ReCast: Reliability-aware Codebook Assisted Lightweight Time Series Forecasting

Xiang Ma[1], **Taihua Chen**[1,2], **Pengcheng Wang**[1], **Xuemei Li**[1], **Caiming Zhang**[1]

[1]Shandong University, Jinan, China
[2]Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, Jinan, China
{xiangma,xmli,czhang}@sdu.edu.cn, {cfair-cth,pengchengwang}@mail.sdu.edu.cn

Abstract

Time series forecasting is crucial for applications in various domains. Conventional methods often rely on global decomposition into trend, seasonal, and residual components, which become ineffective for real-world series dominated by local, complex, and highly dynamic patterns. Moreover, the high model complexity of such approaches limits their applicability in real-time or resource-constrained environments. In this work, we propose a novel **RE**liability-aware **C**odebook-**AS**sisted **T**ime series forecasting framework (**ReCast**) that enables lightweight and robust prediction by exploiting recurring local shapes. ReCast encodes local patterns into discrete embeddings through patch-wise quantization using a learnable codebook, thereby compactly capturing stable regular structures. To compensate for residual variations not preserved by quantization, ReCast employs a dual-path architecture comprising a quantization path for efficient modeling of regular structures and a residual path for reconstructing irregular fluctuations. A central contribution of ReCast is a reliability-aware codebook update strategy, which incrementally refines the codebook via weighted corrections. These correction weights are derived by fusing multiple reliability factors from complementary perspectives by a distributionally robust optimization (DRO) scheme, ensuring adaptability to non-stationarity and robustness to distribution shifts. Extensive experiments demonstrate that ReCast outperforms state-of-the-art (SOTA) models in accuracy, efficiency, and adaptability to distribution shifts.

1 Introduction

In recent years, time series forecasting has gained significant attention due to its critical applications in various real-world applications, including finance, energy, healthcare, and industrial automation (Wen et al. 2023; Ma et al. 2023; Qiu et al. 2024; Shibo et al. 2025). Capturing complex and irregular temporal patterns accurately remains a primary challenge in this domain. Conventional approaches typically address this complexity by globally decomposing time series into trend, seasonal, and residual components, and modeling these components independently (Wu et al. 2021; Zhou et al. 2022; Hu et al. 2025). However, while effective for structured or periodic data, such global decomposition methods often underperform when faced with dynamic and noisy real-world time series (Tang and Zhang 2025). Moreover, these methods typically involve considerable model complexity, which limits their practicality in resource-limited environments (Ansari et al. 2025).

To address these challenges, we introduce a novel **RE**liability-aware **C**odebook-**AS**sisted **T**ime series forecasting (ReCast) framework, focusing explicitly on capturing local patterns. Observing that many real-world series exhibit recurring local shapes rather than clear global regularities (Yeh et al. 2016), ReCast quantizes these local shapes into a learnable codebook, generating discrete embeddings to represent evolving patterns. This codebook-based representation not only captures salient local structures but also reduces model complexity, enabling a inherently lightweight forecasting design. Meanwhile, residual modeling is introduced to capture irregular variations not adequately represented by the quantized embeddings, ensuring robustness to fluctuations without excessively increasing model size.

Specifically, ReCast segments input into patches, quantifying each as discrete embedding using a dynamically updated reliability-aware codebook. As shown in Figure 1, a quantization path is used to forecast the future discrete embeddings, and a residual path learns to estimate the difference between input and its approximate representation reconstructed by discrete embedding. These two paths work in synergy: the quantization path enables lightweight forecasting of regular structures, while the residual path ensures the reliable reconstruction of irregular fluctuations. The prediction results combine outputs from both paths. To reduce overfitting and improve generalization to distribution shifts, we perform random patch sampling, and select only a subset of patches for training and codebook updates. Downsampling is applied prior to quantization, helping to highlight salient local structures and lower computational cost.

More importantly, it can be observed that the performance of ReCast strongly depends on the stability and adaptability of the codebook (Guo et al. 2023). Therefore, we propose an incremental codebook update mechanism centered on a reliability-aware scoring method. This method can robustly guide the update process in response to evolving data distributions, striking a balance between stability and adaptability. Our **contributions** include:

- We propose a codebook assisted lightweight forecasting framework that effectively captures both regular and ir-

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

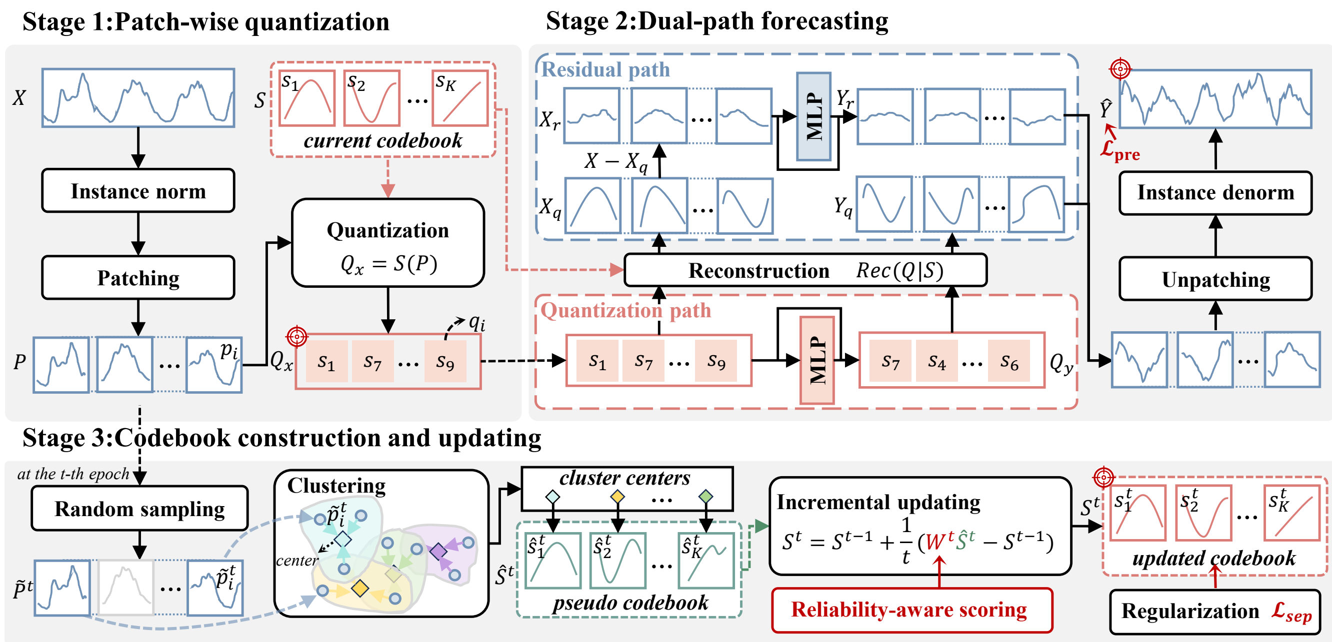

Figure 1: ReCast overview. It comprises patch-wise quantization, dual-path forecasting, codebook construction and updating.

regular local temporal patterns while significantly reducing model complexity.

- We introduce a reliability-aware updating mechanism for codebook, which enhances adaptability and robustness to noise and distribution shifts with low computational cost.
- Extensive experiments show that ReCast achieves superior accuracy, generalization, and robustness, relying on its lightweight architecture and efficient training strategy.

2 Related Work

2.1 Deep Learning-based Forecasting

Recent advances in deep learning have significantly improved time series forecasting by leveraging powerful representation learning capabilities. The convolutional neural network(CNN)-based approaches (Wu et al. 2023) introduce local receptive fields to capture short-term dynamics efficiently, but they lack the ability to capture long-range dependencies. Transformer-based models (Liu et al. 2023; Nie et al. 2023) address this issue by employing self-attention to model global temporal interactions, achieving strong performance across benchmarks. Nonetheless, the quadratic complexity of attention and sensitivity to noise restrict their scalability and robustness in real-world scenarios. In parallel, lightweight MLP-based architectures (Tang and Zhang 2025; Ma et al. 2024) have recently emerged as promising alternatives, offering high efficiency but often struggling to represent heterogeneous and irregular patterns effectively.

2.2 Patch-based Representation Learning

To improve efficiency and capture fine-grained structures, patch-based strategies have gained increasing attention in time series modeling. Instead of processing sequences at raw temporal resolution, some methods (Wang et al. 2025; Nie et al. 2023; Tang and Zhang 2025) divide time series into non-overlapping or partially overlapping patches, enabling models to operate on compact representations and reduce sequence length. While effective in long-horizon forecasting, these methods typically rely on continuous embeddings without explicit mechanisms to leverage recurring local shapes, which are prevalent in real-world time series. Vector quantization (VQ) (Van Den Oord, Vinyals et al. 2017) provides a complementary perspective by discretizing local segments into a finite set of codewords, facilitating representation reuse and improving robustness, as extensively explored in domains such as vision and speech (Tian et al. 2024; Wu et al. 2025). Recent attempts (Shibo et al. 2025; Ansari et al. 2025) to integrate quantization into time series tasks demonstrate its potential to capture recurring patterns efficiently. However, static or heuristic codebooks fail to adapt to real-world data dynamics.

Different from existing methods, ReCast innovatively propose a dual-path forecasting architecture with quantization, capturing both stable recurring shapes and irregular fluctuations. Besides, it introduces a reliability-aware updating incrementally refines codebook, ensuring robust adaptation to distribution shifts.

3 Methodology

In this section, we present ReCast in detail, which has 3 modules: patch-wise quantization, dual-path forecasting, codebook construction and updating, as shown in Figure 1.

3.1 Patch-wise Quantization

Define the historical series as $\mathbf{X} \in \mathbb{R}^{C\times L} = \{\mathbf{x}_i\}_{i=1}^{L}$, and the ground truth future values as $\mathbf{Y} \in \mathbb{R}^{C\times H} = \{\mathbf{x}_i\}_{i=L+1}^{L+H}$.

L and H are the length of the input and forecasting series, respectively. C means the number of variables (or channels). $\mathbf{x}_i$ is a vector of dimension C at time step i. The goal of time series forecasting is to predict $\mathbf{Y}$ based on observed $\mathbf{X}$. ReCast first normalizes the input using instance normalization, which is $\mathbf{X} = (\mathbf{X} - \mu_{in})/\sqrt{\sigma_{in}^2 + \varepsilon}$. μ_{in} and σ_{in} denote the mean and variance of input, and ε is a small constant added for numerical stability. The normalized $\mathbf{X}$ is segmented into patches $\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^{C\times N}$. $\mathbf{p}_i \in \mathbb{R}^{L_p}$ is the i-th patch. L_p is the patch length, and $N = \lceil L/L_p \rceil$.

Each patch is subsequently quantized by assigning it to the nearest codeword in a learnable codebook $\mathbf{S} = \{\mathbf{s}_k\}_{k=1}^{K}$:

$$\begin{aligned} q_i &= \mathbf{S}(\tilde{\mathbf{p}}_i) = \underset{\mathbf{s}_k \in \mathbf{S}}{\arg\min} ||\tilde{\mathbf{p}}_i - \mathbf{s}_k||_2^2, \\ \tilde{\mathbf{p}}_i &= Dsamp(\mathbf{p}_i), \ \ \mathbf{s}_k, \tilde{\mathbf{p}}_i \in \mathbb{R}^{L_p/2} \end{aligned} \tag{1}$$

where $q_i \in \{1, \cdots, K\}$ is the discrete index associated with patch $\mathbf{p}_i$. K is the number of codewords. To reduce computational cost and suppress redundant local fluctuations, we apply downsampling $Dsamp(\cdot)$ on patches prior to quantization. This is supported by the well-established assumption in time series modeling that local patterns demonstrate invariance across scales and redundant morphology (Lu et al. 2022), which makes resolution reduction both meaningful and robust (Senin and Malinchik 2013). $\tilde{\mathbf{p}}_i$ is the downsampled patch of $\mathbf{p}_i$. This operation reduces the dimension of patches to $L_p/2$, significant savings in codebook matching, storage, and embedding projection. Besides, it helps the codebook focus on salient structures, improving robustness and generalization to noisy or distribution shifts. The discrete embeddings for the full input series is organized as $\mathbf{Q}_x = [\mathbf{Q}_1; \cdots; \mathbf{Q}_C]$, and $\mathbf{Q}_i = \{q_j\}_{j=(i-1)\cdot N+1}^{i\cdot N}$ is the discrete embedding of i-th variable. This discrete embeddings serves as the input to downstream forecasting modules.

3.2 Dual-path Forecasting

To simultaneously achieve computational efficiency and representational fidelity, ReCast adopts a dual-path forecasting architecture. This design decomposes the prediction task into two complementary paths, each responsible for capturing distinct aspects of temporal dynamics.

Quantization path To capturing the regular structures and modeling the evolution of local patterns, a lightweight multilayer perceptron (MLP) $\mathcal{M}_{\text{quant}}$ is employed to forecast the discrete indices of future patches:

$$\mathbf{Q}_y = \mathcal{M}_{\text{quant}}(\mathbf{Q}_x), \tag{2}$$

where $\mathbf{Q}_y \in \mathbb{R}^{C\times N_y}$, and $N_y = \lceil H/L_p \rceil$. This path enables compact and efficient modeling of stable local patterns.

Residual path While quantization promotes simplicity, it inevitably discards subtle variations. To mitigate this loss, ReCast introduces a residual correction branch. First, the input $\mathbf{X}$ is approximately reconstructed from its quantized representation via codebook lookup:

$$\begin{aligned} &\mathbf{X}_q = Rec(\mathbf{Q}_x|\mathbf{S}) = Rec(\mathbf{Q}_1; \cdots; \mathbf{Q}_C|\mathbf{S}), \\ &Rec(\mathbf{Q}_i|\mathbf{S}) = Usamp([\mathbf{s}_{q_{(i-1)\cdot N}}|| \cdots ||\mathbf{s}_{q_{i\cdot N}}]), \end{aligned} \tag{3}$$

where $\mathbf{X}_q \in \mathbb{R}^{C\times L}$ denotes the approximate representation of $\mathbf{X}$. $Rec(\mathbf{Q}_i|\mathbf{S})$ means reconstruction from discrete embedding $\mathbf{Q}_i$ using the codebook $\mathbf{S}$. $Usamp(\cdot)$ denotes the upsampling. $||$ denotes the concatenation. The residual component $\mathbf{X}_r = \mathbf{X} - \mathbf{X}_q$ captures fine-scale discrepancies. A separate MLP forecaster $\mathcal{M}_{\text{res}}$ is trained to predict the residual signal for the future window:

$$\mathbf{Y}_r = \mathcal{M}_{\text{res}}(\mathbf{X}_r), \ \ \mathbf{Y}_r \in \mathbb{R}^{C\times H}, \tag{4}$$

The final result combines both paths and is followed by instance denormalization to restore the original scale

$$\begin{aligned} &\hat{\mathbf{Y}} = \sigma_{in}(\mathbf{Y}_q + \mathbf{Y}_r) + \mu_{in}, \ \ \hat{\mathbf{Y}} \in \mathbb{R}^{C\times H}, \\ &\mathcal{L}_{pre} = ||\hat{\mathbf{Y}} - \mathbf{Y}||_1, \end{aligned} \tag{5}$$

where $\mathbf{Y}_q = Rec(\mathbf{Q}_y|\mathbf{S})$. To mitigate the distribution shift effect between the input X and forecasting result, we use instance denormalization by σ_{in} and μ_{in}. We employ the L_1 Loss as the training objective to ensure robustness to outliers and stabilizes training.

3.3 Codebook Construction and Updating

The performance and robustness of ReCast are tightly coupled with the quality of its quantization codebook. Since real-world time series are often non-stationary and subject to distribution shifts, a static codebook is insufficient for capturing evolving local patterns. So, we adopt an incremental updating strategy for codebook construction, which allows the model to gradually refine its representation of local patterns based on data observed over time, as shown in stage 3 of Figure 1. This approach can enable adaptation to evolving distributions, and avoid the instability and overfitting associated with outliers.

Pseudo codebook construction At each epoch, we cluster the patches and obtain cluster centers. These centers are the representative local patterns that can be used to construct pseudo codebooks in the current epoch. The clustering objects are randomly sampled patches from the input. This random sampling reduces computational cost and prevents overfitting (Lu et al. 2022; Senin and Malinchik 2013). To ensure the efficiency, we express the energy function $\mathcal{L}_c$ of clustering in the form of matrix operation:

$$\mathcal{L}_c = Tr((\tilde{\mathbf{P}}^t - \mathbf{M}\hat{\mathbf{S}}^t)^\top I(\tilde{\mathbf{P}}^t - \mathbf{M}\hat{\mathbf{S}}^t)), \tag{6}$$

where $\hat{\mathbf{S}}^t = \{\hat{\mathbf{s}}_k^t\}_{k=1}^{K}$ and $\tilde{\mathbf{P}}^t = \{\tilde{\mathbf{p}}_i^t\}_{i=1}^{C\times N_p}$ denote the cluster center matrix and the sampled downsampled patches at t-th epoch, respectively. $\tilde{\mathbf{p}}_i^t \in \mathbb{R}^{L_p/2}$ is the i-th patch of $\tilde{\mathbf{P}}^t$, and $\hat{\mathbf{s}}_k^t \in \mathbb{R}^{L_p/2}$ is the k-th cluster center of $\hat{\mathbf{S}}^t$. N_p is the number of sampled patches. I is the weight matrix, here we take the identity matrix. $Tr(\cdot)$ is the trace of matrix. $\mathbf{M} \in \mathbb{R}^{(C\times N_p)\times K}$ is the indicator matrix to indicate the membership of patches, which is a learnable binary matrix. $M_{i,j} = 1$ means the patch i belong to cluster j. The update function of cluster center is:

$$\hat{\mathbf{S}}^t = (\mathbf{M}^\top I\tilde{\mathbf{P}}^t)/(\mathbf{M}^\top I\mathbf{M}), \tag{7}$$

the $\hat{\mathbf{S}}^t$ reflects representative local patterns captured from the current training data distribution, which can serve as the pseudo codebook of t-th epoch.

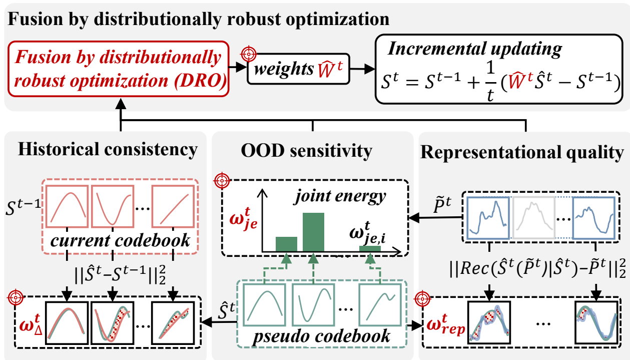

Figure 2: Illustration of reliability-aware scoring, showing three scoring factors and their fusion via distributionally robust optimization (DRO).

Incremental updating To ensure generalization to new patterns or distribution shifts, and avoid drastic changes of embeddings, we introduce a incremental updating strategy for codebook to balance adaptability and stability. At the first epoch, we initialize the codebook as $\mathbf{S}^1 = \hat{\mathbf{S}}^1$. In subsequent epochs, the codebook is updated as:

$$\mathbf{S}^t = \mathbf{S}^{t-1} + \frac{1}{t}(\mathbf{W}^t\hat{\mathbf{S}}^t - \mathbf{S}^{t-1}), \tag{8}$$

where $\mathbf{S}^t$ denotes the codebook of t-th epoch. $\hat{\mathbf{S}}^t$ is the pseudo codebook computed from the current epoch's sampled patches via Equation 7. $\mathbf{W}^t$ is a set of correction weights that adjust the influence of the current epoch's pseudo codebook. W^t_k is the weight for cluster center $\hat{s}^t_k$. Equation 8 can ensure equitable contribution across epochs, while adaptively adjusting by $\mathbf{W}^t$ (See **Appendix A.1** in the extended version for complete proof).

Embedding regularization To promote better utilization of the embedding space and prevent codeword collapse, we introduce a limited separation loss that encourages diversity among the cluster centers:

$$\mathcal{L}_{sep} = log \sum_{i,j=1}^{k} exp(-||\hat{\mathbf{s}}^t_i - \hat{\mathbf{s}}^t_j||_2^2)/\tau, \tag{9}$$

where $\mathcal{L}_{sep}$ promotes the dispersion of embeddings in hidden space and prevents excessive expansion of the space by the temperature τ. $\tau = ||\hat{\mathbf{S}}^t||_2^2$ ensures the embedding space size remains approximately consistent across each epoch. This loss penalizes excessive similarity among codewords, encouraging a well-distributed and expressive codebook.

3.4 Reliability-aware Scoring

As shown in Equation 8, $\mathbf{W}^t$ can control the contribution of each pseudo codeword during updating. Rather than treating all cluster centers equally, ReCast introduces a reliability-aware scoring method that selectively integrates pseudo codewords based on their reliability to ensure robust and adaptive codebook updates. $\mathbf{W}^t = \{w^t_k\}^K_{k=1}$ is computed by aggregating three complementary factors: $\mathbf{w}^t_{rep}, \mathbf{w}^t_{\Delta}, \mathbf{w}^t_{je} \in \mathbb{R}^K$, and meets $\mathbf{W}^t \propto \mathcal{M}_{fus}(\mathbf{w}^t_{rep}, \mathbf{w}^t_{\Delta}, \mathbf{w}^t_{je})$, $\sum^K_{k=1} w^t_k = 1$. $\mathcal{M}_{fus}$ is a fusion function.

Representational quality The $\mathbf{w}^t_{rep}$ evaluates how well $\hat{\mathbf{S}}^t_k$ represents its assigned patches, measured by the intra-cluster reconstruction error:

$$w^t_{rep,k} = 1 - \frac{exp(||\mathbf{B}_k(Rec(\hat{\mathbf{S}}^t(\tilde{\mathbf{P}}^t)|\hat{\mathbf{S}}^t) - \tilde{\mathbf{P}}^t)||_2^2)}{exp(||Rec(\hat{\mathbf{S}}^t(\tilde{\mathbf{P}}^t)|\hat{\mathbf{S}}^t) - \tilde{\mathbf{P}}^t||_2^2) + \varepsilon}, \tag{10}$$

where $w^t_{rep,k} \in \mathbf{w}^t_{rep}$ is the weight assigned to the k-th cluster center. $\mathbf{B}_k$ is a binary matrix to mask values unrelated to the k-th cluster center. $Rec(\hat{\mathbf{S}}^t(\tilde{\mathbf{P}}^t)|\hat{\mathbf{S}}^t)$ is the approximate representation reconstructed from discrete embeddings $\hat{\mathbf{S}}^t(\tilde{\mathbf{P}}^t)$ using pseudo codebook $\hat{\mathbf{S}}^t$. Higher value of $w^t_{rep,k}$ corresponds to better representational quality of the k-th cluster center, which has higher reliability.

Historical consistency The $\mathbf{w}^t_{\Delta}$ assesses the temporal stability of $\hat{\mathbf{S}}^t_k$ by measuring its deviation from the corresponding codeword in the previous epoch:

$$w^t_{\Delta,k} = \frac{exp(||\mathbf{B}_k(\hat{\mathbf{S}}^t - \mathbf{S}^{t-1})||_2^2)}{exp(||\hat{\mathbf{S}}^t - \mathbf{S}^{t-1}||_2^2) + \varepsilon}, \tag{11}$$

where $w^t_{\Delta,k} \in \mathbf{w}^t_{\Delta}$ is the weight assigned to the k-th cluster center. Higher value of $w^t_{\Delta,k}$ denotes the greater difference between $\hat{\mathbf{s}}^t$ and $\mathbf{s}^{t-1}$. Under the constraint of $\mathbf{w}^t_{rep}$, this difference arises because $\mathbf{S}^{t-1}$ lacks sufficient fitting capability for the newly input patches. So $\hat{\mathbf{S}}^t$ should be given a greater weight to adjust the previous codebook, which is consistent with the expression of $\mathbf{w}^t_{\Delta,k}$.

OOD sensitivity The $\mathbf{w}^t_{je}$ measures the OOD sensitivity of $\hat{\mathbf{S}}^t$ by capturing potentially novel or rare patterns, estimated from assignment frequency and variance. The function is similar to joint-energy (Duvenaud et al. 2020):

$$w^t_{je,k} = 1 - \frac{exp(\sum^{C\times N_p}_{i=1} |\tilde{\mathbf{p}}^t_i - \hat{\mathbf{s}}^t_k|)}{exp(\sum^K_{k=1}\sum^{C\times N_p}_{i=1} |\tilde{\mathbf{p}}^t_i - \hat{\mathbf{s}}^t_k|) + \varepsilon}, \tag{12}$$

where $w^t_{je,k} \in \mathbf{w}^t_{je}$ is the weight assigned to the k-th cluster center. Higher value of $w^t_{je,k}$ indicates lower selection probabilities for the k-th cluster. By increasing its corresponding weight, we can prevent the embedding space of the codebook from collapsing into a few fixed codewords, and evaluate adaptability to OOD data (Duvenaud et al. 2020).

Fusion by distributionally robust optimization In ReCast, each pseudo codeword is associated with three normalized reliability scores: $\mathbf{w}^t_{rep}$, $\mathbf{w}^t_{\Delta}$, and $\mathbf{w}^t_{je}$. While these metrics are complementary, their relative importance may vary across epochs and data regimes. Directly assigning fixed weights can be suboptimal or unstable, especially when some metrics are noisy or biased due to transient data conditions (Duchi and Namkoong 2019). Thus, we formulate

Models	ReCast		PatchMLP		TQNet		CycleNet		iTransformer		TimesNet		PatchTST		Dlinear	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.371**	**0.379**	0.374	0.382	0.377	0.393	0.379	0.396	0.407	0.410	0.400	0.406	0.387	0.400	0.403	0.407
ETTm2	**0.265**	**0.309**	0.269	0.311	0.277	0.323	0.266	0.314	0.286	0.327	0.291	0.333	0.281	0.326	0.350	0.401
ETTh1	**0.437**	**0.428**	0.438	0.429	0.441	0.434	0.457	0.441	0.454	0.447	0.458	0.450	0.469	0.454	0.456	0.452
ETTh2	**0.347**	0.385	0.349	**0.378**	0.378	0.402	0.388	0.409	0.383	0.407	0.414	0.427	0.387	0.407	0.559	0.515
ECL	**0.163**	**0.257**	0.171	0.265	0.164	0.259	0.168	0.259	0.178	0.270	0.192	0.295	0.216	0.304	0.212	0.300
Traffic	0.418	**0.272**	**0.417**	0.273	0.445	0.276	0.472	0.301	0.428	0.282	0.620	0.336	0.555	0.362	0.625	0.383
Weather	**0.229**	**0.250**	0.231	0.256	0.242	0.269	0.243	0.271	0.258	0.279	0.259	0.287	0.259	0.281	0.265	0.317
Solar	0.209	0.260	0.211	0.261	**0.198**	**0.256**	0.210	0.261	0.233	0.262	0.319	0.330	0.307	0.641	0.401	0.282
1^{st} Count	**12**		2		2		0		0		0		0		0	

Table 1: Performance Comparison. The best results are highlighted in **bold**, while the second-best results are underlined.

the fusion of reliability metrics as a distributionally robust optimization (DRO) problem (Qi et al. 2021). The goal is to obtain a conservative estimate of a codeword's reliability by minimizing the expected reliability under the worst-case weighting distribution over the three metrics.

Formally, let the score vector for the k-th pseudo codeword at epoch t be denoted as:

$$\mathbf{z}_k^t = [w_{rep,k}^t, w_{\Delta,k}^t, w_{je,k}^t] \in \mathbb{R}^3. \tag{13}$$

Instead of computing a simple average, we consider all possible distributions $\theta \in \Theta_3$ over the three scores, where $\Theta_3 = \{\theta \in \mathbb{R}^3 \mid \sum_{i=1}^3 \theta_i = 1, \theta_i \geq 0\}$. We then define the reliability score $\mathbf{w}_k^t$ as the minimum expected value of $\mathbf{z}_k^t$ under the worst-case distribution within a KL-divergence neighborhood around the uniform distribution $\mathbf{u} = [1/3, 1/3, 1/3]$:

$$\hat{w}_k^t = \min_{\theta \in \mathcal{U}_\gamma} \langle \theta, \mathbf{z}_k^t \rangle, \tag{14}$$

where $\mathcal{U}_\gamma = \{\theta \in \Theta_3 \,|\, \mathcal{D}_{KL}(\theta \,\|\, \mathbf{u}) \leq \gamma\}$. The parameter $\gamma > 0$ determines the size of the uncertainty set: smaller values encourage near-uniform weighting, while larger values permit more skewed, adversarial distributions. This robust optimization problem has a closed-form solution (See **Appendix A.2** in the extended version for complete proof):

$$\hat{w}_k^t = -\gamma \cdot \log \sum_{i=1}^{3} \exp(-\frac{\mathbf{z}_{k,i}^t}{\gamma}). \tag{15}$$

The result is a soft-minimum over the scores, allowing the most reliable metric to dominate while softly discounting others. This formulation can be interpreted as an entropy-regularized minimization over reliability signals.

By adopting this distributionally robust fusion scheme, ReCast is able to adaptively and conservatively score pseudo codewords, mitigating the impact of outliers or transient inconsistencies in individual metrics. This not only enhances the stability of the incremental codebook update but also improves the generalization of non-stationary time series.

Finally, the reliability score $\hat{\mathbf{W}}^t = \{\hat{w}_k^t\}_{k=1}^K$ is used as a weighting coefficient to regulate the effect intensity of pseudo codewords in the codebook update. The Equation 8 can been improved as:

$$\mathbf{S}^t = \mathbf{S}^{t-1} + \frac{1}{t}(\hat{\mathbf{W}}^t \hat{\mathbf{S}}^t - \mathbf{S}^{t-1}). \tag{16}$$

3.5 Learning Objective

The final loss function is:

$$\mathcal{L} = \mathcal{L}_{pre} + w_{sep}\mathcal{L}_{sep}, \tag{17}$$

where w_{sep} is adjustment parameters. During the inference phase, the codebook remains fixed, and only Equation 5 needs to be computed to efficiently obtain prediction results in a lightweight manner.

4 Experiments

4.1 Datasets and Baselines

We evaluate the proposed ReCast on 8 widely used real-world datasets: Electricity (ECL), Traffic, Weather, Solar (Liu et al. 2022; Wu et al. 2021), and 4 ETT datasets (ETTh1, ETTh2, ETTm1, ETTm2) (Zhou et al. 2021). To evaluate the performance of ReCast, we compare it against 8 representative SOTA models from recent years: Transformer-based models: TQNet (Lin et al. 2025), iTransformer (Liu et al. 2023), PatchTST (Nie et al. 2023); CNN-based model: TimeNet (Wu et al. 2023); MLP-based models: PatchMLP (Kong et al. 2025), CycleNet (Lin et al. 2024), DLinear (Zeng et al. 2023).

4.2 Metrics and Implementation Details

The models are evaluated based on both Mean Squared Error (MSE) and Mean Absolute Error (MAE). ReCast is implemented using Pytorch (Paszke et al. 2019) and trained on an Nvidia L40 GPU (48GB). The detailed implementations and the corresponding pseudocode of ReCast are provided in the extended version.

4.3 Main Results

Table 1 compares the forecasting performance of ReCast with baselines across 8 datasets, with lower MSE/MAE values indicating greater forecasting accuracy. ReCast achieves

Setup	Original		-Residual		-Updating		-Random		-Scoring		-DRO	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	**0.371**	**0.379**	0.377	0.395	0.400	0.402	0.377	0.396	0.385	0.399	0.375	0.385
Traffic	**0.418**	**0.272**	0.435	0.281	0.553	0.332	0.427	0.285	0.441	0.285	0.424	0.281
Weather	**0.229**	**0.250**	0.248	0.275	0.257	0.303	0.240	0.271	0.249	0.277	0.237	0.266

Table 2: Ablation study of ReCast.

Model	iTransformer				TimesNet			
	Original		+ReCast		Original		+ReCast	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTm1	0.407	0.410	**0.375**	**0.381**	0.400	0.406	**0.389**	**0.395**
Traffic	0.428	0.282	**0.420**	**0.275**	0.620	0.336	**0.499**	**0.303**
Weather	0.258	0.279	**0.231**	**0.259**	0.259	0.287	**0.245**	**0.272**

Table 3: Portability of ReCast across different backbones.

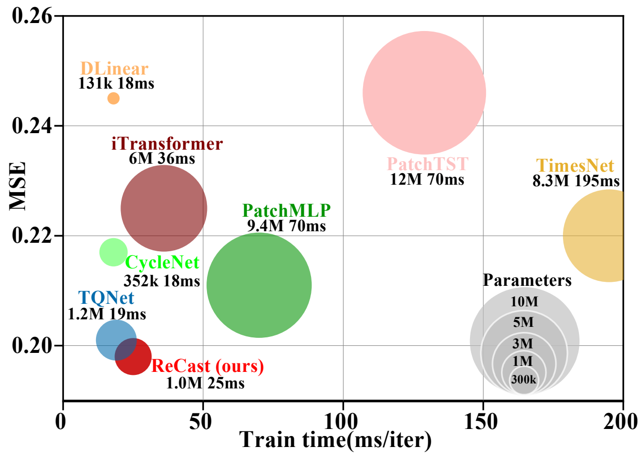

Figure 3: Computational efficiency of ReCast on the ECL dataset (horizon = 720).

the best performance in 12 out of 16 forecasting error metrics, demonstrating overall SOTA accuracy.

Notably, CNN-based models no longer retain a performance advantage due to their limited capacity in modeling long-range dependencies. Transformer-based models lies in modeling temporal contextual dependencies through attention mechanisms, which exhibit high sensitivity to noise. This inherent sensitivity limits their potential for further improving predictive performance. While they occasionally outperform simple MLP-based models, their performance is inconsistent, especially in noisy or irregular settings. Recent lightweight MLP-based models offer improved efficiency, but some of them often struggle to capture intricate inter-variable dependencies.

Moreover, channel-independent models (PatchTST and DLinear) often fail to realize their full potential, suggesting the irreplaceable role of inter-variable interactions. In contrast, ReCast employs a shared codebook across all variables, implicitly facilitating inter-variable interaction and thereby circumventing the performance limitations inherent in channel-independent architectures.

4.4 Model Analysis

Ablation Study Four variants are designed to assess the contributions of ReCast's core components:'**-Residual**' disables the residual path, retaining only the quantization path; '**-Updating**'freezes the codebook, preventing incremental updates; '**-Random**'removes both downsampling during quantization and random sampling during codebook construction; '**-Scoring**' disables the reliability-aware fusion weights $\hat{W}_t$ in Equation 16, treating all pseudo codewords equally during codebook updates; '**-DRO**' uniformly weights the three scores. These variants can systematically evaluate the effects of dual-path architecture, robust enhancement operation, incremental updating, reliability-aware scoring, and DRO on model performance.

The results of Table 2 lead to several key observations: 1) All ablated variants exhibit degraded performance relative to the full ReCast model, validating the effectiveness of each component. 2) The performance drop in '**-Residual**' highlights the critical role of the residual path in recovering fine-grained variations that are lost during quantization. 3)The performance deterioration in '**-Updating**' and '**-Scoring**' confirms that both dynamic codebook refinement and reliability-aware weighting are essential for capturing evolving local patterns and ensuring adaptability to distribution shifts. 4) The degradation observed in '**-Random**' underscores the importance of downsampling and random sampling for reducing overfitting and computational cost, while preserving performance. 5) The gap between '**-Scoring**' and '**-DRO**' reveals the importance of the DRO-based fusion strategy, which avoids over-reliance on any single score and enables robust reliability estimation.

Portability To evaluate the portability of ReCast, we examine whether its dual-path forecasting architecture and reliability-aware codebook mechanism can generalize beyond the original MLP-based backbone. Specifically, we replace the MLP backbone with two widely used backbones: iTransformer, representative of Transformer-based methods, and TimesNet, representative of CNN-based methods. As reported on Table 3, integrating ReCast's dual-path framework with either iTransformer or TimesNet improves forecasting performance. These results demonstrate that the proposed architecture is not tightly coupled with any specific backbone type and can be seamlessly adapted to a broad range of

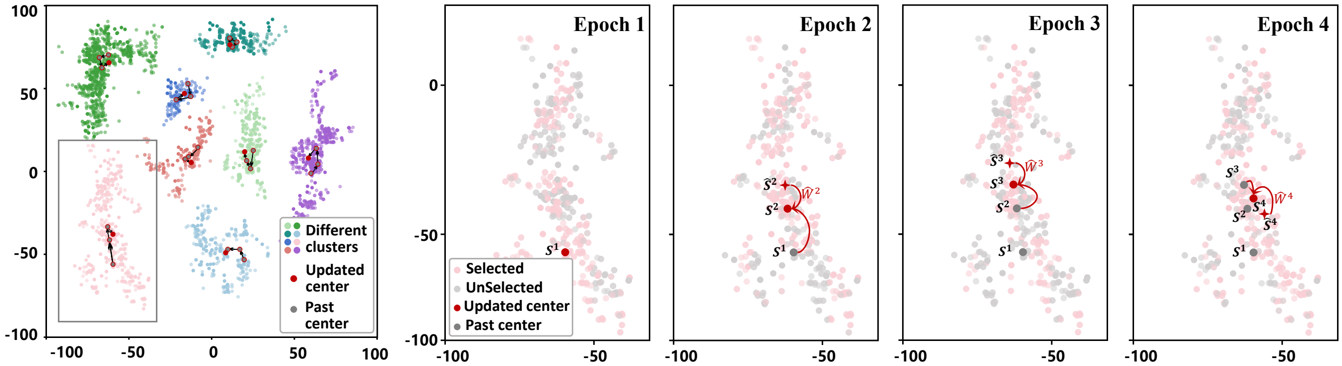

Figure 4: Visualization of codebook evolution and cluster assignments across epochs.

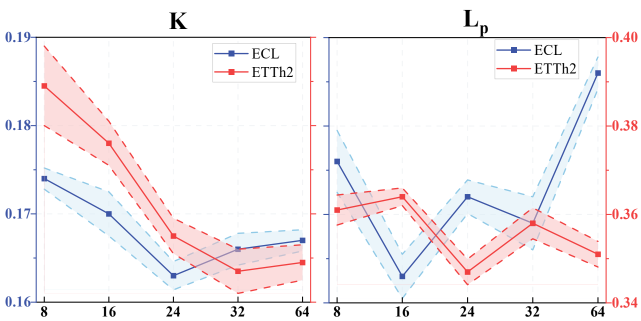

Figure 5: Parameter sensitivity.

forecasting models, thereby confirming its strong portability and general applicability.

Efficiency Benefiting from its lightweight dual-path architecture and a series of efficiency-oriented design choices, such as patch-wise quantization, residual correction, and selective sampling, ReCast achieves state-of-the-art forecasting accuracy while maintaining low computational overhead. As illustrated in Figure 3, ReCast consistently ranks among the top-performing models in terms of both parameter efficiency and training speed, without compromising predictive performance. These results highlight ReCast's ability to strike a favorable balance between forecasting accuracy and computational complexity, making it well-suited for deployment in resource-constrained environments.

Parameter sensitivity Figure 5 shows the performance under different hyperparameters (the number of clusters (codewords) K and the patch length L_p).

4.5 Visualization

ReCast performs patch-wise clustering to generate discrete embeddings, its forecasting accuracy hinges on clustering quality and the representational capacity of the resulting cluster centers (codewords). To intuitively illustrate the codebook construction and update process, Figure 4 presents qualitative visualizations. The left side of Figure 4 shows clustering results over 8 clusters and the evolution of cluster centers across epochs, where each color denotes a distinct cluster. Despite random sampling, cluster assignments remain stable and centers converge smoothly, demonstrating the robustness of the clustering. The right side of Figure 4 illustrates the temporal dynamics of codebook updates. Taking epoch 2 as an example, the pseudo codebook $\hat{\mathbf{S}}^2$ better fits the current data distribution than $\mathbf{S}^1$, and the reliability-aware update assigns higher weight to $\hat{\mathbf{S}}^2$, shifting $\mathbf{S}^2$ closer to it. This confirms that the proposed reliability-aware update mechanism effectively balances adaptation and stability, supporting robust and generalizable forecasting.

4.6 Limitations

Despite its demonstrated accuracy and efficiency, ReCast presents a notable practical limitation: As shown in Figure 5, its performance is sensitive to the choice of K and L_p. These parameters influence the trade-off between representational granularity and generalization capability to OOD patterns, yet are currently set empirically without adaptive or theoretical guidance. A promising direction is to scale ReCast to a pre-trained large language model with a richer codebook, diverse patch configurations, and heterogeneous time series pre-training, thereby improving robustness and reducing hyperparameter sensitivity.

5 Conclusion

In this work, we present ReCast, a novel codebook-assisted framework for reliable and efficient time series forecasting. Our dual-path architecture innovatively combines patch-wise quantization for capturing recurring local patterns with residual modeling for recovering irregular variations, achieving an optimal balance between lightweight design and forecasting accuracy. The proposed reliability-aware codebook update mechanism, supported by a reliability-aware scoring strategies, ensures robust adaptation to distribution shifts while maintaining stability. Extensive experiments across 8 real-world datasets demonstrate that ReCast outperforms SOTA baselines, achieving superior accuracy with significantly reduced computational complexity.

Acknowledgments

The authors appreciate the financial support by the National Natural Science Foundation of China (NSFC) Joint Fund with Zhejiang Integration of Informatization and Industrialization under Key Project (Grant Number U22A2033), the Postdoctoral Fellowship Program of CPSF under Grant Number GZC20251643, the NSFC under Grant Number 62576193.

References

Ansari, A. F.; Stella, L.; Turkmen, A. C.; Zhang, X.; Mercado, P.; Shen, H.; Shchur, O.; Rangapuram, S. S.; Arango, S. P.; Kapoor, S.; et al. 2025. Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research*.

Duchi, J.; and Namkoong, H. 2019. Variance-based regularization with convex objectives. *Journal of Machine Learning Research*, 20(68): 1–55.

Duvenaud, D.; Wang, J.; Jacobsen, J.; Swersky, K.; Norouzi, M.; and Grathwohl, W. 2020. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations (ICLR)*, volume 4.

Guo, H.; Peng, S.; Yan, Y.; Mou, L.; Shen, Y.; Bao, H.; and Zhou, X. 2023. Compact neural volumetric video representations with dynamic codebooks. *Advances in Neural Information Processing Systems*, 36: 75884–75895.

Hu, Y.; Liu, P.; Zhu, P.; Cheng, D.; and Dai, T. 2025. Adaptive multi-scale decomposition framework for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 17359–17367.

Kong, Y.; Wang, Z.; Nie, Y.; Zhou, T.; Zohren, S.; Liang, Y.; Sun, P.; and Wen, Q. 2025. Unlocking the power of lstm for long term time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 11968–11976.

Lin, S.; Chen, H.; Wu, H.; Qiu, C.; and Lin, W. 2025. Temporal Query Network for Efficient Multivariate Time Series Forecasting. In *Forty-second International Conference on Machine Learning*.

Lin, S.; Lin, W.; Hu, X.; Wu, W.; Mo, R.; and Zhong, H. 2024. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37: 106315–106345.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35: 5816–5828.

Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; and Long, M. 2023. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. *arXiv preprint arXiv:2310.06625*.

Lu, Y.; Wu, R.; Mueen, A.; Zuluaga, M. A.; and Keogh, E. 2022. Matrix profile XXIV: scaling time series anomaly detection to trillions of datapoints and ultra-fast arriving data streams. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 1173–1182.

Ma, X.; Li, X.; Fang, L.; Zhao, T.; and Zhang, C. 2024. U-mixer: An unet-mixer architecture with stationarity correction for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 14255–14262.

Ma, X.; Li, X.; Feng, W.; Fang, L.; and Zhang, C. 2023. Dynamic graph construction via motif detection for stock prediction. *Information Processing & Management*, 60(6): 103480.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Qi, Q.; Guo, Z.; Xu, Y.; Jin, R.; and Yang, T. 2021. An online method for a class of distributionally robust optimization with non-convex objectives. *Advances in Neural Information Processing Systems*, 34: 10067–10080.

Qiu, X.; Hu, J.; Zhou, L.; Wu, X.; Du, J.; Zhang, B.; Guo, C.; Zhou, A.; Jensen, C. S.; Sheng, Z.; et al. 2024. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. *Proceedings of the VLDB Endowment*, 17(9): 2363–2377.

Senin, P.; and Malinchik, S. 2013. Sax-vsm: Interpretable time series classification using sax and vector space model. In *2013 IEEE 13th international conference on data mining*, 1175–1180. IEEE.

Shibo, F.; Zhao, P.; Liu, L.; Wu, P.; and Shen, Z. 2025. Hdt: Hierarchical discrete transformer for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 746–754.

Tang, P.; and Zhang, W. 2025. Unlocking the Power of Patch: Patch-Based MLP for Long-Term Time Series Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12640–12648.

Tian, K.; Jiang, Y.; Yuan, Z.; Peng, B.; and Wang, L. 2024. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37: 84839–84865.

Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.

Wang, M.; Yang, J.; Yang, B.; Li, H.; Gong, T.; Yang, B.; and Cui, J. 2025. Towards Lightweight Time Series Forecasting: a Patch-wise Transformer with Weak Data Enriching. *arXiv preprint arXiv:2501.10448*.

Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; and Sun, L. 2023. Transformers in time series: a survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 6778–6786.

Wu, C.; Chen, X.; Wu, Z.; Ma, Y.; Liu, X.; Pan, Z.; Liu, W.; Xie, Z.; Yu, X.; Ruan, C.; et al. 2025. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 12966–12977.

Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; and Long, M. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34: 22419–22430.

Yeh, C.-C. M.; Zhu, Y.; Ulanova, L.; Begum, N.; Ding, Y.; Dau, H. A.; Silva, D. F.; Mueen, A.; and Keogh, E. 2016. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, 1317–1322. IEEE.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are Transformers Effective for Time Series Forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, 27268–27286. PMLR.