source data is irrelevant or of low quality [184], the additional samples may introduce noise and hinder performance. Therefore, developing principled sample quality metrics that quantify the relevance of real and synthetic training samples between source and target domains is crucial. Such metrics can reduce training costs and improve the efficiency and effectiveness of TSFM pre-training in practical applications.

2) Domain-specific Dataset Construction

Large-scale source datasets for TSFM pre-training often combine data from multiple domains, making it challenging to identify domain-specific temporal patterns while distinguishing cross-domain relationships. Capturing these patterns enables more effective transfer of pre-trained knowledge to downstream tasks. Incorporating domain-specific subsequence characteristics, such as those in ECG [76] and human activity [133] during dataset construction can promote the learning of domain-relevant representations and improve transfer efficiency. Unlike words in NLP [67], time series subsequences lack explicit semantics, limiting the direct adoption of NLP pre-training strategies. Designing domain-specific datasets helps encode meaningful, discriminative subsequence representations, enhancing both cross-domain generalization and interpretability in critical areas such as healthcare and security.

B. Pre-training Techniques

1) Pre-training Architecture and Paradigm Design

Designing suitable neural architectures and pre-training paradigms is vital for advancing TSFMs, as they determine scalability and adaptability to downstream tasks. Studies have examined architectures for time series modeling, such as MLPs [185], CNNs [7], GNNs [43], Transformers [11], and Mamba [37]. MLPs are efficient and lightweight and Transformers enable scalable pre-training and transfer. Future research should explore hybrid and adaptive designs that balance efficiency, robustness, and domain adaptability. Pre-training paradigms, including encoder-only, decoder-only, and encoder–decoder frameworks, should align with task requirements [11], [22]. Encoder-only models suit discriminative tasks like classification [21], while decoder-only models favor generative tasks such as forecasting but require careful optimization [25]. Encoder–decoder structures, though complex, generally support multi-task settings. Equally important are pre-training objectives, including supervised, unsupervised, and self-supervised strategies [12], which influence generalization and transferability. Future progress depends on unified and scalable pre-training architectures and paradigms that flexibly adapt to diverse temporal structures and deployment needs.

C. Post-training Techniques

1) Incremental Post-training

Incremental learning, also known as continual or lifelong learning [186], has been widely applied in language and vision foundation models to enable efficient adaptation and reduce computational costs. In contrast, its application in TSFMs remains limited. Despite the rapid emergence of TSFMs, few have consistently outperformed traditional supervised methods across diverse domains, which constrains the exploration of continual learning in this field. Nonetheless, TSFMs hold strong potential as adaptable backbones for evolving time series tasks. Incremental fine-tuning across heterogeneous scenarios could enhance adaptability, mitigate catastrophic forgetting, and improve long-term model utility. Future research should develop TSFM-oriented incremental post-training strategies, including task-aware regularization, rehearsal mechanisms, and dynamic architecture adaptation, to strengthen robustness and scalability.

2) Agent-based Post-training

An agent is an autonomous system that perceives its environment, makes decisions from observations, and takes actions to achieve specific objectives [187]. Unlike conventional models that perform single-step input–output mapping, agents interact with dynamic environments, adapt to feedback, and integrate multiple decision-making components, extending capabilities beyond prediction or classification. Inspired by LLM-based agents, some studies [187] have applied agent frameworks to time series modeling. Therefore, agents can serve as perception and reasoning modules in TSFM post-training for future work, enabling integration of multi-task and multi-modal capabilities, robustness in dynamic environments, and efficient adaptation to small-scale or evolving domains.

VIII. Conclusion

This survey provides a comprehensive review of TSFMs trained from scratch, spanning pre-training to post-training. We examine modeling strategies for both regularly and irregularly sampled time series, emphasizing key techniques in temporal dependencies, frequency modeling, variable relationships, and shapelet-based features. Our analysis indicates that current research primarily focuses on pre-training paradigms and architectural design, while systematic studies on dataset construction and post-training remain limited. High-quality source datasets are crucial for enhancing TSFM generalization and robustness, whereas appropriate downstream benchmarks, baselines, and evaluation metrics are essential for reliable performance assessment. Furthermore, post-training is critical for integrating pre-trained knowledge with domain-specific representations, thereby improving transferability of TSFMs.

References

[1] M. Middlehurst, P. Schäfer, and A. Bagnall, “Bake off redux: a review and experimental evaluation of recent time series classification algorithms,” *Data Mining and Knowledge Discovery*, vol. 38, no. 4, pp. 1958–2031, 2024.

[2] K. Yi, Q. Zhang, W. Fan, L. Cao, S. Wang, H. He, G. Long, L. Hu, Q. Wen, and H. Xiong, “A survey on deep learning based time series analysis with frequency transformation,” in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 6206–6215.

[3] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian, “Accurate medium-range global weather forecasting with 3d neural networks,” *Nature*, vol. 619, no. 7970, pp. 533–538, 2023.

[4] S. Zhang, Y. Du, W. Wang, X. He, F. Cui, L. Zhao, B. Wang, Z. Hu, Z. Wang, Q. Xia *et al.*, “Ecgfm: A foundation model for ecg analysis trained on a multi-center million-ecg dataset,” *Information Fusion*, p. 103363, 2025.

[5] R. Liu, G. Meng, B. Yang, C. Sun, and X. Chen, “Dislocated time series convolutional neural architecture: An intelligent fault diagnosis approach for electric machine,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1310–1320, 2016.

[6] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, “Informer: Beyond efficient transformer for long sequence time-series forecasting,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[7] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, “Inceptiontime: Finding alexnet for time series classification,” *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[8] X. Qiu, Z. Li, W. Qiu, S. Hu, L. Zhou, X. Wu, Z. Li, C. Guo, A. Zhou, Z. Sheng *et al.*, “Tab: Unified benchmarking of time series anomaly detection methods,” *arXiv preprint arXiv:2506.18046*, 2025.

[9] W. Cao, D. Wang, J. Li, H. Zhou, Y. Li, and L. Li, “Brits: bidirectional recurrent imputation for time series,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 6776–6786.

[10] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin, “One Fits All: Power General Time Series Analysis by Pretrained LM,” in *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

[11] M. Goswami, K. Szafer, A. Choudhry, Y. Cai, S. Li, and A. Dubrawski, “Moment: a family of open time-series foundation models,” in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 16 115–16 152.

[12] Q. Ma, Z. Liu, Z. Zheng, Z. Huang, S. Zhu, Z. Yu, and J. T. Kwok, “A survey on time-series pre-trained models,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 7536–7555, 2024.

[13] M. Jin, Q. Wen, Y. Liang, C. Zhang, S. Xue, X. Wang, J. Zhang, Y. Wang, H. Chen, X. Li *et al.*, "Large models for time series and spatio-temporal data: A survey and outlook," *arXiv preprint arXiv:2310.10196*, 2023.

[14] J. Chen, J. E. Lenssen, A. Feng, W. Hu, M. Fey, L. Tassiulas, J. Leskovec, and R. Ying, "From similarity to superiority: Channel clustering for time series forecasting," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 130 635–130 663.

[15] Y. Luo, B. Zhang, Z. Liu, and Q. Ma, "Hi-patch: Hierarchical patch gnn for irregular multivariate time series," in *Forty-second International Conference on Machine Learning*, 2025.

[16] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, "One fits all: Power general time series analysis by pretrained lm," in *Advances in neural information processing systems*, vol. 36, 2023, pp. 43 322–43 355.

[17] Z. Li, S. Li, and X. Yan, "Time series as images: Vision transformer for irregularly sampled time series," *Advances in Neural Information Processing Systems*, vol. 36, pp. 49 187–49 204, 2023.

[18] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time Series Forecasting by Reprogramming Large Language Models," in *The Twelfth International Conference on Learning Representations*, 2023.

[19] M. Chen, L. Shen, Z. Li, X. J. Wang, J. Sun, and C. Liu, "Visionts: Visual masked autoencoders are free-lunch zero-shot time series forecasters," in *Forty-second International Conference on Machine Learning*, 2025.

[20] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations*, 2023.

[21] C. Lin, X. Wen, W. Cao, C. Huang, J. Bian, S. Lin, and Z. Wu, "Nutime: Numerically multi-scaled embedding for large- scale time-series pretraining," *Transactions on Machine Learning Research*, 2024. [Online]. Available: https://openreview.net/forum?id=TwiSBZ0p9u

[22] A. F. Ansari, L. Stella, A. C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor *et al.*, "Chronos: Learning the language of time series," *Transactions on Machine Learning Research*, 2024.

[23] Y. Liu, G. Qin, Z. Shi, Z. Chen, C. Yang, X. Huang, J. Wang, and M. Long, "Sundial: A Family of Highly Capable Time Series Foundation Models," in *Forty-Second International Conference on Machine Learning*, 2025.

[24] G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo, "Unified training of universal time series forecasting transformers," in *International Conference on Machine Learning*. PMLR, 2024, pp. 53 140–53 164.

[25] Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long, "Timer: generative pre-trained transformers are large time series models," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 32 369–32 399.

[26] V. Feofanov, S. Wen, M. Alonso, R. Ilbert, H. Guo, M. Tiomoko, L. Pan, J. Zhang, and I. Redko, "Mantis: Lightweight calibrated foundation model for user-friendly time series classification," *arXiv preprint arXiv:2502.15637*, 2025.

[27] L. Luo, W. Gui, Y. Liu, Z. Zhang, Y. Zhang, F. Wang, Z. Guo, Z. Ma, X. Liu, H. He *et al.*, "Emind: A foundation model for multi-task electromagnetic signals understanding," *arXiv preprint arXiv:2508.18785*, 2025.

[28] H. Li, B. Deng, C. Xu, Z. Feng, V. Schlegel, Y.-H. Huang, Y. Sun, J. Sun, K. Yang, Y. Yu *et al.*, "Mira: Medical time series foundation model for real-world health data," in *Advances in neural information processing systems*, 2025.

[29] X. Shi, S. Wang, Y. Nie, D. Li, Z. Ye, Q. Wen, and M. Jin, "Time-moe: Billion-scale time series foundation models with mixture of experts," in *The Thirteenth International Conference on Learning Representations*, 2025.

[30] X. Liu, J. Liu, G. Woo, T. Aksu, Y. Liang, R. Zimmermann, C. Liu, J. Li, S. Savarese, C. Xiong, and D. Sahoo, "Moirai-MoE: Empowering Time Series Foundation Models with Sparse Mixture of Experts," in *Forty-Second International Conference on Machine Learning*, 2025.

[31] J. A. Miller, M. Aldosari, F. Saeed, N. H. Barna, S. Rana, I. B. Arpinar, and N. Liu, "A survey of deep learning and foundation models for time series forecasting," *arXiv preprint arXiv:2401.13912*, 2024.

[32] Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, and Q. Wen, "Foundation models for time series analysis: A tutorial and survey," in *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2024, pp. 6555–6565.

[33] J. Ye, W. Zhang, K. Yi, Y. Yu, Z. Li, J. Li, and F. Tsung, "A survey of time series foundation models: Generalizing time series representation with large language model," *arXiv preprint arXiv:2405.02358*, 2024.

[34] S. R. K. Kottapalli, K. Hubli, S. Chandrashekhara, G. Jain, S. Hubli, G. Botla, and R. Doddaiah, "Foundation models for time series: A survey," *arXiv preprint arXiv:2504.04011*, 2025.

[35] X. Liu, T. Aksu, J. Liu, Q. Wen, Y. Liang, C. Xiong, S. Savarese, D. Sahoo, J. Li, and C. Liu, "Empowering time series analysis with synthetic data: A survey and outlook in the era of foundation models," *arXiv preprint arXiv:2503.11411*, 2025.

[36] Y. Sun, Z. Xie, D. Chen, E. Eldele, and Q. Hu, "Hierarchical classification auxiliary network for time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 19, 2025, pp. 20 743–20 751.

[37] Y. Wu, X. Meng, H. Hu, J. Zhang, Y. Dong, and D. Lu, "Affirm: Interactive mamba with adaptive fourier filters for long-term time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 20, 2025, pp. 21 599–21 607.

[38] Z. Qiao, C. Liu, Y. Zhang, M. Jin, Q. Pham, Q. Wen, P. Suganthan, X. Jiang, and S. Ramasamy, "Multi-scale finetuning for encoder-based time series foundation models," in *Advances in Neural Information Processing Systems*, 2025.

[39] Z. Liu, Q. Ma, P. Ma, and L. Wang, "Temporal-frequency co-training for time series semi-supervised learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 7, 2023, pp. 8923–8931.

[40] R. Cheng, X. Jia, Q. Li, R. Xing, J. Huang, Y. Zheng, and Z. Xie, "FAT: Frequency-Aware Pretraining for Enhanced Time-Series Representation Learning," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, ser. KDD '25. Association for Computing Machinery, 2025, pp. 310–321.

[41] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," in *International Conference on Learning Representations*, 2022.

[42] E. Fons, A. Sztrajman, Y. El-Laham, L. Ferrer, S. Vyetrenko, and M. Veloso, "LSCD: Lomb–Scargle Conditioned Diffusion for Time series Imputation," in *Forty-Second International Conference on Machine Learning*, 2025.

[43] Y. Wang, Y. Xu, J. Yang, M. Wu, X. Li, L. Xie, and Z. Chen, "Fully-connected spatial-temporal graph for multivariate time-series data," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 14, 2024, pp. 15 715–15 724.

[44] L. Ye and E. Keogh, "Time series shapelets: a novel technique that allows accurate, interpretable and fast classification," *Data mining and knowledge discovery*, vol. 22, no. 1, pp. 149–182, 2011.

[45] N. Zhang and S. Sun, "Multiview unsupervised shapelet learning for multivariate time series clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4981–4996, 2022.

[46] Z. Liu, Y. Luo, B. Li, E. Eldele, M. Wu, and Q. Ma, "Learning soft sparse shapes for efficient time-series classification," in *Forty-second International Conference on Machine Learning*, 2025.

[47] J. Wang, W. Du, Y. Yang, L. Qian, W. Cao, K. Zhang, W. Wang, Y. Liang, and Q. Wen, "Deep learning for multivariate time series imputation: A survey," *arXiv preprint arXiv:2402.04059*, 2024.

[48] C. W. Tan, C. Bergmeir, F. Petitjean, and G. I. Webb, "Time series extrinsic regression: Predicting numeric values from time series data," *Data Mining and Knowledge Discovery*, vol. 35, no. 3, pp. 1032–1060, 2021.

[49] S. Aminikhanghahi and D. J. Cook, "A survey of methods for time series change point detection," *Knowledge and information systems*, vol. 51, no. 2, pp. 339–367, 2017.

[50] M. Perslev, M. H. Jensen, S. Darkner, P. J. Jennum, and C. Igel, "U-time: a fully convolutional network for time series segmentation applied to sleep staging," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 4415–4426.

[51] D. Song, N. Xia, W. Cheng, H. Chen, and D. Tao, "Deep r-th root of rank supervised joint binary embedding for multivariate time series retrieval," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2229–2238.

[52] C. Chang, J. Hwang, Y. Shi, H. Wang, W.-C. Peng, T.-F. Chen, and W. Wang, "Time-IMM: A Dataset and Benchmark for Irregular Multimodal Multivariate Time Series," *arXiv preprint*, 2025.

[53] Z. Liu, D. Chen, W. Pei, Q. Ma *et al.*, "Scale-teaching: Robust multi-scale training for time series classification with noisy labels," *Advances*

in Neural Information Processing Systems, vol. 36, pp. 33 726–33 757, 2023.

[54] R. Wu and E. J. Keogh, "Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress," *IEEE transactions on knowledge and data engineering*, vol. 35, no. 3, pp. 2421–2429, 2021.

[55] Q. Yao, C.-H. H. Yang, R. Jiang, Y. Liang, M. Jin, and S. Pan, "Towards neural scaling laws for time series foundation models," in *The Thirteenth International Conference on Learning Representations*, 2025.

[56] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[57] L. Shen and J. Kwok, "Non-autoregressive conditional diffusion models for time series prediction," in *International Conference on Machine Learning*. PMLR, 2023, pp. 31 016–31 029.

[58] S. Xie, V. Feofanov, M. Alonso, A. Odonnat, J. Zhang, T. Palpanas, and I. Redko, "Cauker: classification time series foundation models can be pretrained on synthetic data only," *arXiv preprint arXiv:2508.02879*, 2025.

[59] E. O. Taga, M. E. Ildiz, and S. Oymak, "Timepfn: Effective multivariate time series forecasting with synthetic data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 19, 2025, pp. 20 761–20 769.

[60] Y. Rotem, N. Shimoni, L. Rokach, and B. Shapira, "Transfer learning for time series classification using synthetic data generation," in *International Symposium on Cyber Security, Cryptology, and Machine Learning*. Springer, 2022, pp. 232–246.

[61] S. Dooley, G. S. Khurana, C. Mohapatra, S. V. Naidu, and C. White, "Forecastpfn: Synthetically-trained zero-shot forecasting," *Advances in Neural Information Processing Systems*, vol. 36, pp. 2403–2426, 2023.

[62] P. Emami, A. Sahu, and P. Graf, "Buildingsbench: A large-scale dataset of 900k buildings and benchmark for short-term load forecasting," *Advances in Neural Information Processing Systems*, vol. 36, pp. 19 823–19 857, 2023.

[63] W. Wang, K. Wu, Y. B. Li, D. Wang, X. Zhang, and J. Liu, "Mitigating data scarcity in time series analysis: A foundation model with series-symbol data generation," *arXiv preprint arXiv:2502.15466*, 2025.

[64] T. Lan, H. D. Le, J. Li, W. He, M. Wang, C. Liu, and C. Zhang, "Towards foundation models for zero-shot time series anomaly detection: Leveraging synthetic data and relative context discrepancy," *arXiv preprint arXiv:2509.21190*, 2025.

[65] J. ALSING, T. Edwards, B. D. Wandelt, J. Alvey, and N. H. Nguyen, "Scaling-laws for large time-series models," in *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024.

[66] C. Rousseau, T. Boschi, G. Cornacchia, D. Salwala, A. Pascale, and J. B. Moreno, "Forging time series with language: A large language model approach to synthetic data generation," in *Advances in neural information processing systems*, 2025.

[67] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," in *International Conference on Learning Representations*, 2017.

[68] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[69] S. Gao, T. Koker, O. Queen, T. Hartvigsen, T. Tsiligkaridis, and M. Zitnik, "Units: A unified multi-task time series model," *Advances in Neural Information Processing Systems*, vol. 37, pp. 140 589–140 631, 2024.

[70] A. Das, W. Kong, R. Sen, and Y. Zhou, "A decoder-only foundation model for time-series forecasting," in *International Conference on Machine Learning*. PMLR, 2024, pp. 10 148–10 167.

[71] K. Rasul, A. Ashok, A. R. Williams, A. Khorasani, G. Adamopoulos, R. Bhagwatkar, M. Biloš, H. Ghonia, N. Hassen, A. Schneider *et al.*, "Lag-llama: Towards foundation models for time series forecasting," in *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

[72] H. Zhang, Y. Liu, Y. Qiu, H. Liu, Z. Pei, J. Wang, and M. Long, "Timesbert: A bert-style foundation model for time series understanding," *arXiv preprint arXiv:2502.21245*, 2025.

[73] A. Dempster, N. M. Foumani, C. W. Tan, L. Miller, A. Mishra, M. Salehi, C. Pelletier, D. F. Schmidt, and G. I. Webb, "Monster: Monash scalable time series evaluation repository," *arXiv preprint arXiv:2502.15122*, 2025.

[74] G. Woo, C. Liu, A. Kumar, and D. Sahoo, "Pushing the limits of pre-training for time series forecasting in the cloudops domain," *arXiv preprint arXiv:2310.05063*, 2023.

[75] B. Gow, T. Pollard, L. A. Nathanson, A. Johnson, B. Moody, C. Fernandes, N. Greenbaum, J. W. Waks, P. Eslami, T. Carbonati *et al.*, "Mimic-iv-ecg: Diagnostic electrocardiogram matched subset," *Type: dataset*, vol. 6, pp. 13–14, 2023.

[76] K. McKeen, S. Masood, A. Toma, B. Rubin, and B. Wang, "Ecg-fm: An open electrocardiogram foundation model," *arXiv preprint arXiv:2408.05178*, 2024.

[77] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *The Eleventh International Conference on Learning Representations*, 2023.

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

[79] I. Silva, G. Moody, D. J. Scott, L. A. Celi, and R. G. Mark, "Predicting In-Hospital Mortality of ICU Patients: The PhysioNet/Computing in Cardiology Challenge 2012," *Computing in cardiology*, vol. 39, pp. 245–248, 2012.

[80] V. Vidulin, M. Lustrek, B. Kaluza, R. Piltaver, and J. Krivec, "Localization data for person activity," UCI Machine Learning Repository, 2010.

[81] M. Menne, C. Williams, Jr., and R. Vose, "Long-term daily and monthly climate records from stations across the contiguous united states (u.s. historical climatology network)," 1 2016.

[82] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh, "The ucr time series archive," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, pp. 1293–1305, 2019.

[83] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The uea multivariate time series classification archive, 2018," *arXiv preprint arXiv:1811.00075*, 2018.

[84] C. W. Tan, A. Dempster, C. Bergmeir, and G. I. Webb, "Multirocket: multiple pooling operators and transformations for fast and effective time series classification," *Data Mining and Knowledge Discovery*, vol. 36, no. 5, pp. 1623–1646, 2022.

[85] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, no. 1, p. 160035, 2016.

[86] M. A. Reyna, C. S. Josef, R. Jeter, S. P. Shashikumar, M. B. Westover, S. Nemati, G. D. Clifford, and A. Sharma, "Early Prediction of Sepsis From Clinical Data: The PhysioNet/Computing in Cardiology Challenge 2019," *Critical Care Medicine*, vol. 48, no. 2, p. 210, 2020.

[87] A. Reiss and D. Stricker, "Introducing a New Benchmarked Dataset for Activity Monitoring," in *2012 16th International Symposium on Wearable Computers*, 2012, pp. 108–109.

[88] K.-H. Lai, D. Zha, J. Xu, Y. Zhao, G. Wang, and X. Hu, "Revisiting time series outlier detection: Definitions and benchmarks," in *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 1)*.

[89] S. Zhang, B. Guo, A. Dong, J. He, Z. Xu, and S. X. Chen, "Cautionary tales on air-quality improvement in beijing," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2205, p. 20170457, 2017.

[90] W. Tang, L. Liu, and G. Long, "Interpretable time-series classification on few-shot samples," in *2020 International joint conference on neural networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[91] A. De Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, 2016.

[92] M. Yu, X. Guo, Z. Li, Y. Shu *et al.*, "Towards measuring and modeling geometric structures in time series forecasting via image modality," *arXiv preprint arXiv:2507.23253*, 2025.

[93] D. Kudrat, Z. Xie, Y. Sun, T. Jia, and Q. Hu, "Patch-wise structural loss for time series forecasting," *arXiv preprint arXiv:2503.00877*, 2025.

[94] W. Ye, Z. Xu, and N. Gui, "Non-stationary diffusion for probabilistic time series forecasting," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=afpc1MFMYU

[95] Z. Li, X. Qiu, P. Chen, Y. Wang, H. Cheng, Y. Shu, J. Hu, C. Guo, A. Zhou, C. S. Jensen *et al.*, "Tsfm-bench: A comprehensive and unified benchmark of foundation models for time series forecasting," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 5595–5606.

[96] X. Qiu, J. Hu, L. Zhou, X. Wu, J. Du, B. Zhang, C. Guo, A. Zhou, C. S. Jensen, Z. Sheng *et al.*, "Tfb: Towards comprehensive and fair

benchmarking of time series forecasting methods," *Proceedings of the VLDB Endowment*, vol. 17, no. 9, pp. 2363–2377, 2024.

[97] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[98] S. J. Talukder, Y. Yue, and G. Gkioxari, "Totem: Tokenized time series embeddings for general time series analysis," *Transactions on Machine Learning Research*, 2024.

[99] J. Early, G. Cheung, K. Cutajar, H. Xie, J. Kandola, and N. Twomey, "Inherently interpretable time series classification via multiple instance learning," in *The Twelfth International Conference on Learning Representations*, 2024.

[100] V. Ekambaram, S. Kumar, A. Jati, S. Mukherjee, T. Sakai, P. Dayama, W. M. Gifford, and J. Kalagnanam, "Tspulse: Dual space tiny pre-trained models for rapid time-series analysis," *arXiv preprint arXiv:2505.13033*, 2025.

[101] S. Kim, K. Choi, H.-S. Choi, B. Lee, and S. Yoon, "Towards a rigorous evaluation of time-series anomaly detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 7, 2022, pp. 7194–7201.

[102] A. Huet, J. M. Navarro, and D. Rossi, "Local evaluation of time series anomaly detection algorithms," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 635–645.

[103] J. Paparrizos, P. Boniol, T. Palpanas, R. S. Tsay, A. Elmore, and M. J. Franklin, "Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection," *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2774–2787, 2022.

[104] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: poster and demo track*, vol. 1, pp. 59–63, 2012.

[105] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Advances in neural information processing systems*, vol. 12, 1999.

[106] W. Du, J. Wang, L. Qian, Y. Yang, Z. Ibrahim, F. Liu, Z. Wang, H. Liu, Z. Zhao, Y. Zhou *et al.*, "Tsi-bench: Benchmarking time series imputation," *arXiv preprint arXiv:2406.12747*, 2024.

[107] X. Chen, X. Li, B. Liu, and Z. Li, "Biased temporal convolution graph network for time series forecasting with missing values," in *The Twelfth International Conference on Learning Representations*, 2023.

[108] S. Chen, G. Long, J. Jiang, and C. Zhang, "Federated foundation models on heterogeneous time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 15, 2025, pp. 15 839–15 847.

[109] R. Lunelli, A. Nicolson, S. M. Pröll, S. J. Reinstadler, A. Bauer, and C. Dlaska, "Benchecg and xecg: a benchmark and baseline for ecg foundation models," *arXiv preprint arXiv:2509.10151*, 2025.

[110] T. Yang, P. Zhang, M. Zheng, Y. Shi, L. Jing, J. Huang, and N. Li, "Wirelessgpt: A generative pre-trained multi-task learning framework for wireless communication," *IEEE Network*, vol. 39, no. 5, pp. 58–65, 2025.

[111] M. P. Hung, A. Saeed, and D. Ma, "Boosting masked ecg-text auto-encoders as discriminative learners," in *Forty-second International Conference on Machine Learning*, 2025.

[112] J. Pu, S. Zhao, L. Cheng, Y. Chang, R. Wu, T. Lv, and R. Zhang, "Examining the effect of pre-training on time series classification," *arXiv preprint arXiv:2309.05256*, 2023.

[113] C. Shyalika, H. K. Bagga, A. Bhatt, R. Prasad, A. A. Ghazo, and A. Sheth, "Time series foundational models: Their role in anomaly detection and prediction," *arXiv preprint arXiv:2412.19286*, 2024.

[114] B. Hela, P. P. Handigol, and P. Arjunan, "Are time series foundation models good for energy anomaly detection?" in *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*, 2025, pp. 656–665.

[115] M. Wang, T. Ma, and S. B. Cohen, "Pre-training time series models with stock data customization," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 3019–3030.

[116] Y. Liu, G. Qin, X. Huang, J. Wang, and M. Long, "Timer-XL: Long-Context Transformers for Unified Time Series Forecasting," in *The Thirteenth International Conference on Learning Representations*, 2024.

[117] N. Gruver, M. A. Finzi, S. Qiu, and A. G. Wilson, "Large Language Models Are Zero-Shot Time Series Forecasters," in *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

[118] Z. Liu, J. Yang, M. Cheng, Y. Luo, and Z. Li, "Generative Pretrained Hierarchical Transformer for Time Series Forecasting," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '24. Association for Computing Machinery, 2024, pp. 2003–2013.

[119] H. Zhang, C. Xu, Y.-F. Zhang, Z. Zhang, L. Wang, and J. Bian, "Timeraf: Retrieval-augmented foundation model for zero-shot time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[120] R. R. Chowdhury, J. Li, X. Zhang, D. Hong, R. K. Gupta, and J. Shang, "PrimeNet: Pre-training for Irregular Multivariate Time Series," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 6, pp. 7184–7192, 2023.

[121] J. Xiao, Y. Chen, G. Cong, W. Nejdl, and S. Gottschalk, "FlexTSF: A Universal Forecasting Model for Time Series with Variable Regularities," *arXiv preprint*, 2024.

[122] H. Shi, S. Du, Y. Yang, J. Zhang, T. Li, and Y. Zheng, "A knowledge-guided pre-training temporal data analysis foundation model for urban computing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, pp. 6259–6271, 2025.

[123] Y. Zhang, M. Liu, S. Zhou, and J. Yan, "UP2ME: Univariate Pre-training to Multivariate Fine-tuning as a General-purpose Framework for Multivariate Time Series Analysis," in *International Conference on Machine Learning*. PMLR, 2024, pp. 59 358–59 381.

[124] A. F. Ansari, O. Shchur, J. Küken, A. Auer, B. Han, P. Mercado, S. S. Rangapuram, H. Shen, L. Stella, X. Zhang, M. Goswami, S. Kapoor, D. C. Maddix, P. Guerron, T. Hu, J. Yin, N. Erickson, P. M. Desai, H. Wang, H. Rangwala, G. Karypis, Y. Wang, and M. Bohlke-Schneider, "Chronos-2: From Univariate to Universal Forecasting," *arXiv preprint*, 2025.

[125] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[126] C.-C. M. Yeh, X. Dai, H. Chen, Y. Zheng, Y. Fan, A. Der, V. Lai, Z. Zhuang, J. Wang, L. Wang *et al.*, "Toward a foundation model for time series data," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 4400–4404.

[127] P. Malhotra, V. TV, L. Vig, P. Agarwal, and G. Shroff, "Timenet: Pre-trained deep recurrent neural network for time series classification," *arXiv preprint arXiv:1706.08838*, 2017.

[128] K. Kashiparekh, J. Narwariya, P. Malhotra, L. Vig, and G. Shroff, "Convtimenet: A pre-trained deep convolutional neural network for time series classification," in *2019 international joint conference on neural networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[129] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Transfer learning for time series classification," in *2018 IEEE international conference on big data (Big Data)*. IEEE, 2018, pp. 1367–1376.

[130] A. Ismail-Fawaz, M. Devanne, S. Berretti, J. Weber, and G. Forestier, "Finding foundation models for time series classification with a pretext task," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2024, pp. 123–135.

[131] Z. Liu, Y. Wang, B. Li, J. Zheng, E. Eldele, M. Wu, and Q. Ma, "A unified shape-aware foundation model for time series classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

[132] N. Huang, H. Wang, Z. He, M. Zitnik, and X. Zhang, "Repurposing foundation model for generalizable medical time series classification," *arXiv preprint arXiv:2410.03794*, 2024.

[133] X. Zhang, D. Teng, R. R. Chowdhury, S. Li, D. Hong, R. Gupta, and J. Shang, "Unimts: Unified pre-training for motion time series," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 107 469–107 493.

[134] L. Bickmann, L. Plagwitz, A. Büscher, L. Eckardt, and J. Varghese, "Exchangeai: An end-to-end platform and efficient foundation model for electrocardiogram analysis and fine-tuning," *arXiv preprint arXiv:2503.13570*, 2025.

[135] Y. Jeong, E. Yang, J. H. Ryu, I. Park, and M. Kang, "Anomalybert: Self-supervised transformer for time series anomaly detection using data degradation scheme," *arXiv preprint arXiv:2305.04468*, 2023.

[136] X. Wang, Q. Xu, K. Xu, T. Yu, B. Ding, D. Feng, and Y. Dou, "Large pretrained foundation model for key performance indicator multivariate time series anomaly detection," *IEEE Open Journal of the Computer Society*, 2024.

[137] P. Yan, A. Abdulkadir, P.-P. Luley, M. Rosenthal, G. A. Schatte, B. F. Grewe, and T. Stadelmann, "A comprehensive survey of deep transfer learning for anomaly detection in industrial time series: Methods,

applications, and directions," *IEEE Access*, vol. 12, pp. 3768–3789, 2024.

[138] A. Garza, C. Challu, and M. Mergenthaler-Canseco, "Timegpt-1," *arXiv preprint arXiv:2310.03589*, 2023.

[139] C. Maru and S. Sato, "Ratfm: Retrieval-augmented time series foundation model for anomaly detection," *arXiv preprint arXiv:2506.02081*, 2025.

[140] G. G. González, P. Casas, E. Martínez, and A. Fernández, "Towards foundation auto-encoders for time-series anomaly detection," *arXiv preprint arXiv:2507.01875*, 2025.

[141] Q. Shentu, B. Li, K. Zhao, Y. Shu, Z. Rao, L. Pan, B. Yang, and C. Guo, "Towards a general time series anomaly detector with adaptive bottlenecks and dual adversarial decoders," in *The Thirteenth International Conference on Learning Representations*, 2025.

[142] S. Malladi, T. Gao, E. Nichani, A. Damian, J. D. Lee, D. Chen, and S. Arora, "Fine-tuning language models with just forward passes," *Advances in Neural Information Processing Systems*, vol. 36, pp. 53 038–53 075, 2023.

[143] K. Ning, Z. Pan, Y. Liu, Y. Jiang, J. Y. Zhang, K. Rasul, A. Schneider, L. Ma, Y. Nevmyvaka, and D. Song, "Ts-rag: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster," in *Advances in Neural Information Processing Systems*, 2025.

[144] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[145] T. Nie, Y. Mei, G. Qin, J. Sun, and W. Ma, "Channel-aware low-rank adaptation in time series forecasting," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 3959–3963.

[146] D. Gupta, A. Bhatti, S. Parmar, C. Dan, Y. Liu, B. Shen, and S. Lee, "Low-rank adaptation of time series foundational models for out-of-domain modality forecasting," in *Proceedings of the 26th International Conference on Multimodal Interaction*, 2024, pp. 382–386.

[147] D. Wu, Y. Shen, and H. Jin, "Uncertainty-aware fine-tuning on time series foundation model for anomaly detection," 2024. [Online]. Available: https://openreview.net/forum?id=W1wlE4bPqP

[148] L. Pan, Z. Chen, H. Li, G. Liu, Z. Xu, Z. Liu, H. Wang, and Y. Wei, "Mixture of low rank adaptation with partial parameter sharing for time series forecasting," *arXiv preprint arXiv:2505.17872*, 2025.

[149] R. Zhou, Y. Zhang, and Y. Dong, "H-tuning: Toward low-cost and efficient ecg-based cardiovascular disease detection with pre-trained models," in *Forty-second International Conference on Machine Learning*, 2025.

[150] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*. PMLR, 2019, pp. 2790–2799.

[151] P. Niu, T. Zhou, X. Wang, L. Sun, and R. Jin, "Understanding the role of textual prompts in llm for time series forecasting: an adapter view," *arXiv e-prints*, pp. arXiv–2311, 2023.

[152] R. Ilbert, V. Feofanov, M. Tiomoko, I. Redko, and T. Palpanas, "User-friendly foundation model adapters for multivariate time series classification," in *2025 IEEE 41st International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 2025, pp. 136–144.

[153] A. Benechehab, V. Feofanov, G. Paolo, A. Thomas, M. Filippone, and B. Kégl, "Adapts: Adapting univariate foundation models to probabilistic multivariate time series forecasting," in *Forty-second International Conference on Machine Learning*, 2025.

[154] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[155] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.

[156] Z. Liu, W. Pei, D. Lan, and Q. Ma, "Diffusion language-shapelets for semi-supervised time-series classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 38, no. 13, 2024, pp. 14 079–14 087.

[157] Y. Chen, S. Huang, Y. Cheng, P. Chen, Z. Rao, Y. Shu, B. Yang, L. Pan, and C. Guo, "Aimts: Augmented series and image contrastive learning for time series classification," in *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE Computer Society, 2025, pp. 1952–1965.

[158] C. Liu, Q. Xu, H. Miao, S. Yang, L. Zhang, C. Long, Z. Li, and R. Zhao, "Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 18, 2025, pp. 18 780–18 788.

[159] R. King, T. Yang, and B. J. Mortazavi, "Multimodal pretraining of medical time series and notes," in *Machine Learning for Health (ML4H)*. PMLR, 2023, pp. 244–255.

[160] C. Wang, Q. Qi, J. Wang, H. Sun, Z. Zhuang, J. Wu, L. Zhang, and J. Liao, "Chattime: A unified multimodal time series foundation model bridging numerical and textual data," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 12, 2025, pp. 12 694–12 702.

[161] J. Wu, "Anomaly detection in medical via multimodal foundation models," *Frontiers in Bioengineering and Biotechnology*, vol. 13, p. 1644697, 2025.

[162] P. Liu, H. Guo, T. Dai, N. Li, J. Bao, X. Ren, Y. Jiang, and S.-T. Xia, "Calf: Aligning llms for time series forecasting via cross-modal fine-tuning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 18, 2025, pp. 18 915–18 923.

[163] J. Chen, Z. Zhao, G. Nurbek, A. Feng, A. Maatouk, L. Tassiulas, Y. Gao, and R. Ying, "Trace: Grounding time series in context for multimodal embedding and retrieval," in *Advances in Neural Information Processing Systems*, 2025.

[164] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International journal of computer vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[165] H. Xing, Z. Xiao, R. Qu, Z. Zhu, and B. Zhao, "An efficient federated distillation learning system for multitask time series classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–12, 2022.

[166] X. Yang, Q. Li, C. Zhang, and P. C. Woodland, "Knowledge distillation from multiple foundation models for end-to-end speech recognition," *arXiv preprint arXiv:2303.10917*, 2023.

[167] S. Abbaspourazad, A. Mishra, J. Futoma, A. C. Miller, and I. Shapiro, "Wearable accelerometer foundation models for health via knowledge distillation," *arXiv preprint arXiv:2412.11276*, 2024.

[168] J. Chan, Z. Chen, and E. Pan, "Foundation models knowledge distillation for battery capacity degradation forecast," *arXiv preprint arXiv:2505.08151*, 2025.

[169] Y. Qi, H. Hu, D. Lei, J. Zhang, Z. Shi, Y. Huang, Z. Chen, X. Lin, and Z.-J. M. Shen, "Timehf: Billion-scale time series models guided by human feedback," *arXiv preprint arXiv:2501.15942*, 2025.

[170] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize with human feedback," in *Advances in neural information processing systems*, vol. 33, 2020, pp. 3008–3021.

[171] H. Lai, X. Liu, J. Gao, J. Cheng, Z. Qi, Y. Xu, S. Yao, D. Zhang, J. Du, Z. Hou *et al.*, "A survey of post-training scaling in large language models," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 2771–2791.

[172] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[173] Y. Luo, Y. Zhou, M. Cheng, J. Wang, D. Wang, T. Pan, and J. Zhang, "Time series forecasting as reasoning: A slow-thinking approach with reinforced llms," *arXiv preprint arXiv:2506.10630*, 2025.

[174] C. Chang, Y. Shi, D. Cao, W. Yang, J. Hwang, H. Wang, J. Pang, W. Wang, Y. Liu, W.-C. Peng *et al.*, "A survey of reasoning and agentic systems in time series with large language models," *arXiv preprint arXiv:2509.11575*, 2025.

[175] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[176] J. Zhang, L. Feng, X. Guo, Y. Wu, Y. Dong, and D. Xu, "Timemaster: Training time-series multimodal llms to reason via reinforcement learning," *arXiv preprint arXiv:2506.13705*, 2025.

[177] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu *et al.*, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[178] Z. Liu, P. Han, H. Yu, H. Li, and J. You, "Time-r1: Towards comprehensive temporal reasoning in llms," *arXiv preprint arXiv:2505.13508*, 2025.

[179] Y. Xiao, E. Sun, T. Chen, F. Wu, D. Luo, and W. Wang, "Trading-r1: Financial trading with llm reasoning via reinforcement learning," *arXiv preprint arXiv:2509.11420*, 2025.

[180] T. Wu and J. Ortiz, "Rlad: Time series anomaly detection through reinforcement learning and active learning," *arXiv preprint arXiv:2104.00543*, 2021.

[181] Y. Li, P. Ni, and V. Chang, "Application of deep reinforcement learning in stock trading strategies and stock forecasting," *Computing*, vol. 102, no. 6, pp. 1305–1322, 2020.

[182] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in neural information processing systems*, vol. 36, pp. 53 728–53 741, 2023.

[183] W. Niu, Z. Xie, Y. Sun, W. He, M. Xu, and C. Hao, "Langtime: A language-guided unified model for time series forecasting with proximal policy optimization," in *Forty-second International Conference on Machine Learning*, 2025.

[184] N. Karaouli, D. Coquenet, E. Fromont, M. Mermillod, and M. Reyboz, "How foundational are foundation models for time series forecasting?" *arXiv preprint arXiv:2510.00742*, 2025.

[185] V. Ekambaram, A. Jati, P. Dayama, S. Mukherjee, N. Nguyen, W. M. Gifford, C. Reddy, and J. Kalagnanam, "Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series," *Advances in Neural Information Processing Systems*, vol. 37, pp. 74 147–74 181, 2024.

[186] Y. Zhao, J. Li, Z. Song, and Y. Tian, "Language-inspired relation transfer for few-shot class-incremental learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 2, pp. 1089–1102, 2025.

[187] H. Zhao, X. Zhang, J. Wei, Y. Xu, Y. He, S. Sun, and C. You, "Timeseriesscientist: A general-purpose ai agent for time series analysis," *arXiv preprint arXiv:2510.01538*, 2025.

[188] Z. Liu, K. Zeng, Q. Ma, and J. T. Kwok, "Complematch: Boosting time-series semi-supervised classification with temporal-frequency complementarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–14, 2025.

[189] B. Cohen, E. Khwaja, Y. Doubli, S. Lemaachi, C. Lettieri, C. Masson, H. Miccinilli, E. Ramé, Q. Ren, A. Rostamizadeh *et al.*, "This time is different: An observability perspective on time series foundation models," in *Advances in neural information processing systems*, 2025.

[190] H. Prabhakar Kamarthi and B. A. Prakash, "Large pre-trained time series models for cross-domain time series analysis tasks," *Advances in Neural Information Processing Systems*, vol. 37, pp. 56 190–56 214, 2024.

[191] L. N. Darlow, Q. Deng, A. Hassan, M. Asenov, R. Singh, A. Joosen, A. Barker, and A. Storkey, "DAM: Towards a Foundation Model for Forecasting," in *The Twelfth International Conference on Learning Representations*, 2023.

[192] D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu, "TEMPO: Prompt-based Generative Pre-trained Transformer for Time Series Forecasting," in *The Twelfth International Conference on Learning Representations*, 2023.

[193] H. Gui, X. Li, and X. Chen, "Vector Quantization Pretraining for EEG Time Series with Random Projection and Phase Alignment," in *International Conference on Machine Learning*. PMLR, 2024, pp. 16 731–16 750.

[194] S. Lisa, G. Dominic, B. Jan Niklas, Q. Yongrong, Z. Na, K. Dmitry, T. Andreas, S. Fabian, R. Jacob, F. Katrin, D. Sebastian, and B. Philipp, "Trace: Contrastive learning for multi-trial time-series data in neuroscience," in *Advances in neural information processing systems*, 2025.

[195] K.-H. Lai, D. Zha, G. Wang, J. Xu, Y. Zhao, D. Kumar, Y. Chen, P. Zumkhawaka, M. Wan, D. Martinez *et al.*, "Tods: An automated time series outlier detection system," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 18, 2021, pp. 16 060–16 062.

APPENDIX A
RELATED SURVEY ANALYSIS

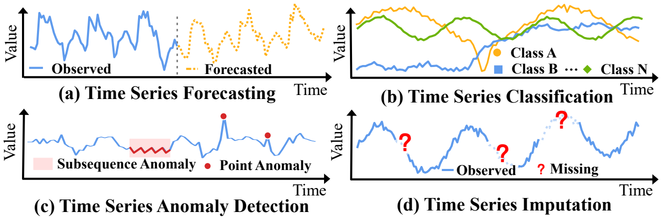
We position our survey through a systematic comparison along four dimensions, including first online timeline, focus topics, data types, and post-training techniques, as summarized in Table II. Ma et al. [12] provide an overview of time-series pre-trained models by categorizing existing approaches into supervised, unsupervised, and self-supervised paradigms. However, their review primarily concentrates on pre-training strategies and covers a limited set of studies that leverage large-scale source datasets (e.g., collections exceeding one million time series) for foundation model construction. Jin et al. [13] present a comprehensive review of large models adapted for time series and spatio-temporal data, organized by data types, model families, scopes, and downstream applications. Notably, this work early on discusses adapting large language foundation models to time-series tasks, which differs from our focus on training time-series foundation models from scratch. Miller et al. [31] mainly center on foundation models for time-series forecasting, with an emphasis on knowledge graphs and large language models fine-tuned using scientific domain knowledge. Liang et al. [32] argue that prior surveys often lack a mechanistic understanding of why foundation models benefit time-series analysis and thus propose a methodology-oriented taxonomy covering architectures, pre-training objectives, adaptation strategies, and data modalities. Similarly, Ye et al. [33] distinguish between pre-training time-series foundation models from scratch and adapting large language models to time-series tasks. Kottapalli et al. [34] provide an in-depth review of pre-trained foundation models for time-series applications, analyzing architectural choices, patch-based representations, training objectives, univariate versus multivariate settings, and probabilistic modeling paradigms. Despite their breadth, these surveys either emphasize pre-training techniques alone or focus predominantly on transferring large language models to time-series domains, without explicitly examining the role of large-scale real-world source datasets in building time-series foundation models.

In contrast to the above works, we argue that large-scale source datasets drawn from real-world time-series scenarios are indispensable for training foundation models from scratch, as reliance solely on synthetic data may limit robustness in complex deployment settings. While Liu et al. [35] offer a comprehensive review of synthetic data generation for time-series foundation model pre-training and fine-tuning with large language models, our survey highlights that real-world source datasets remain a critical yet underexplored component for scalable and reliable TSFM construction. More importantly, existing surveys largely overlook irregular time-series settings and seldom discuss post-training techniques beyond standard fine-tuning. Unlike previous work, this survey adopts a vertical perspective spanning from pre-training to post-training, systematically reviewing from-scratch time-series foundation models, the interplay between real and synthetic source datasets, and emerging post-training strategies—such as reinforcement learning–based post-training—together with their evaluation on diverse target domains.

TABLE II
COMPARISON OF SURVEYS ON FOCUS TOPICS, DATASETS, DATA TYPES, AND POST-TRAINING TECHNIQUES.

Survey	**First Online Time**	**Focus Topic**			**Data Types**		**Post-training Techniques**		
		Datasets	**Pre-training**	**Post-training**	**Regular**	**Irregular**	**Supervised**	**Collaborative**	**Reinforcement**
Ma et al. [12]	May-2023	✗	✓	✗	✓	✗	○	○	○
Jin et al. [13]	Oct-2023	✗	✓	✗	✓	✗	○	○	○
Miller et al. [31]	Jan-2024	✗	✓	✗	✓	✗	○	○	○
Liang et al. [32]	Mar-2024	✗	✓	✗	✓	✗	○	○	○
Ye et al. [33]	May-2024	✗	✓	✗	✓	✗	○	○	○
Liu et al. [35]	Mar-2025	✓	✗	✗	✓	✗	○	○	○
Kottapalli et al. [34]	Apr-2025	✗	✓	✗	✓	✗	○	○	○
Our Survey	Jan-2026	✓	✓	✓	✓	✓	●	●	●

APPENDIX B
TIME SERIES DOWNSTREAM TASKS



(a) Time Series Forecasting

(b) Time Series Classification

(c) Time Series Anomaly Detection

(d) Time Series Imputation

Fig. 16. Common time series downstream tasks for TSFMs.

A. Time Series Forecasting

Time series forecasting (TSF) is one of the fundamental task in time-series analysis [6], [36], aiming to infer future values or trends from historical observations by modeling complex temporal dynamic structures (see Figure 16a). Conceptually, TSF maps the past L observations into the next H steps:

$$\hat{\mathbf{X}}_{t+1:t+H} = F_{\Phi}\big(\mathbf{X}_{t-L+1:t}\big), \tag{10}$$

where $\mathbf{X}_{t-L+1:t} \in \mathbb{R}^{L\times V}$ denotes the past L observations of a V-variate series. H is the forecast horizon, F_{Φ} represents the forecasting model parameterized by Φ.

B. Time Series Classification

Time series classification (TSC) assigns discrete labels to entire sequences, generally of fixed length (Figure 16b). Given a dataset $\mathcal{D} = (\mathbf{X}^{(n)}, \mathbf{Y}^{(n)})_{n=1}^{N}$, where $\mathbf{X}^{(n)} \in \mathbb{R}^{T\times V}$ is the input sequence and $\mathbf{Y}^{(n)} \in \{0,1\}^{C}$ is its one-hot label, the objective is to train a classifier C_{Φ} that maps $\mathbf{X}^{(n)}$ to class scores [7], [188]. A softmax layer then transforms these scores into a probability distribution over the C classes:

$$\hat{\mathbf{p}} = \mathrm{softmax}\big(C_{\Phi}(\mathbf{X})\big), \qquad \hat{y} = \arg\max_{c\in\{1,\dots,C\}} \hat{p}_c, \tag{11}$$

where $\hat{\mathbf{p}} \in [0,1]^{C}$ denotes the estimated class probabilities and $\hat{y}$ represents the predicted label.

C. Time Series Anomaly Detection

Time series anomaly detection (TSAD) identifies point or subsequence outliers that deviate from the overall data distribution [8] (Figure 16c). Given a multivariate time series $\{\mathbf{X}_t\}_{t=1}^{T}$ with $\mathbf{X}_t \in \mathbb{R}^{V}$, TSAD trains a detector to compute a deviation score from local context and converts it into a binary decision via thresholding:

$$s_t = A_{\psi}(\mathbf{X}_{t-w:t+w}), \qquad \hat{y}_t = \mathbf{1}[s_t > \tau], \tag{12}$$

where $\mathbf{X}_{t-w:t+w}$ denotes the multivariate window centered at time t, A_{ψ} is the detection model that assigns a score s_t to each window, τ is the decision threshold, and $\hat{y}_t \in \{\texttt{normal}, \texttt{anomaly}\}$ represents the predicted label.

D. Time Series Imputation

Time series imputation (TSI) aims to estimate missing values using the observed values [9] (see Figure 16d). The general formulation of TSI is defined as:

$$\begin{aligned} \mathbf{Z} &= I_{\Phi}(\mathbf{X}\odot\mathbf{M},\ \mathbf{M},\ \boldsymbol{\Delta}), \\ \hat{\mathbf{X}} &= \mathbf{M}\odot\mathbf{X}\ +\ (\mathbf{1}-\mathbf{M})\odot\mathbf{Z}, \end{aligned} \tag{13}$$

where $\mathbf{X} \in \mathbb{R}^{T\times V}$ denotes the raw series and $\mathbf{Z}$ is the estimated completed series, $\mathbf{M} \in \{0,1\}^{T\times V}$ is the observation mask, and I_{Φ} is the TSI model parameterized by Φ. The operator $\odot$ denotes the Hadamard product, and $\boldsymbol{\Delta}$ (optional) encodes time gaps for irregular sampling, which is crucial for handling continuous-time data.

APPENDIX C
TIME SERIES DATASETS FOR TSFMS

Table III summarizes the statistical characteristics of large-scale source datasets used for pre-training time-series foundation models, including both real-world datasets collected from practical application scenarios and synthetically generated time-series corpora. Table V reports detailed statistics of commonly adopted small-scale target datasets for evaluating time-series foundation models across downstream tasks, including time-series forecasting, classification, anomaly detection, and imputation. Together, these datasets serve as standardized benchmarks for assessing the transferability and effectiveness of TSFMs.

TABLE III
SUMMARY OF LARGE-SCALE SOURCE DATASETS FOR TSFMS. * INDICATES DATASETS WITH IRREGULARLY SAMPLED TIME SERIES. 'ANY' INDICATES SYNTHETIC DATA, WITH THE NUMBER OF SAMPLES FREELY CONFIGURABLE.

Name	Number of	Sequence	Number of	Downstream Tasks			
	Samples (Points)	Length	Channels	TSF	TSC	TSAD	TSI
CTN-TL [60]	15M	512	1	✗	✓	✗	✗
BuildingsBench [62]	1.79M	24/168	1	✓	✗	✗	✗
S2 [63]	Any	512	1	✓	✓	✓	✓
CAUKER [58]	Any	512	1	✗	✓	✗	✗
TimeRCD [64]	Any	100-10,000	1	✗	✗	✓	✗
CloudOps [74]	0.36M	673–8,640	1	✓	✗	✗	✗
MIMIC-IV-ECG [75]	0.8M	5,000	12	✗	✓	✗	✗
ECG-FM [76]	1.5M	5,000	12	✗	✓	✗	✗
MOMENT [11]	13M	512	1	✓	✓	✓	✓
LOTSA [24]	4M	1,000–5,000	1	✓	✗	✗	✗
Lag-LLaMa [71]	11K	1024	1	✓	✗	✗	✗
Chronos [22]	11M	71-350,640	1	✓	✗	✗	✗
TimesFM [70]	207M	37-553,308	1	✓	✗	✗	✗
UTSD [25]	7.4M (28B)	1,440	1	✓	✗	✗	✓
NuTime [21]	1.89M	512	1	✗	✓	✗	✗
Time-300B [29]	48.2M (300B)	-	1	✓	✗	✗	✗
MONSTER [73]	10.3K-59.3M	24-47,998	1-113	✗	✓	✗	✗
DADA [141]	- (400M)	-	1	✗	✗	✓	✗
MIRA [28] *	- (454B)	-	1	✓	✗	✗	✗
BOOM [189]	2.8K (350M)	-	M	✗	✓	✗	✗
EMind [27]	81.1M	128-4,096	1	✓	✗	✗	✗
TSPulse [100]	- (1B)	-	1	✓	✓	✓	✗
UNITS [69]	0.34M	24-1,152	1-963	✓	✓	✗	✗
LPTM [190]	-	-	M	✓	✓	✗	✗
TimeBench [23]	Any (1032B)	480-2,880	1	✓	✗	✗	✗
Time-IMM [52] *	- (1.48M)	1,743-91,098	4-30	✓	✗	✗	✗

TABLE IV
SUMMARY OF EXISTING TSFMS TRAINED FROM SCRATCH.

Method	Year	Publisher	Parameters	Task-Agnostic	Task-Specific
CTN-TL [60]	2022	ISCSCM	-	✗	✓
TimeGPT-1 [138]	2023	arXiv	-	✗	✓
Lag-LLaMa [71]	2023	arXiv	200M	✗	✓
DAM [191]	2024	ICLR	7M	✗	✓
TEMPO [192]	2024	ICLR	-	✗	✓
MOMENT [11]	2024	ICML	385M	✓	✗
Timer [25]	2024	ICML	67M	✓	✗
TimesFM [70]	2024	ICML	200M	✗	✓
Moirai [24]	2024	ICML	311M	✗	✓
VQ-MTM [193]	2024	ICML	-	✗	✓
Chronos [22]	2024	TMLR	710M	✗	✓
NuTime [21]	2024	TMLR	2M	✓	✗
TinyTimeMixers [185]	2024	NeurIPS	<1M	✗	✓
UniMTS [133]	2024	NeurIPS	-	✗	✓
UniTS [69]	2024	NeurIPS	8.24M	✓	✗
TimesBERT [72]	2025	arXiv	85.6M	✓	✗
FFTS [108]	2025	AAAI	-	✓	✗
Time-MoE [29]	2025	ICLR	2.4B	✗	✓
DADA [141]	2025	ICLR	-	✗	✓
Mantis [26]	2025	arXiv	8M	✗	✓
SymTime [63]	2025	arXiv	-	✓	✗
ChatTime [160]	2025	AAAI	350M	✗	✓
D-BETA [111]	2025	ICML	-	✓	✗
Moirai-MoE [30]	2025	ICML	935M	✗	✓
Sundial [23]	2025	ICML	444M	✗	✓
ECGFM [4]	2025	INF	20M	✗	✓
CardX [134]	2025	aXiv	15M	✗	✓
xECG [109]	2025	aXiv	0.35M	✓	✗
EMind [27]	2025	aXiv	8.1M	✓	✗
WirelessGPT [110]	2025	IEEE-Net	80M	✓	✗
TSPulse [100]	2025	arXiv	1M	✓	✗
MIRA [28]	2025	NeurIPS	455M	✗	✓
Trace [194]	2025	NeurIPS	-	✗	✓
TOTO [189]	2025	NeurIPS	151M	✗	✓
TimeRCD [64]	2025	arXiv	-	✗	✓
Chronos-2 [124]	2025	arXiv	120M	✗	✓

APPENDIX D
TIME SERIES PRE-TRAINING FOR TSFMS

Table IV provides a summary of recent time-series foundation models that are pre-trained from scratch using large-scale source datasets. The table compares these models along several key dimensions, including publication year, venue, model size, and the scale of learnable parameters.

In addition, Table IV distinguishes between task-agnostic TSFMs, which are designed to support multiple downstream tasks such as forecasting, classification, and anomaly detection, and task-specific TSFMs that focus on a single objective (e.g., classification-oriented foundation models). This comparison highlights the evolving design choices and modeling trends in large-scale time-series pre-training.

APPENDIX E
POST-TRAINING TECHNIQUE ANALYSIS

Table VI presents a systematic comparison of three representative post-training paradigms for time-series foundation models: supervised, collaborative, and reinforcement-based post-training for TSFMs. The comparison highlights their distinct optimization goals, data dependencies, and learning signals. Supervised fine-tuning relies on large-scale labeled datasets and optimizes standard task losses to improve performance on specific prediction objectives, whereas reinforcement fine-tuning updates model parameters through reward or preference signals, aiming to align model behaviors with external criteria rather than explicit labels.

In contrast, collaborative fine-tuning augments the base TSFM by interacting with auxiliary networks or external modules, enabling joint optimization through coupled loss functions. This paradigm typically requires fewer labeled samples and emphasizes efficient adaptation and cross-modal or cross-model integration. Overall, Table VI clarifies how these paradigms differ in terms of training supervision, reliance on external components, and applicable use cases, providing a concise taxonomy of post-training strategies for TSFMs.

TABLE V
SUMMARY OF SMALL-SCALE TARGET DATASETS FOR TSFMS. FOR DATASETS USED IN TIME SERIES ANOMALY DETECTION (TSAD) TASKS, "# NUMBER OF SAMPLES" REFERS TO THE NUMBER OF POINTS IN EACH SEQUENCE.

Name	Data Types		# Number of Samples			# Series Length	# Channels	Downstream Tasks			
	Regular	Irregular	Train	Validation	Test			TSF	TSC	TSAD	TSI
ETTh1, ETTh2 [6]	✓	✗	8,545	2,881	2,881	{96, ..., 720}	7	✓	✗	✗	✓
ETTm1, ETTm2 [6]	✓	✗	34,465	11,521	11,521	{96, ..., 720}	7	✓	✗	✗	✓
Electricity [77]	✓	✗	18,317	2,633	5,261	{96, ..., 720}	321	✓	✗	✗	✓
Traffic [77]	✓	✗	12,185	1,757	3,509	{96, ..., 720}	862	✓	✗	✗	✓
Weather [77]	✓	✗	36,792	5,271	10,540	{96, ..., 720}	21	✓	✗	✗	✓
Exchange [77]	✓	✗	5,120	665	1,422	{96, ..., 720}	8	✓	✗	✗	✓
ILI [77]	✓	✗	617	74	170	{24, ..., 60}	7	✓	✗	✗	✗
Solar-Energy [78]	✓	✗	36,601	5,161	10,417	{96, 192, 336, 720}	137	✓	✗	✗	✓
PEMS [78]	✓	✗	15,701	5,216	434	{12, 24, 48, 96}	358	✓	✗	✗	✓
128 UCR [82]	✓	✗	60,555	None	130,603	{15, . . . , 2844}	1	✗	✓	✗	✓
30 UEA [83]	✓	✗	61,149	None	40,770	{8, . . . , 17,984}	{2, . . . , 1,345}	✗	✓	✗	✓
250 UCR [54]	✓	✗	2,238,349	None	6,143,541	{6,684, . . . , 900,000}	1	✗	✗	✓	✗
MSL [8]	✓	✗	44,653	11,664	73,729	44,653 / 73,729	55	✗	✗	✓	✗
SMAP [8]	✓	✗	108,146	27,037	427,617	108,146 / 427,617	25	✗	✗	✓	✗
PSM [8]	✓	✗	105,984	26,497	87,841	105,984 / 87,841	25	✗	✗	✓	✗
SMD [8]	✓	✗	566,724	141,681	708,420	566,724 / 708,420	38	✗	✗	✓	✗
SWaT [8]	✓	✗	396,000	99,000	449,919	396,000 / 449,919	51	✗	✗	✓	✗
NIPS-TS-SWAN [195]	✓	✗	60,000	None	60,000	60,000	38	✗	✗	✓	✗
NIPS-TS-GECCO [195]	✓	✗	69,260	None	69,261	69,261	9	✗	✗	✓	✗
MIMIC-III [79]	✗	✓	17,212	1,913	2,125	96	96	✓	✓	✗	✗
MIMIC-IV [79]	✗	✓	14,477	1,609	1,788	971	100	✓	✗	✗	✗
PhysioNet'12 [79]	✗	✓	9,704	1,078	1,199	48	36	✓	✓	✗	✓
PhysioNet'19 [86]	✗	✓	31,042	3,880	3,881	60	34	✗	✓	✗	✗
Human Activity [80]	✗	✓	949	193	218	131	12	✓	✗	✗	✗
USHCN [81]	✗	✓	902	100	112	337	5	✓	✗	✗	✗
PAM [87]	✗	✓	4,266	533	534	600	17	✗	✓	✗	✗

TABLE VI
COMPARISON OF DIFFERENT POST-TRAINING PARADIGMS FOR TSFMS.

Aspect	**Supervised Fine-tuning**	**Collaborative Fine-tuning**	**Reinforcement Fine-tuning**
Main Objective	Fit model to labeled data	Enhance model via external models/modules	Align model via reward signals
Data Requirement	Large labeled dataset	Base model + auxiliary network	Preference/reward data
Training Signal	Supervised loss (e.g., CE, MSE)	Joint loss from base and external models	Reward-based optimization
External Models	✗	✓	✗
Typical Techniques	Standard backprop	Adapter, LoRA, multi-modal fusion	PPO, DPO
Common Use Cases	Classification, regression	Multi-modal integration, efficient adaptation	Alignment with human preferences
Optimization Target	$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}\, \mathcal{L}(f_\theta(x), y)$	$\min_{\theta,\phi} \mathcal{L}(f_\theta(x), y) + \lambda \mathcal{L}_{ext}(g_\phi(x), f_\theta(x))$	$\max_{\theta} \mathbb{E}_{x\sim\mathcal{D}}\, R(f_\theta(x))$