R-Tuning: Wavelet-Decomposed Replay and Semantic Alignment for Continual Adaptation of Pretrained Time-Series Models

Tianyi Yin[1,2], Jingwei Wang[2,3*], Chenze Wang[2], Han Wang[2], Jiexuan Cai[2], Min Liu[2], Yunlong Ma[2*], Kun Gao[4], Yuting Song[5], Weiming Shen[6]

[1]Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University
[2]College of Electronics and Information Engineering, Tongji University
[3]Shanghai Institute of Intelligent Science and Technology, Tongji University
[4]Zhongguancun Academy
[5]Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A*STAR)
[6]Huazhong University of Science and Technology
{yin_tianyi, jwwang, wangchenze, 2210126, 2431940, lmin, evanma}@tongji.edu.cn,
gaokun@bjzgca.edu.cn, Song_Yuting@a-star.edu.sg, wshen@ieee.org

Abstract

Pre-trained models have demonstrated exceptional generalization capabilities in time-series forecasting; however, adapting them to evolving data distributions remains a significant challenge. A key hurdle lies in accessing the original training data, as fine-tuning solely on new data often leads to catastrophic forgetting. To address this issue, we propose Replay Tuning (R-Tuning), a novel framework designed for the continual adaptation of pre-trained time-series models. R-Tuning constructs a unified latent space that captures both prior and current task knowledge through a frequency-aware replay strategy. Specifically, it augments model-generated samples via wavelet-based decomposition across multiple frequency bands, generating trend-preserving and fusion-enhanced variants to improve representation diversity and replay efficiency. To further reduce reliance on synthetic samples, R-Tuning introduces a latent consistency constraint that aligns new representations with the prior task space. This constraint guides joint optimization within a compact and semantically coherent latent space, ensuring robust knowledge retention and adaptation. Extensive experimental results demonstrate the superiority of R-Tuning, which reduces MAE and MSE by up to 46.9% and 46.8%, respectively, on new tasks, while preserving prior knowledge with gains of up to 5.7% and 6.0% on old tasks. Notably, under few-shot settings, R-Tuning outperforms all state-of-the-art baselines even when synthetic proxy samples account for only 5% of the new task dataset.

Code — https://github.com/Ivan-YinTY/R-Tuning
Extended version — https://arxiv.org/abs/2511.11685

Introduction

Time-series forecasting plays a crucial role in various domains, including transportation, energy systems, and industrial management (Wang et al. 2024a). Traditional forecasting methods, including statistical models and early deep learning models, have demonstrated strong performance on task-specific datasets (Li, Wu, and Liu 2023). However, they typically operate under a one-model-per-task paradigm, which limits their capability to diverse and evolving real-world conditions (Zhang et al. 2024).

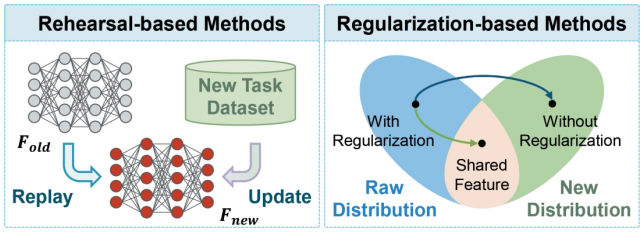

Figure 1: Two types of adaptation methods. Rehearsal-based methods mitigate forgetting by training on synthesized samples and new data. Regularization-based methods constrain parameter updates within the shared feature space.

The emergence of pre-trained models (PTMs) has enabled a shift toward general-purpose forecasting systems (Kim et al. 2025). By converting temporal signals into discrete tokens and leveraging transformer-based architectures, PTMs can perform a wide range of forecasting tasks without explicit fine-tuning (Gruver et al. 2023). These models enhance generalization and reusability across tasks. However, once released or deployed, PTMs often become static and cannot adapt smoothly to new or evolving knowledge encountered over time.

Existing adaptation strategies require access to both the original training data and new task data to update the model jointly (Gu et al. 2024). However, for PTMs, this is rarely practical due to concerns over data privacy, proprietary usage restrictions, and the high cost of fine-tuning large models (Zhou et al. 2024a). Conversely, updating the model with only new data causes catastrophic forgetting, where prior task performance deteriorates significantly.

Continual learning methods offer promising solutions for

*Corresponding authors: Yunlong Ma, Jingwei Wang.

this issue, especially generative rehearsal and regularization-based approaches (Zhou et al. 2024b). As shown in Figure 1, generative replay synthesizes past data to retain prior knowledge, but identifying informative and representative samples from the generated distribution is non-trivial, especially for time-series signals that contain multi-scale temporal patterns (Cai et al. 2024). On the other hand, regularization-based methods aim to limit the extent to which the model deviates from prior knowledge of the task. However, they often face a familiar dilemma: learn fast, forget faster, meaning that learning new tasks more effectively usually leads to forgetting old ones more quickly (Biderman et al. 2024).

To address these challenges, we propose Replay Tuning (R-Tuning), a novel framework for continual adaptation specifically designed for PTMs in time-series forecasting. Our method integrates wavelet-based replay to improve the representativeness of old-task samples and semantic alignment to preserve latent knowledge structures across tasks, enabling stable and efficient adaptation to new tasks without forgetting. Specifically, we generate frequency-aware synthetic samples through multi-band wavelet decomposition, preserving trend-level information while enriching sample diversity. These samples are combined with new task data to guide adaptation. Simultaneously, a semantic alignment constraint ensures the latent space of the adapted model remains consistent with the original, enabling stable knowledge integration and reducing the dependence on large replay buffers.

Our contributions are summarized as follows:

1. We propose R-Tuning, a continual learning method that enables efficient and stable adaptation of pre-trained time-series models through wavelet-based replay and semantic alignment.
2. We demonstrate that using only 4%–5% synthetic samples (relative to the new task dataset size) is sufficient to achieve optimal performance across old and new tasks.
3. Extensive experiments on multiple benchmarks validate the generalization and efficiency of our method, consistently outperforming state-of-the-art methods under few-shot scenarios.

Related Work

Pre-trained Models for Time-Series Forecasting

Foundation models for time-series forecasting aim to generalize across diverse domains, tasks, and temporal patterns by learning transferable representations from large-scale time-series corpora (Liu et al. 2025a). Current designs largely follow three architectural paradigms: encoder-decoder, encoder-only, and decoder-only.

Encoder-Decoder Architectures Encoder-decoder architectures integrate a context-aware encoder with a generative decoder, enabling sequence-to-sequence modeling for forecasting tasks (Liu et al. 2025b). These models typically support flexible conditioning and multi-resolution generation. For instance, Chronos (Ansari et al. 2024) extends the encoder-decoder structure from the T5 language model to temporal data, introducing a task-aware forecasting interface. Apollo-Forecast (Yin et al. 2025) further enhances Chronos by incorporating anti-aliasing tokenization and a parallel decoding mechanism, thereby improving inference accuracy and efficiency.

While encoder-decoder models exhibit strong generalization performance across a wide range of forecasting tasks, their architectural complexity leads to increased training cost and lower efficiency, especially under constrained parameter budgets.

Encoder-Only Architectures To simplify model structure and improve training efficiency, some approaches adopt encoder-only designs that extract contextual representations from the input sequence for direct forecasting (Liang et al. 2024). AutoTimes and Moirai are two representative encoder-only models. AutoTimes (Liu et al. 2024a) leverages decoder-only PTMs to perform autoregressive time-series forecasting with flexible context and minimal trainable parameters. Moirai (Woo et al. 2024) adopts a masked encoder design to address cross-frequency learning and multivariate forecasting, achieving strong zero-shot performance across diverse datasets.

However, encoder-only models suffer from limited expressiveness due to their low-rank structure (Dong, Cordonnier, and Loukas 2021) and often underperform in generative forecasting tasks that involve long-term, fine-grained extrapolation.

Decoder-Only Architectures Decoder-only architectures generate time series in an autoregressive manner, akin to language models. Their unidirectional attention structure aligns naturally with the forecasting objective, enabling strong modeling of sequential dependencies (Das et al. 2024). Representative models include TimerXL, which utilizes linear-time attention to handle long input sequences efficiently (Liu et al. 2024b), and GPT4TS, which enhances generalization and forecasting accuracy by modifying the feedforward layers within residual blocks (Zhou et al. 2023). These models are favored for their architectural simplicity and long-term forecasting capabilities.

However, despite these strengths, current foundation models (regardless of design) remain static after pretraining or deployment (Han et al. 2025). They lack mechanisms to adapt to new data distributions or evolving task requirements continually, motivating recent efforts to explore continual learning and adaptation techniques tailored for time-series foundation models.

Continual Learning for PTMs

Among the various continual learning paradigms, only a few are practical for PTMs due to common constraints such as fixed architectures, inaccessibility of original training data, and high retraining costs (Wang et al. 2024b). In this context, two mainstream approaches have gained traction: rehearsal-based and regularization-based methods.

Rehearsal-based Methods Rehearsal-based methods aim to preserve prior knowledge by constructing a shared feature space between old and new tasks (Xu et al. 2024). A

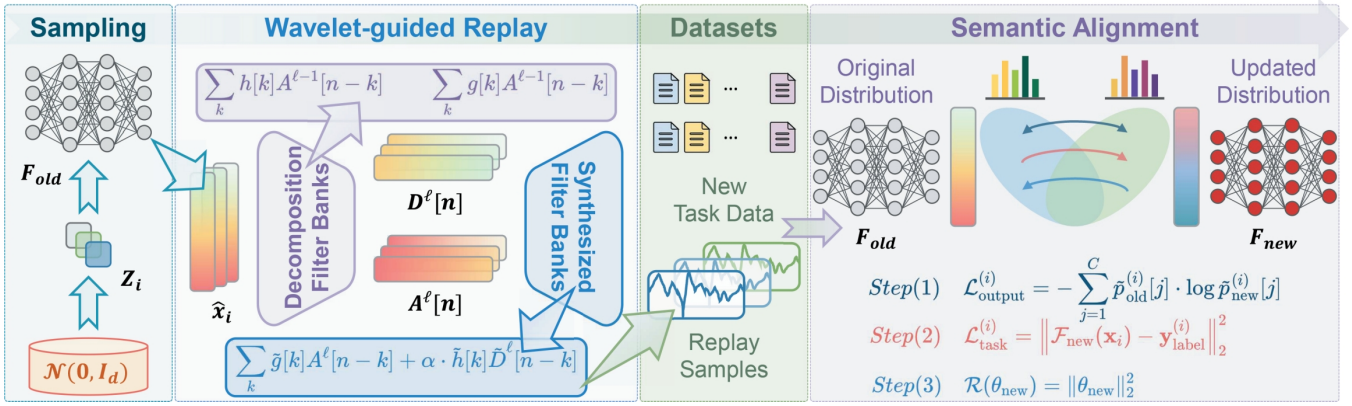

Figure 2: The architecture of R-Tuning. Random seeds are sampled from the Gaussian distribution as inputs to the old model. High-quality synthetic samples are generated via the Wavelet-guided Replay mechanism and combined with the new task dataset for fine-tuning. The Semantic Alignment mechanism ensures a performance balance between the new and old tasks.

common strategy involves synthesizing proxy samples from previous tasks using PTMs, then mixing them with new data during adaptation (Sun, Ho, and Lee 2019), enabling replay without access to the original datasets. Feature-level variants, such as RanPAC (McDonnell et al. 2023), further reduce reliance on raw data by applying random projection to compress embeddings, followed by clustering to select a representative coreset for replay (Tong et al. 2025). These methods are computationally efficient and lightweight.

However, such approaches often ignore the multi-frequency nature of time-series signals. Samples selected purely based on spatial similarity may lack critical temporal structures (especially low-frequency trends and high-frequency variations), limiting the expressiveness of the replay buffer. It becomes a bottleneck in adapting PTMs for real-world forecasting tasks involving distributional shifts.

Regularization-based Methods Regularization-based methods constrain the model to retain old knowledge by penalizing significant deviations in parameters or outputs. For example, Elastic Weight Consolidation (EWC) introduces a Fisher-based penalty to minimize updates on important weights (Kurniawan et al. 2024), while Learning without Forgetting (LwF) utilizes distillation to align predictions between the old and new models (Bonato et al. 2024).

Despite their effectiveness, these methods inherently face a trade-off: restricting model drift helps preserve old tasks but often hinders learning new ones (Kalajdzievski 2024). This "learn fast, forget faster" dilemma highlights the need for more adaptive strategies that can achieve both knowledge retention and generalization, especially in time-series settings where signals evolve over time and across domains.

Methodology

To address the challenges of adapting pre-trained time-series models without access to the original training data, we propose a novel continual fine-tuning method named Replay Tuning (R-Tuning). R-Tuning builds a compact, frequency-aware training sample space that jointly represents both the old and new task distributions. It comprises two key modules: Wavelet-guided Replay, which synthesizes and selects high-quality replay samples using frequency-aware decomposition, and Semantic Alignment, which maintains consistency between new and old task representations by aligning their latent semantic spaces. An overview of the proposed approach is shown in Figure 2.

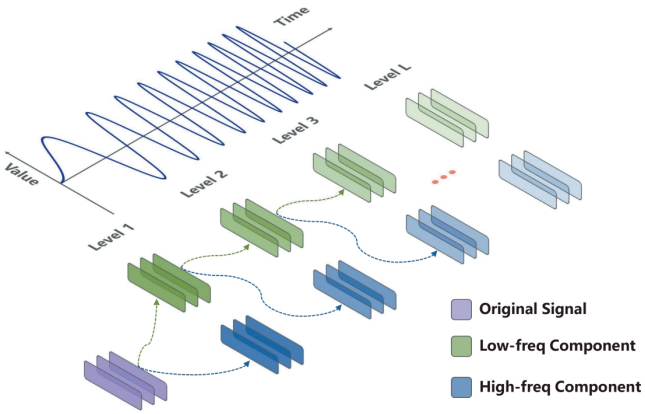

Figure 3: Decompose the signal using Redundant Wavelet Transform. Freq denotes frequency.

Wavelet-guided Replay

The Wavelet-guided Replay module enhances the quality and representativeness of replayed samples by leveraging the frequency structures inherent in time-series data. It consists of two components: Wavelet-based Sample Synthesis and Frequency-aware Replay Strategy.

Wavelet-based Sample Synthesis We begin by sampling $\{\mathbf{z}_i\}_{i=1}^N$ from a standard multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\mathbf{z}_i \in \mathbb{R}^d$ and d is the input dimensionality of the pretrained model. These vectors are passed into a frozen pretrained model $\mathcal{F}_{\text{old}}$ to produce synthetic time series samples $\hat{\mathbf{x}}_i \in \mathbb{R}^T$, where T denotes the sequence length. Each

synthetic sequence $\hat{\mathbf{x}}_i$ captures temporal features embedded in the original training distribution, serving as proxies for previously seen data.

To extract frequency-localized components, we apply the Redundant Wavelet Transform (RWT) to each synthetic sequence. We initialize the approximation as $A^0 = \hat{\mathbf{x}}_i$, and recursively decompose the signal up to level L, generating a set of approximation coefficients $\{A^\ell\}_{\ell=1}^L$ and detail coefficients $\{D^\ell\}_{\ell=1}^L$ as follows:

$$D^\ell[n] = \sum_k h[k] A^{\ell-1}[n-k], \tag{1}$$

$$A^\ell[n] = \sum_k g[k] A^{\ell-1}[n-k], \tag{2}$$

where $h[k]$ and $g[k]$ are the high-pass and low-pass decomposition filters derived from the Daubechies wavelet of order 4 (db4).

The above process is shown in Figure 3. The decomposition low-pass filter coefficients for db4 are expressed as:

$$g[0] = \frac{1+\sqrt{3}}{4\sqrt{2}}, \qquad g[1] = \frac{3+\sqrt{3}}{4\sqrt{2}}, \tag{3}$$

$$g[2] = \frac{3-\sqrt{3}}{4\sqrt{2}}, \qquad g[3] = \frac{1-\sqrt{3}}{4\sqrt{2}}, \tag{4}$$

and the corresponding high-pass decomposition filter is defined by:

$$h[k] = (-1)^k g[3-k], \quad \text{for } k = 0, 1, 2, 3. \tag{5}$$

The reconstruction filters $\tilde{g}[k]$ and $\tilde{h}[k]$ are derived using time-reversal and sign alternation operations on $g[k]$ and $h[k]$, respectively:

$$\tilde{g}[k] = g[k], \quad \tilde{h}[k] = (-1)^k g[k], \tag{6}$$

forming the synthesis filter bank used to reconstruct time-domain signals from wavelet coefficients and ensuring perfect reconstruction.

Unlike the discrete wavelet transform, which incorporates downsampling that may introduce aliasing even with orthogonal mirror filters, the RWT avoids this issue by inserting zeros between coefficients. This design enhances the effective receptive field while preserving temporal resolution.

Frequency-aware Replay Strategy Building upon the multiscale decomposition described above, we construct multiple variants of each synthetic signal, each emphasizing distinct spectral characteristics to reflect specific temporal properties. Specifically, we remove the highest k levels of detail coefficients $\{D^{L-k+1}, \ldots, D^L\}$ and perform reconstruction via the Inverse Redundant Wavelet Transform.

For each level ℓ from L down to 1, the reconstruction is carried out iteratively as:

$$A^{\ell-1}[n] = \sum_k \tilde{g}[k] A^\ell[n-k] + \alpha \cdot \tilde{h}[k] D^\ell[n-k], \tag{7}$$

where $\alpha \in [0, 1]$ is a level-dependent detail scaling coefficient that controls the contribution of high-frequency components during reconstruction. This modification enables smoother spectral filtering and generates frequency-aware replay samples with varying degrees of local variation. After completing all inverse steps, we obtain the reconstructed signal $\tilde{\mathbf{x}}_i^{(L-k)} = A^0$, which preserves components from the lower-frequency subbands while omitting selected high-frequency details.

This strategy enables the generation of multiple replay samples for each synthetic sequence, each emphasizing different spectral bands and encoding complementary characteristics, such as global trends and local fluctuations. These frequency-aware variants enrich the replay set by expanding

Algorithm 1: Replay Tuning (R-Tuning)

Input: Pretrained model $\mathcal{F}_{\text{old}}$, new data $\mathcal{D}_{\text{new}}$
Parameter: Latent dim d, sequence length T, wavelet levels L, detail discard depth k, replay size N, scale factor α, temperature τ, loss weights λ, β
Output: Adapted model $\mathcal{F}_{\text{new}}$

Sample latent codes: $\{\mathbf{z}_i\}_{i=1}^N \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
Generate synthetic samples: $\hat{\mathbf{x}}_i = \mathcal{F}_{\text{old}}(\mathbf{z}_i), \quad \hat{\mathbf{x}}_i \in \mathbb{R}^T$
for $i = 1$ to N **do**
  Initialize $A^0 = \hat{\mathbf{x}}_i$
  for $\ell = 1$ to L **do**
    $D^\ell[n] = \sum_k h[k] A^{\ell-1}[n-k]$
    $A^\ell[n] = \sum_k g[k] A^{\ell-1}[n-k]$
  end for
  for $j = 1$ to k **do**
    for $\ell = L$ down to 1 **do**
      $\tilde{D}^\ell = \begin{cases} D^\ell, & \ell \le L-j \\ \mathbf{0}, & \ell > L-j \end{cases}$
      $A^{\ell-1}[n] = \sum_k \tilde{g}[k] A^\ell[n-k] + \alpha \cdot \tilde{h}[k] \tilde{D}^\ell[n-k]$
    end for
    $\tilde{\mathbf{x}}_i^{(j)} = A^0$
  end for
end for
Construct replay set: $\mathcal{D}_{\text{replay}} = \{\hat{\mathbf{x}}_i\} \cup \{\tilde{\mathbf{x}}_i^{(j)}\}$
Merge to training set: $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{new}} \cup \mathcal{D}_{\text{replay}}$
Initialize: $\mathcal{F}_{\text{new}} \leftarrow \mathcal{F}_{\text{old}}$
for $\mathbf{x}_i \in \mathcal{D}_{\text{train}}$ **do**
  $\mathbf{y}_{\text{old}} = \mathcal{F}_{\text{old}}(\mathbf{x}_i), \quad \mathbf{y}_{\text{new}} = \mathcal{F}_{\text{new}}(\mathbf{x}_i)$
  $\tilde{p}_{\text{old}}^{(i)}[j] = \frac{e^{y_{\text{old},j}^{(i)}/\tau}}{\sum_{k=1}^C e^{y_{\text{old},k}^{(i)}/\tau}}, \quad \tilde{p}_{\text{new}}^{(i)}[j] = \frac{e^{y_{\text{new},j}^{(i)}/\tau}}{\sum_{k=1}^C e^{y_{\text{new},k}^{(i)}/\tau}}$
  $\mathcal{L}_{\text{output}}^{(i)} = -\sum_{j=1}^C \tilde{p}_{\text{old}}^{(i)}[j] \cdot \log \tilde{p}_{\text{new}}^{(i)}[j]$
  $\mathcal{L}_{\text{task}}^{(i)} = \left\| \mathcal{F}_{\text{new}}(\mathbf{x}_i) - \mathbf{y}_{\text{label}}^{(i)} \right\|_2^2$
end for
Compute average losses:
$\mathcal{L}_{\text{output}} = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_i \mathcal{L}_{\text{output}}^{(i)}$
$\mathcal{L}_{\text{task}} = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_i \mathcal{L}_{\text{task}}^{(i)}$
Compute regularization: $\mathcal{R}(\theta_{\text{new}}) = \|\theta_{\text{new}}\|_2^2$
Compute total loss:
$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{output}} + \beta \cdot \mathcal{R}(\theta_{\text{new}})$
Update $\mathcal{F}_{\text{new}}$ via gradient descent on $\mathcal{L}_{\text{total}}$
return $\mathcal{F}_{\text{new}}$

the diversity of temporal dynamics while preserving task-specific features from previous domains.

We further construct the final training set $\mathcal{D}_{\text{train}}$ by combining the new task samples with the full-spectrum and frequency-filtered synthetic signals:

$$\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{new}} \cup \{\hat{\mathbf{x}}_i\}_{i=1}^{N} \cup \{\tilde{\mathbf{x}}_i^{(L-k)}\}_{i=1}^{N'}. \tag{8}$$

This comprehensive dataset facilitates continual adaptation by simultaneously encoding old-task knowledge and new-task representations.

Semantic Alignment via Latent Distillation

To complement the Wavelet-guided Replay module and further improve the retention of prior knowledge while reducing reliance on synthetic sample quality, we introduce a Semantic Alignment module based on latent space distillation. It enhances the consistency between the new model's output distribution and that of the raw pretrained model, thereby reinforcing task-specific knowledge and promoting more stable continual adaptation without requiring access to original training data.

Distillation Mechanism To mitigate semantic drift and prevent catastrophic forgetting during continual adaptation, we adopt a distillation mechanism that transfers knowledge from the original pretrained model $\mathcal{F}_{\text{old}}$ to the new model $\mathcal{F}_{\text{new}}$. This mechanism aligns the output distributions of the two models on the replay-augmented dataset $\mathcal{D}_{\text{train}}$, thereby maintaining consistency with prior knowledge even when original task labels are unavailable.

For each training input $\mathbf{x}_i \in \mathcal{D}_{\text{train}}$, we obtain the output logits from both models:

$$\mathbf{y}_{\text{new}} = \mathcal{F}_{\text{new}}(\mathbf{x}_i), \quad \mathbf{y}_{\text{old}} = \mathcal{F}_{\text{old}}(\mathbf{x}_i). \tag{9}$$

To produce softened prediction distributions that reveal relative confidence, we apply temperature scaling with $\tau \in (0, 1]$ directly to the logits. Specifically, the softened prediction probabilities for class j are given by:

$$\tilde{p}_{\text{old}}^{(i)}[j] = \frac{\exp(y_{\text{old},j}^{(i)}/\tau)}{\sum_{k=1}^{C} \exp(y_{\text{old},k}^{(i)}/\tau)}, \tag{10}$$

$$\tilde{p}_{\text{new}}^{(i)}[j] = \frac{\exp(y_{\text{new},j}^{(i)}/\tau)}{\sum_{k=1}^{C} \exp(y_{\text{new},k}^{(i)}/\tau)}, \tag{11}$$

where C is the number of output classes and $y_{\text{old},j}^{(i)}$ and $y_{\text{new},j}^{(i)}$ are the j-th logits from the old and new models, respectively. The distillation loss is computed as the cross-entropy between these softened distributions:

$$\mathcal{L}_{\text{output}}^{(i)} = -\sum_{j=1}^{C} \tilde{p}_{\text{old}}^{(i)}[j] \cdot \log \tilde{p}_{\text{new}}^{(i)}[j], \tag{12}$$

and its expectation over the training set is:

$$\mathcal{L}_{\text{output}} = \mathbb{E}_{\mathbf{x}_i \sim \mathcal{D}_{\text{train}}} \left[\mathcal{L}_{\text{output}}^{(i)} \right]. \tag{13}$$

This loss encourages $\mathcal{F}_{\text{new}}$ to approximate the predictive behavior of $\mathcal{F}_{\text{old}}$, preserving prior task knowledge under a softened output distribution. Temperature scaling not only improves the informativeness of supervision by spreading the probability mass but also smooths the optimization landscape. From a gradient perspective, the derivative of the temperature-scaled softmax is:

$$\frac{\partial \tilde{p}[j]}{\partial z_m} = \frac{1}{\tau} \cdot \tilde{p}[j] \cdot (\delta_{jm} - \tilde{p}[m]), \tag{14}$$

where $\tilde{p}[j]$ denotes the temperature-scaled probability and δ_{jm} is the Kronecker delta.

As τ increases, the gradient magnitude decreases, which flattens the loss surface and stabilizes training. Therefore, distillation not only acts as a semantic regularizer but also as an optimization stabilizer for continual learning.

Training Objective The training objective integrates both the original task loss and the distillation loss. On one hand, we minimize the task-specific loss using the ground truth targets:

$$\mathcal{L}_{\text{task}} = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{\mathbf{x}_i \in \mathcal{D}_{\text{train}}} \left\| \mathcal{F}_{\text{new}}(\mathbf{x}_i) - \mathbf{y}_{\text{label}}^{(i)} \right\|_2^2, \tag{15}$$

where $\mathbf{y}_{\text{label}}^{(i)}$ is the ground truth target for input $\mathbf{x}_i$.

At the same time, we incorporate the knowledge from the frozen model via the distillation loss, leading to a composite objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{output}} + \beta \mathcal{R}(\theta), \tag{16}$$

where λ and β are hyperparameters that control the trade-off among the task loss, the distillation loss, and a regularization term $\mathcal{R}(\theta)$ is implemented via L2 weight decay in the Adam.

This joint optimization enables the model to integrate new knowledge while maintaining performance on previous tasks, thereby improving stability and generalization in continual learning.

Experiments

Baselines and Experimental Settings

To assess the effectiveness of the proposed method, we conduct experiments using six SOTA pretrained time-series forecasting models, spanning encoder-decoder (Chronos, Apollo), encoder-only (Moirai, AutoTimes), and decoder-only (TimerXL, GPT4TS) architectures. These models represent diverse architectural designs and are widely adopted as strong baselines in time-series prediction tasks. We compare our approach with four categories of continual adaptation methods: rehearsal-based (RanPAC, LAMOL), regularization-based (LwF, EWC), vanilla fine-tuning (FT), and frozen-parameter adaptation. Experiments are conducted on five real-world public datasets (ETT, ECL, Traffic, Weather, and Solar), covering a broad range of domains, including energy, meteorology, and transportation (Liu et al. 2024c). To further avoid potential data leakage issues during pretraining, we also include an in-house crowd flow dataset collected from a commercial area in Shanghai, spanning from January 1, 2024, to July 1, 2025, with pedestrian counts recorded at 5-minute intervals. All experiments are run on cloud servers equipped with dual L40 (48GB) GPUs.

	Method	Chronos	Apollo	Moirai	AutoTimes	TimerXL	GPT4TS
MAE	Raw	5.3±0.7	5.3±1.0	5.2±1.1	5.5±1.1	4.9±0.9	5.6±1.0
	FT	5.4±0.8/-1.13%	5.3±0.6/-0.19%	5.3±1.2/-2.31%	5.6±1.5/-2.56%	4.9±0.3/+1.21%	5.6±1.0/+0.71%
	Frozen	5.3±0.4/-0.75%	5.3±1.3/+0.00%	5.1±0.7/+1.92%	5.4±0.9/+0.92%	5.0±1.1/-1.42%	5.5±1.4/+2.32%
	LwF	5.3±0.5/+0.19%	5.2±1.0/+1.51%	5.0±0.8/+3.27%	5.5±1.2/-1.10%	4.8±0.7/+2.63%	5.4±0.6/+3.57%
	EWC	5.2±1.4/+1.13%	5.2±0.9/+1.13%	5.2±1.1/-0.77%	5.3±1.3/+2.38%	4.9±0.5/+0.61%	5.5±0.8/+1.60%
	RanPAC	5.2±0.3/+1.51%	5.1±0.7/+3.77%	5.1±1.5/+1.73%	5.1±0.4/+6.59%	4.8±0.9/+2.23%	5.5±1.2/+1.96%
	LAMOL	5.3±0.6/+0.00%	5.2±1.4/+1.13%	5.1±0.5/+1.35%	5.4±1.0/+0.55%	4.8±1.3/+3.04%	5.4±0.7/+3.03%
	R-Tuning	5.2±1.1/+2.83%	5.1±0.8/+4.15%	5.1±1.3/+2.31%	5.2±0.6/+4.21%	4.9±0.2/+1.62%	5.3±1.5/+5.70%
MSE	Raw	6.9±1.1	6.9±1.0	6.7±0.9	7.1±0.8	6.4±1.1	7.3±1.1
	FT	7.4±1.3/-8.31%	7.8±0.4/-13.70%	7.4±0.7/-9.36%	7.7±1.0/-8.78%	7.0±1.5/-9.08%	8.1±0.9/-11.85%
	Frozen	6.6±0.6/+3.21%	6.8±1.1/+0.73%	6.7±0.3/+0.00%	6.9±0.8/+1.56%	6.4±1.2/-0.47%	7.5±0.7/-2.89%
	LwF	6.9±1.0/-0.87%	7.0±1.5/-2.19%	6.8±1.4/-0.59%	7.1±0.5/-0.71%	6.5±0.4/-2.19%	7.2±1.3/+0.69%
	EWC	7.2±1.2/-4.37%	7.1±0.7/-3.35%	6.8±0.9/-0.89%	7.1±0.6/-1.27%	6.5±1.0/-2.19%	7.4±1.4/-1.93%
	RanPAC	6.7±0.5/+3.06%	6.8±0.8/+0.29%	6.4±1.1/+4.31%	6.8±0.7/+3.97%	6.2±0.9/+2.82%	6.9±1.2/+4.41%
	LAMOL	6.9±1.4/+0.15%	7.0±0.3/-2.04%	6.6±0.8/+2.23%	6.9±1.1/+2.83%	6.4±0.6/+0.00%	7.0±0.5/+3.44%
	R-Tuning	6.7±1.0/+2.77%	6.5±1.5/+5.98%	6.4±0.4/+5.50%	6.8±0.9/+3.82%	6.1±0.7/+4.23%	7.1±0.3/+2.07%

Table 1: Average MAE/MSE (with error margins) on **Old** tasks. The relative percentages (calculated with 3 decimal places precision) are provided after slashes. Best results are underlined. All absolute values have been scaled by a factor of 10.

	Method	Chronos	Apollo	Moirai	AutoTimes	TimerXL	GPT4TS
MAE	Raw	1.9±0.3	1.9±0.2	1.9±0.4	2.0±0.3	1.8±0.3	2.0±0.2
	FT	1.1±0.2/+42.19%	1.1±0.3/+44.27%	1.1±0.1/+41.27%	1.1±0.4/+43.43%	1.0±0.3/+41.90%	1.2±0.2/+42.65%
	Frozen	1.9±0.3/+2.08%	1.9±0.1/+3.12%	1.9±0.2/+1.59%	2.0±0.4/-0.51%	1.7±0.2/+3.35%	2.0±0.3/+4.41%
	LwF	1.3±0.4/+33.33%	1.2±0.2/+34.90%	1.2±0.3/+35.98%	1.3±0.1/+34.85%	1.2±0.2/+33.52%	1.4±0.4/+33.82%
	EWC	1.2±0.1/+35.42%	1.2±0.3/+38.02%	1.2±0.2/+35.98%	1.3±0.2/+32.32%	1.1±0.4/+36.31%	1.3±0.1/+35.29%
	RanPAC	1.2±0.3/+39.58%	1.1±0.2/+42.71%	1.1±0.1/+41.27%	1.1±0.3/+42.42%	1.0±0.2/+41.90%	1.2±0.4/+41.18%
	LAMOL	1.0±0.3/+45.31%	1.1±0.2/+41.67%	1.1±0.4/+43.92%	1.1±0.1/+44.95%	1.0±0.2/+43.02%	1.2±0.3/+43.14%
	R-Tuning	1.0±0.2/+45.31%	1.0±0.3/+46.88%	1.1±0.1/+41.27%	1.1±0.4/+44.44%	1.0±0.3/+41.90%	1.1±0.2/+46.57%
MSE	Raw	1.5±0.2	1.5±0.3	1.5±0.1	1.6±0.4	1.4±0.2	1.6±0.3
	FT	0.9±0.3/+42.21%	0.9±0.1/+44.16%	0.9±0.2/+39.07%	1.0±0.2/+39.24%	0.8±0.4/+42.66%	1.0±0.1/+41.72%
	Frozen	1.5±0.4/+3.90%	1.4±0.2/+6.49%	1.4±0.3/+5.30%	1.5±0.1/+5.70%	1.4±0.3/+5.59%	1.5±0.2/+4.91%
	LwF	1.0±0.1/+35.71%	0.9±0.4/+39.61%	0.9±0.3/+40.40%	1.0±0.2/+38.61%	0.8±0.1/+41.96%	1.0±0.4/+36.81%
	EWC	0.9±0.2/+38.96%	0.9±0.3/+40.91%	0.9±0.1/+39.74%	1.0±0.3/+39.87%	0.9±0.2/+39.16%	1.0±0.4/+38.04%
	RanPAC	1.0±0.3/+37.66%	0.9±0.1/+42.86%	0.9±0.2/+42.38%	0.9±0.4/+40.51%	0.8±0.3/+41.96%	0.9±0.2/+42.33%
	LAMOL	0.9±0.4/+40.26%	0.9±0.2/+40.91%	0.9±0.1/+41.72%	0.9±0.3/+40.51%	0.9±0.2/+39.86%	0.9±0.3/+44.17%
	R-Tuning	0.9±0.1/+42.21%	0.8±0.2/+46.75%	0.9±0.3/+43.71%	0.9±0.4/+42.41%	0.8±0.2/+41.26%	0.9±0.1/+43.56%

Table 2: Average MAE/MSE (with error margins) on **New** tasks. The relative percentages (calculated with 3 decimal places precision) are provided after slashes. Best results are underlined. All absolute values have been scaled by a factor of 10.

Main Results

Performance Comparison To evaluate the effectiveness of R-Tuning, we compare it against six representative continual adaptation methods, including rehearsal-based, regularization-based, vanilla fine-tuning, and parameter-freezing approaches. All experiments are conducted using official pretrained time-series forecasting models.

In our experimental configuration, the five public datasets are designated as old tasks, while an in-house pedestrian flow dataset is used as the new task. Since these public datasets are frequently used in the pretraining of existing models, we assume the base models have already been extensively exposed to them. Meanwhile, to mitigate the impact of varying initial performances across base models and datasets, we calculate the average MAE and MSE over the five old tasks. This strategy ensures a more reliable assessment of knowledge retention for old tasks. Each model is then adapted to the new task using only 10% of its training set under a few-shot setting. R-Tuning further incorporates

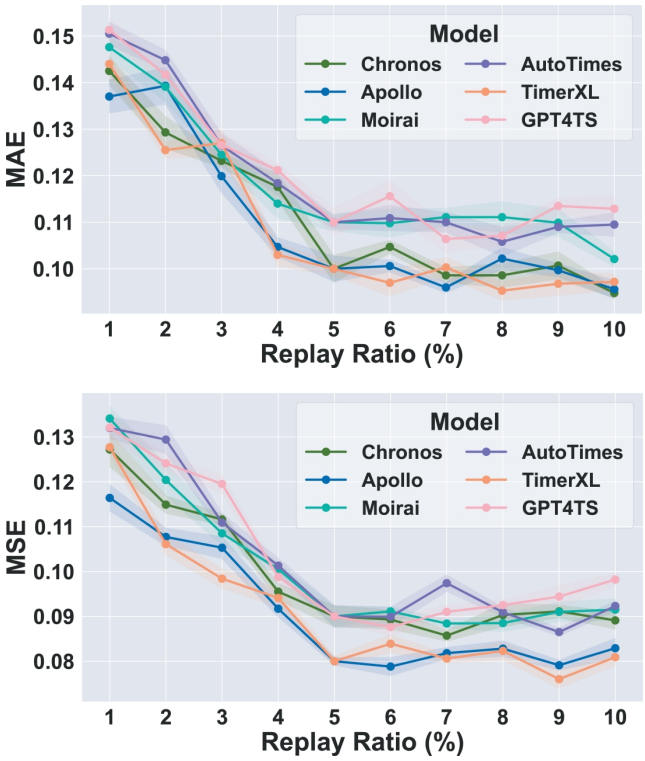

Figure 4: The performance under different proportions of the synthetic replay sample with $\tau = 3$ and $\lambda = 0.2$.

2000 synthetic replay samples via Wavelet-guided Replay and applies latent distillation with $\tau = 3$ and $\lambda = 0.2$.

As shown in Table 1 and 2, R-Tuning achieves consistent performance improvements across diverse architectures and task settings. In the best case, it reduces new-task MAE and MSE by 46.88% and 46.75%, respectively (using Apollo), while simultaneously improving old-task MAE and MSE by up to 5.70% (GPT4TS) and 5.98% (Apollo), respectively. This result underscores not only our framework's enhanced adaptability to novel tasks but also its superior retention of prior knowledge and robust stability across previous tasks. Compared with baseline methods, R-Tuning consistently outperforms alternatives, offering average gains of approximately 2–5% on old tasks and over 40% on new tasks across all models (see the Appendix for more details).

Replay Sample Efficiency To investigate the impact of the number of replay samples on continual adaptation performance, we conduct a controlled experiment that focuses on varying the quantity of replay samples. Specifically, we adjust the ratio of synthetic replay samples from 1% to 10% of the new-task training set using the Wavelet-guided Replay strategy. Except for that, all other settings are kept consistent with those in the previous section.

As illustrated in Figure 4, model performance consistently improves as the proportion of synthetic replay data increases from 1% to 5%, with the average MAE decreasing from 0.1455 to 0.1050, and the average MSE dropping from 0.1283 to 0.0867, across all evaluated models. Beyond a replay ratio of 5%, however, the performance gains plateau, with MAE and MSE stabilizing around 0.10 and 0.08, respectively. These results suggest that R-Tuning achieves near-optimal performance with as little as 4%–5% synthetic replay data, underscoring the effectiveness of its frequency-aware augmentation strategy in low-replay regimes.

	Method	Old	New
MAE	Raw	0.55 ± 0.11	0.20 ± 0.03
	Vanilla FT	0.56 ± 0.15	0.11 ± 0.04
	R-Tuning w/o Replay	0.55 ± 0.12	0.13 ± 0.01
	R-Tuning w/o Align	0.54 ± 0.10	0.11 ± 0.01
	R-Tuning	0.52 ± 0.06	0.11 ± 0.04
MSE	Raw	0.71 ± 0.08	0.16 ± 0.04
	Vanilla FT	0.77 ± 0.10	0.10 ± 0.02
	R-Tuning w/o Replay	0.71 ± 0.05	0.10 ± 0.02
	R-Tuning w/o Align	0.69 ± 0.11	0.09 ± 0.03
	R-Tuning	0.68 ± 0.09	0.09 ± 0.04

Table 3: Ablation results on AutoTimes using 2000 synthetic samples and latent distillation ($\tau = 3$, $\lambda = 0.2$).

Ablation Study To disentangle the contributions of key components in R-Tuning, we conduct an ablation study comparing four configurations: (1) fine-tuning without any continual learning methods; (2) R-Tuning without Wavelet-guided Replay; (3) R-Tuning without Semantic Alignment; and (4) full R-Tuning with both components enabled. All experiments follow the same protocol and evaluation metrics as in the *Performance Comparison* section. We use AutoTimes as the base model in the following experiments.

Table 3 reports the ablation results. For old tasks, vanilla fine-tuning increases the MAE from 0.55 ± 0.11 to 0.56 ± 0.15 and the MSE from 0.71 ± 0.08 to 0.77 ± 0.10, indicating clear forgetting. Introducing Semantic Alignment alone reduces the MAE and MSE to 0.55 ± 0.12 and 0.71 ± 0.05, while Wavelet-guided Replay alone achieves 0.54 ± 0.10 and 0.69 ± 0.11. The full R-Tuning configuration achieves the best retention, with MAE and MSE further reduced to 0.52 ± 0.06 and 0.68 ± 0.09. On new tasks, all configurations outperform the pretrained model (0.20 ± 0.03, 0.16 ± 0.04), and the full setup maintains strong adaptation performance (0.11 ± 0.04, 0.09 ± 0.04).

These results demonstrate that Semantic Alignment mitigates semantic drift by stabilizing latent representations, while Wavelet-guided Replay enriches the training distribution through frequency-aware synthesis. Their combination yields the best balance between knowledge retention and new-task adaptation.

Conclusion

In this paper, we propose R-Tuning, a replay-enhanced continual adaptation framework that integrates Wavelet-guided Replay and Semantic Alignment to balance knowledge retention and adaptation. Future work will explore online and multi-modal extensions under dynamic distribution shifts.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grants No. 72374154 and 62273261), the National Key Research and Development Program of China (Grant No. 2024YFB3312300), and the Program of China Scholarship Council (Grant No. 202506260363). The work was carried out in part while the first author, Tianyi Yin, was a Visiting PhD Student at A*STAR, Singapore.

References

Ansari, A. F.; Stella, L.; Turkmen, C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Arango, S. P.; Kapoor, S.; et al. 2024. Chronos: Learning the language of time series. arXiv:2403.07815.

Biderman, D.; Portes, J.; Ortiz, J. J. G.; Paul, M.; Greengard, P.; Jennings, C.; King, D.; Havens, S.; Chiley, V.; Frankle, J.; et al. 2024. Lora learns less and forgets less. arXiv:2405.09673.

Bonato, J.; Pelosin, F.; Sabetta, L.; and Nicolosi, A. 2024. Mind: Multi-task incremental network distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11105–11113. Vancouver, Canada: AAAI Press.

Cai, W.; Liang, Y.; Liu, X.; Feng, J.; and Wu, Y. 2024. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 11141–11149. Vancouver, Canada: AAAI Press.

Das, A.; Kong, W.; Sen, R.; and Zhou, Y. 2024. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, 10148–10167. Vienna, Austria: PMLR.

Dong, Y.; Cordonnier, J.-B.; and Loukas, A. 2021. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, 2793–2803. Virtual, Online: PMLR.

Gruver, N.; Finzi, M.; Qiu, S.; and Wilson, A. G. 2023. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36: 19622–19635.

Gu, J.; Wang, K.; Jiang, W.; and You, Y. 2024. Summarizing stream data for memory-constrained online continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12217–12225. Vancouver, Canada: AAAI Press.

Han, X.; Wang, Y.; Feng, J.; Hu, Q.; Deng, C.; et al. 2025. LOIRE: LifelOng learning on Incremental data via pre-trained language model gRowth Efficiently. In *The Thirteenth International Conference on Learning Representations*. Singapore: PMLR.

Kalajdzievski, D. 2024. Scaling laws for forgetting when fine-tuning large language models. arXiv:2401.05605.

Kim, J.; Kim, H.; Kim, H.; Lee, D.; and Yoon, S. 2025. A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges. *Artificial Intelligence Review*, 58(7): 1–95.

Kurniawan, M. R.; Song, X.; Ma, Z.; He, Y.; Gong, Y.; Qi, Y.; and Wei, X. 2024. Evolving parameterized prompt memory for continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 13301–13309. Vancouver, Canada: AAAI Press.

Li, Y.; Wu, K.; and Liu, J. 2023. Self-paced ARIMA for robust time series prediction. *Knowledge-Based Systems*, 269: 110489.

Liang, Y.; Wen, H.; Nie, Y.; Jiang, Y.; Jin, M.; Song, D.; Pan, S.; and Wen, Q. 2024. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 11, 6555–6565. New York, NY, USA: Association for Computing Machinery.

Liu, C.; Xu, Q.; Miao, H.; Yang, S.; Zhang, L.; Long, C.; Li, Z.; and Zhao, R. 2025a. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 18780–18788. Philadelphia, Pennsylvania: AAAI Press.

Liu, X.; Aksu, T.; Liu, J.; Wen, Q.; Liang, Y.; Xiong, C.; Savarese, S.; Sahoo, D.; Li, J.; and Liu, C. 2025b. Empowering Time Series Analysis with Synthetic Data: A Survey and Outlook in the Era of Foundation Models. arXiv:2503.11411.

Liu, Y.; Qin, G.; Huang, X.; Wang, J.; and Long, M. 2024a. Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems*, 37: 122154–122184.

Liu, Y.; Qin, G.; Huang, X.; Wang, J.; and Long, M. 2024b. Timer-XL: Long-Context Transformers for Unified Time Series Forecasting. arXiv:2410.04803.

Liu, Y.; Zhang, H.; Li, C.; Huang, X.; Wang, J.; and Long, M. 2024c. Timer: Generative Pre-trained Transformers Are Large Time Series Models. In *International Conference on Machine Learning*, 32369–32399. Vienna, Austria: PMLR.

McDonnell, M. D.; Gong, D.; Parvaneh, A.; Abbasnejad, E.; and Van den Hengel, A. 2023. Ranpac: Random projections and pre-trained models for continual learning. *Advances in Neural Information Processing Systems*, 36: 12022–12053.

Sun, F.-K.; Ho, C.-H.; and Lee, H.-Y. 2019. Lamol: Language modeling for lifelong language learning. arXiv:1909.03329.

Tong, R.; Liu, Y.; Shi, J. Q.; and Gong, D. 2025. Coreset selection via reducible loss in continual learning. In *The Thirteenth International Conference on Learning Representations*. Singapore: PMLR.

Wang, C.; Wang, H.; Zhang, X.; Liu, Q.; Liu, M.; and Xu, G. 2024a. A transformer-based industrial time series prediction model with multivariate dynamic embedding. *IEEE Transactions on Industrial Informatics*, 21(2): 1813–1822.

Wang, L.; Zhang, X.; Su, H.; and Zhu, J. 2024b. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8): 5362–5383.

Woo, G.; Liu, C.; Kumar, A.; Xiong, C.; Savarese, S.; and Sahoo, D. 2024. Unified Training of Universal Time Series Forecasting Transformers. In *International Conference on Machine Learning*, 53140–53164. Vienna, Austria: PMLR.

Xu, Z.; Jiang, F.; Niu, L.; Deng, Y.; Poovendran, R.; Choi, Y.; and Lin, B. Y. 2024. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. arXiv:2406.08464.

Yin, T.; Wang, J.; Ma, Y.; Wang, H.; Wang, C.; Zhao, Y.; Liu, M.; and Shen, W. 2025. Apollo-Forecast: Overcoming Aliasing and Inference Speed Challenges in Language Models for Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 22173–22181. Philadelphia, Pennsylvania: AAAI Press.

Zhang, X.; Chowdhury, R. R.; Gupta, R. K.; and Shang, J. 2024. Large language models for time series: a survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 8335–8343. Jeju, South Korea: IJCAI Organization.

Zhou, D.-W.; Sun, H.-L.; Ning, J.; Ye, H.-J.; and Zhan, D.-C. 2024a. Continual learning with pre-trained models: a survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 8363–8371. Jeju, South Korea: IJCAI Organization.

Zhou, D.-W.; Wang, Q.-W.; Qi, Z.-H.; Ye, H.-J.; Zhan, D.-C.; and Liu, Z. 2024b. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 9851–9873.

Zhou, T.; Niu, P.; Sun, L.; Jin, R.; et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in Neural Information Processing Systems*, 36: 43322–43355.