RAC-NET: PATCH-BASED MULTIVARIATE TIME SERIES FORECASTING VIA CODEBOOK ROUTING AND SMALL LANGUAGE MODEL DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The computational overhead of large pre-trained language models and the resource demands of real-world multivariate time-series forecasting (MTSF) can be addressed using the knowledge distillation (KD) framework. Existing approaches provide limited investigation of cached semantic supervision for lightweight forecasting without language-model inference. To address these limitations, the study proposes regime-aware codebook (RAC-Net), a semantic representation distillation framework that uses a small language model (SLM) and a lightweight patch-based Transformer network for MTSF. RAC-Net first leverages a pre-trained small language model (SLM) prompted with statistical trends to perform semantic reasoning and produce high-fidelity latent representations. A lightweight student backbone model is developed with a low-rank codebook routing mechanism. The directional cosine alignment loss function aligns the student routing representations with the latent representations of the SLM teacher model. After training, the teacher model is decoupled, leaving the lightweight trained student model for inference. Extensive evaluation across seven publicly available datasets demonstrates that RAC-Net achieves competitive forecasting performance with inference latencies of 2.9–13.67 ms per multivariate sample. On a high-dimensional *Electricity* dataset at a long horizon of 720, RAC-Net achieves 23.17% and 16% relative improvements in mean squared error compared to PatchTST and iTransformer, respectively, with $0.87G$ inference MAC operations.

1 INTRODUCTION

Accurate and computationally efficient multivariate time series forecasting (MTSF) has widespread adoption in various domains, including weather forecasting, energy grid management, stock analysis, and traffic planning (Liu et al., 2026). Time series data contain periodic patterns, multi-scale dynamics, and high-frequency local fluctuations and macro-level regimes (Wang et al., 2024). Consequently, the forecasting network's performance depends on effective modelling of these data patterns. Within short context lengths, the local cycles are easy to model due to their high frequency, daily fluctuation, and repetitiveness. However, accurately capturing macro-regimes and global cycles such as seasonal shifts or monthly or weekly trends is a challenging task and remains an open research gap. These cycles occur less frequently, as they require a long time span and are often masked by missing and extreme values and high-frequency noise (Niu et al., 2026; Sun & Boning, 2026). Despite these challenges, effectively capturing global patterns is necessary because they are strongly correlated with the future target sequences. Authors (Cao et al., 2026) performed correlation analysis and reported that global dependencies are more informative than local cycles for accurate MTSF.

In recent years, various deep learning networks, including recurrent neural networks (RNN) (Lin et al., 2025b; Chen et al., 2025a), convolutional neural networks (CNN) (Cheng et al., 2025; Liu et al., 2022), Transformers (Zhou et al., 2021; Saravana et al., 2026), and Mamba models (Zhang et al., 2023), have been reported for MTSF. The networks are efficient for local pattern understanding and provide forecasts over a fixed look-back window L. Consequently, these networks lack dataset-level regime dynamics and macroscopic temporal context that lie beyond L. To capture global patterns, the researchers implemented different methodologies. The authors (Yu et al., 2026; Zhang et al., 2026; Wang et al., 2024; 2025a) explore seasonal-trend decomposition and separated-periodic component analysis techniques. Studies (Wu et al., 2022; Liu et al., 2025d) convert the data sequence from 1D to 2D to capture intra-period and inter-period dependencies. Some studies employ the fast Fourier transform (FFT) (Chen et al., 2025b) and wavelet transform (Wei et al., 2025), which operate in the frequency domain to model cyclic patterns. These techniques have shown promising results for MTSF; however, they suffer from two main drawbacks. These techniques operate in fixed look-back windows. Increasing the look-back window yields performance gains for modern Transformer architectures, while it incurs computational costs and quadratic memory growth. More importantly, extending the look-back window does not guarantee the model's access to global dataset-level regime dynamics. Moreover, these techniques rely on computationally intensive networks that suffer from high inference latency and MAC counts. High

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

inference latency limits their utility in real-time applications, and with massive MAC counts, the model cannot be deployed on resource-constrained devices.

To overcome the challenges of capturing global patterns and macro-regimes, distinct operational states characterised by shared statistical primitives, while ensuring computationally efficient deployability, this study proposed a regime-aware codebook network (RAC-Net) for MTSF. In RAC-Net, a knowledge distillation framework is introduced that grounds data-wide macro-regime dynamics into compact geometric centroids. Prior to the training of the student model, a pre-trained small language model (SLM) is prompted with statistical trends to generate rationale tokens, and high-fidelity latent targets are produced through the mean-pooling operation of top-layer hidden representations. Unlike conventional distillation techniques that transfer scalar logits, RAC-Net leverages directional angular alignment and distils the SLM's hidden state into the student representation space. The student model comprises a lightweight patch-based Transformer integrated with a dynamic low-rank mixture codebook. The low-rank adapter matrices $(\boldsymbol{A}_k, \boldsymbol{B}_k)$ with routing gates modulate the temporal features of the low-rank mixture codebook, subjected to the extracted latent representation from the SLM. During the inference stage, the SLM is decoupled, leaving the trained lightweight model for forecasting. This allows the RAC-Net to capture macro-regimes during network training while performing inference with millisecond latency and minimal MAC operations, making it suitable for real-time applications and resource-constrained devices. Overall, the following are the main contributions of the proposed study:

- The study proposes a knowledge distillation architecture for MTSF using a pre-trained SLM-grounded latent representation, eliminating training and inference-time language-model executions through offline centroid prompting. Unlike existing methods that rely on prompt-tuning and continuous distillation with active language-model inference, RAC-Net first generates a fixed dictionary by capturing macro-regime dynamics in the offline stage through prompting over geometric clusters. These regime representations are distilled into a low-rank mixture codebook embedded within the lightweight student model.
- We propose a patch-based Transformer with a dynamic low-rank mixture codebook that routes patch embeddings via a rank r adapter and is regularised with a multi-objective loss function to enforce geometric distillation fidelity and codebook activation diversity.
- Across seven benchmark datasets, RAC-Net achieves competitive MTSF performance with low runtime overhead and parameter footprint, facilitating real-world deployment.

2 Related Work

Deep learning and Transformer models: Sequential recurrent unit-based techniques (Lin et al., 2025b; Chen et al., 2025a), CNN-based methods (Cheng et al., 2025; Liu et al., 2022), multilayer perceptron techniques (Jiang et al., 2025; Murad et al., 2025), Transformer-based models (Zhou et al., 2021; Saravana et al., 2026), and Mamba-based techniques (Zhang et al., 2023) are widely used for MTSF in the literature. This technique focuses on periodicity modelling and converting the representation to a 2D structural transformation to capture complex data dependencies. In periodicity modelling, the Autoformer (Wu et al., 2021), DLinear (Zeng et al., 2023), and TimeMixer(Wang et al., 2024) networks have been recently reported. These techniques perform seasonal-trend decomposition to capture patterns. FiLM (Zhou et al., 2022), FITS (Xu et al., 2024), DUET (Qiu et al., 2025) and FilteNet (Yi et al., 2024) operate in the frequency domain for time-series forecasting. These networks leverage transformation techniques such as the FFT to convert time-series input into spectral components. With frequency domain modelling, these networks separate periodic patterns and long-term cycles from high-frequency noise, improving the model's generalisation. TimesNet (Wu et al., 2022), P2D-former (Hong et al., 2026a), LightTS (Zhang et al., 2022), MDCNet (Su et al., 2024) and Times2D (Nematirad et al., 2025) networks use data transformation with 1D-to-2D representation for MTSF. TimesNet uses a CNN-based vision backbone to learn 2D temporal variation in 2D space. Similar to TimesNet, the MDCNet leverages a 2D CNN network with a variational mode-decomposed time series component. The LightTS network leverages continuous and interval sampling to extract short-term local patterns and long-term global dependencies, respectively. These sample representations are then processed by MLP network. Times2D uses frequency-domain modelling to decompose input data and then feeds the representation to a 2D CNN to learn frequency-domain dependencies. Recently reported xPatch (Stitsyuk & Choi, 2025) and PITS (Lee et al., 2024) studies focus on lightweight forecasting networks. To reduce parameter count, these networks leverage compact tokenisation and hierarchical dependencies. Despite the compact tokenisation and periodicity decomposition, these architectures rely on unguided feature extraction and have limited forecasting performance on complex multivariate datasets such as *Traffic* and *Electricity*.

Retrieval Augmented forecasting: In (Han et al., 2025; Anonymous, 2025; Yang et al., 2025) the retrieval augmented method is implemented for MTSF. These methods first retrieve similar segments for context from external repositories or full history and then feed that information to the forecasting module. These networks are mostly used in a few-shot setting, where the network is fed with the recurring patterns alongside the current window. However, these networks rely on similarity search algorithms and don't capture temporal patterns themselves, thereby motivating techniques that can learn global data dependencies.

Knowledge Distillation Forecasting : For MTSF, limited work has been performed with knowledge distillation methods. In (Floratos et al., 2022), a KD method is proposed for online financial forecasting that uses output-matching paradigms to minimise the difference between teacher and student predictions. The regime-aware adaptive knowledge distillation-based latent manifold projection (LMP-Net) technique (Hong et al., 2026b) is tested across multiple datasets. LMP-Net employs similarity-preserving KD and optimal transport for feature alignment at the topological and geometric levels. The regime-aware setting reduces teacher misguidance on student models through a confidence-weighted adaptive gating mechanism. The STRIDE network (Ahamed et al., 2026), comprising LLM, SLM and TSFM, is proposed for MTSF. During the training phase, the LLM is first used to generate reasoning knowledge for global and macro-regimes, and then reasoning knowledge is distilled for the SLM. During inference, the SLM hidden dimensions are projected to TSFM for forecasting. STRIDE provides a semantic distillation framework in which reasoning representations are produced and then distilled to subsequent models. However, leveraging LLM for semantic representation generation and relying on an active SLM during inference introduce a significant computational burden, motivating techniques that are deployable on edge devices. By computing the scaling coefficients, X-LoRA (Buehler & Buehler, 2024) uses a gating mechanism to mix multiple LoRA experts' outputs within the LLM dense transformer layer. The cross-modal LLM fine-tuning (CALF) (Liu et al., 2025c), distillation method TimeKD (Liu et al., 2025a), and cross-modal caching TimeCMA (Liu et al., 2025b) frameworks remove the LLM branch during inference. However, in X-LoRA, TimeKD and CALF, the student training phase operates with an active, end-to-end forward pass through the LLM, resulting in high training-time computational costs. TimeCMA uses static last-token prompt embeddings, which cannot capture multi-step temporal regime dynamics. Existing work has explored language-model-assisted forecasting, teacher-free inference through distillation, global retrieval, and routed low-rank adapters separately. Less attention has been given to whether a small number of cached language-model embeddings derived from representative statistical patterns can supervise an input-adaptive, lightweight multivariate forecaster. To bridge this gap, RAC-Net trains its student model without an end-to-end forward pass through the SLM. First, it performs offline statistical centroid prompting to generate a fixed cached dictionary, and then via directional cosine distillation, the regime representations are transferred to an input-adaptive low-rank mixture codebook for MTSF without requiring an active SLM.

3 METHODOLOGY

This section describes the RAC-Net mathematical formulation, which integrates the reasoning distillation of SLM into a small patch-based Transformer with a dynamic low-rank mixture codebook for MTSF. The proposed methodological framework leverages the SLM in the offline caching stage, leaving the lightweight student model for inference.

3.1 PROBLEM FORMULATION AND DATA STANDARDISATION

Let $\mathbf{X}$ denotes the multivariate time series sequence with a look-back context L across channels C.

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L]^\top = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(C)}] \in \mathbb{R}^{L \times C} \tag{1}$$

where the spatial vector across all channels C at discrete timestamp $t \in \{1, 2, \cdots, L\}$ is denoted by $\mathbf{x}_t = [x_{t,1}, x_{t,2}, \ldots, x_{t,C}]^\top \in \mathbb{R}^C$ and the temporal trajectory for the individual channel $c \in \{1, 2, \cdots, C\}$ is represented by $\mathbf{x}^{(c)} = [x_{1,c}, x_{2,c}, \ldots, x_{L,c}]^\top \in \mathbb{R}^L$.

The objective of the forecasting model is to predict $\mathbf{Y} \in \mathbb{R}^{H \times C}$ over the forecasting horizon H.

$$\mathbf{Y} = [\mathbf{x}_{L+1}, \mathbf{x}_{L+2}, \ldots, \mathbf{x}_{L+H}]^\top \in \mathbb{R}^{H \times C} \tag{2}$$

The multivariate sequence $\mathbf{X}$ is first decomposed into C univariate series $\{\mathbf{x}^{(c)}\}_{c=1}^{C} \subset \mathbb{R}^L$. To enable the patch-based encoder to extract local temporal features uniformly across the channels prior to the higher-order spatial correlations reintroduction, the batch tensor is reshaped from $\mathbb{R}^{B \times L \times C}$ to an instance-flattened representation $\mathbb{R}^{(B \cdot C) \times L}$. Where B denotes the batch size.

For each channel $c \in \{1, 2, \cdots, C\}$, univariate series $\mathbf{x}^{(c)} \in \mathbb{R}^L$ are standardised using reversible instance normalisation (RevIN). The RevIN mitigates cross-variate scale distortion and non-stationary distribution shifts across the data batches.

$$\tilde{\mathbf{x}}^{(c)} = \text{RevIN}_{\text{norm}}(\mathbf{x}^{(c)}) = \frac{\mathbf{x}^{(c)} - \mu^{(c)}}{\sqrt{(\sigma^{(c)})^2 + \epsilon}} \cdot \gamma^{(c)} + \beta^{(c)}, \tag{3}$$

Where instance statistics are represented by

$$\mu^{(c)} = \frac{1}{L} \sum_{t=1}^{L} x_{tc} \tag{4}$$

$$\sigma^{(c)} = \sqrt{\frac{1}{L} \sum_{t=1}^{L} (x_{tc} - \mu^{(c)})^2 + \epsilon} \tag{5}$$

The learnable affine parameters are denoted by γ and β, while $\epsilon = 1 \times 10^{-5}$ is used for numerical stability. The network's predictions are restored to the original scale using $\text{RevIN}_{\text{denorm}}(\cdot)$ function. For scale-invariant statistical primitive extraction, the standardised zero-mean and unit-variance sequences are preserved.

$$\bar{\mathbf{x}}^{(c)} = \frac{\mathbf{x}^{(c)} - \mu^{(c)}}{\sqrt{(\sigma^{(c)})^2 + \epsilon}}. \tag{6}$$

In the proposed framework, the student model uses the RevIN-transformed series $\tilde{\mathbf{x}}^{(c)}$ while the SLM is executed with standardised representations $\bar{\mathbf{x}}^{(c)}$. This enables the student model to capture instance dynamics during backpropagation, while the SLM's prompting and geometric clustering remain scale-invariant and stationary.

3.2 Grounded semantic knowledge distillation

The proposed RAC-Net network leverages SLM to extract high-fidelity reasoning trajectories from data patterns, which are then distilled to the student model. The SLMs have millions of parameters, and executing their reasoning on raw continuous floating-point sequences results in severe tokenisation degradation and numeric hallucinations. To overcome this problem, deterministic statistical mapping is performed, $\Phi : \mathbb{R}^L \to \mathcal{S}$, which extracts invariant temporal trends to ensure hallucination-resistant semantic distillation.

$$\Phi(\bar{\mathbf{x}}^{(c)}) = \left\{ \beta_1(\bar{\mathbf{x}}^{(c)}),\, \sigma^2(\bar{\mathbf{x}}^{(c)}),\, \mathcal{P}_{\text{Top-}\iota}(\bar{\mathbf{x}}^{(c)}) \} \right\} \in \mathcal{S}, \tag{7}$$

The dominant spectral trends are extracted using the 1D discrete FFT, denoted by $\mathcal{P}_{\text{Top-}\iota}(\cdot)$. The $\sigma^2(\bar{\mathbf{x}}^{(c)})$ and $\beta_1(\bar{\mathbf{x}}^{(c)})$ represent empirical variance, and ordinary least-squares trend slope, respectively. The slope trajectory is calculated using closed-form linear regression across discrete temporal indices.

$$\beta_1(\bar{\mathbf{x}}^{(c)}) = \frac{\sum_{t=1}^{L}(t - \bar{t})\bar{x}_t^{(c)}}{\sum_{t=1}^{L}(t - \bar{t})^2}, \tag{8}$$

where $\bar{t} = \frac{L+1}{2}$ denotes the temporal midpoint.

The 1D discrete FFT is executed over the mean-centred sequence, and top-ι spectral signals are identified as

$$\mathcal{P}_{\text{Top-}\iota}(\bar{\mathbf{x}}^{(c)}) = \left\{ \frac{L}{k_j} \;\middle|\; k_j = \arg\text{top-}\,\iota_{k \in \{1,\ldots,\lfloor L/2 \rfloor\}} \left| \mathcal{F}(\bar{\mathbf{x}}^{(c)} - \bar{\mu})_k \right| \right\}_{j=1}^{\iota}. \tag{9}$$

In the proposed study, we set $\iota = 1$ to separate the primary base cycle while retaining prompt representation's compactness and determinism.

3.2.1 Representative centroid caching

For multivariate time series data, querying the SLM for each input window is computationally inefficient and intractable. To mitigate this problem, the proposed system first normalised training sequences and then clustered them into $M = 128$ geometric centroids $\{\boldsymbol{u}_m\}_{m=1}^{M}$. The pre-trained SLM is called with the prompting technique. Every centroid $\boldsymbol{u}_m$ is added into the prompt $\mathcal{T}_{\text{prompt}}$ after being mapped through $\Phi(\boldsymbol{u}_m)$. The SLM interprets every centroid with $\mathcal{T}_{\text{prompt}}$ and generates temporal regime rationale tokens J_m. The model extracts the top-layer hidden representations of the generated tokens $\{\mathbf{h}_j^{(m)}\}_{j=1}^{J_m} \subset \mathbb{R}^{d_{\text{SLM}}}$ and aggregates through temporal mean pooling.

$$\bar{\mathbf{h}}^{(m)} = \frac{1}{J_m} \sum_{j=1}^{J_m} \mathbf{h}_j^{(m)} \in \mathbb{R}^{d_{\text{SLM}}}. \tag{10}$$

Through linear projection operations, the student model dimension d_{target} is distilled with a pooled rationale vector $\bar{\mathbf{h}}^{(m)}$ with a non-trainable orthogonal linear projection matrix $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{d_{\text{target}} \times d_{\text{SLM}}}$ and unit hypersphere normalisation.

$$\mathbf{z}_{TA}^{(m)} = \frac{\mathbf{W}_{\text{proj}} \bar{\mathbf{h}}^{(m)}}{\max\left(\|\mathbf{W}_{\text{proj}} \bar{\mathbf{h}}^{(m)}\|_2,\, \epsilon \right)} \in \mathbb{S}^{d_{\text{target}}-1}, \tag{11}$$

where ϵ is used to ensure numerical stability.
All the reasoning dictionaries are cached on the disk.

$$\mathcal{Z}_{\text{cached}} = \{\boldsymbol{z}_{TA}^{(m)}\}_{m=1}^{M} \tag{12}$$

During model training, for every input sequence i, the teacher targets are retrieved through the nearest-neighbour method.

$$\boldsymbol{z}_i = \boldsymbol{z}_{TA}^{(\arg\min_m ||\bar{\mathbf{x}}_i - \boldsymbol{u}_m||_2^2)} \tag{13}$$

After the caching operation completes, the SLM is offloaded, leaving VRAM for the student model training.

Under standardisation (Eq. 6), individual instance windows satisfy $\sigma(\bar{\mathbf{x}}_i) \approx 1$. Computing the centroid vector's dispersion u_m evaluates $\sigma(\frac{1}{|S_m|}\sum_{i\in S_m}\bar{\mathbf{x}}_i)$. For out-of-phase cyclical sequences or zero-mean stochastic noise clusters, the pointwise amplitude of $\mathbf{u_m}$ is attenuated by temporal interference, resulting in the empirical centroid volatility $\sigma(\mathbf{u_m})$ reflecting intra-cluster phase coherence and central tendency rather than raw series energy.

3.3 Cross-channel attention with horizon forecasting

The RAC-Net framework combines the temporal patch representation with spatial attention exchange across multivariate sequences $\mathbf{X} \in \mathbb{R}^{B\times L\times C}$ before the forecast horizon H projection. The first $(B\cdot C, L)$ format is created by taking the transpose and flattening the input tensor across the channel and then embedding it into temporal patch representations $\mathbf{P} \in \mathbb{R}^{(B\cdot C)\times P\times d_{\text{model}}}$. To retain the variate identity, $\mathbf{P}$ is reshaped to $(B, C, P, d_{\text{model}})$ and augmented with channel tokens $\mathbf{E}_C \in \mathbb{R}^{C\times d_{\text{model}}}$

$$\mathbf{P}_{\text{aug}} = \mathbf{P} + \mathbf{1}_B \otimes \mathbf{E}_C \otimes \mathbf{1}_P^\top \in \mathbb{R}^{B\times C\times P\times d_{\text{model}}}. \tag{14}$$

The augmented tensor is reshaped back to $(B\cdot C, P, d_{\text{model}})$ and processed via the Transformer encoder to yield a temporal feature matrix $\mathbf{H} \in \mathbb{R}^{(B\cdot C)\times P\times d_{\text{model}}}$. The representation $\mathbf{H}$ is unflattened to $(B, C, P, d_{\text{model}})$ and permuted to $(B, P, C, d_{\text{model}})$ to compute spatial dependencies across every temporal patch position. Grouping the channels C into sequence tokens yields:

$$\mathbf{H}_{\text{spatial}} = \text{Permute}_{(0,2,1,3)}(\text{Reshape}(\mathbf{H}, (B, C, P, d_{\text{model}}))) \in \mathbb{R}^{(B\cdot P)\times C\times d_{\text{model}}}. \tag{15}$$

Across C multi-head attention is applied, which is followed by a residual connection and linear normalisation.

$$\mathbf{H}_{\text{inter}} = \text{LayerNorm}\left(\mathbf{H}_{\text{spatial}} + \text{MHA}(\mathbf{Q} = \mathbf{H}_{\text{spatial}}, \mathbf{K} = \mathbf{H}_{\text{spatial}}, \mathbf{V} = \mathbf{H}_{\text{spatial}})\right) \in \mathbb{R}^{(B\cdot P)\times C\times d_{\text{model}}}. \tag{16}$$

The channel-flattened format $\mathbf{H}_f \in \mathbb{R}^{(B\cdot C)\times P\times d_{\text{model}}}$ is restored with spatial representations.

3.4 Dynamic Low-Rank Mixture Codebook

The student model consists of a Transformer encoder with 1D temporal patch tokenisation. The model processes every normalised sequence $\tilde{\mathbf{x}}^{(c)}$ and produces a temporal representation $\boldsymbol{H} \in \mathbb{R}^{P\times d_{\text{model}}}$. Where the number of patches is denoted by P and the hidden dimension size of Transformer encoder layers is represented by d_{model}. To dynamically update models' representations with SLM latent vectors, a learnable codebook $\mathcal{C}$ is created.

$$\mathcal{C} = \{(\boldsymbol{A}_k, \boldsymbol{B}_k)\}_{k=1}^K \tag{17}$$

Every key k in the codebook is parameterised with low-rank modulation adapter rank-r, which is decomposed into down-projection $\boldsymbol{A}_k \in \mathbb{R}^{d_{\text{model}}\times r}$ and up-projection $\boldsymbol{B}_k \in \mathbb{R}^{r\times d_{\text{model}}}$.

For reasoning keys, routing probabilities $\boldsymbol{p} \in \Delta^{K-1}$ are calculated at the sequence-window level. With global average pooling, the patch representations from the Transformer encoder are condensed.

$$\bar{\mathbf{h}} = \mathcal{G}(\mathbf{H}) = \frac{1}{P}\sum_{j=1}^{P}\mathbf{h}_j \in \mathbb{R}^{d_{\text{model}}} \tag{18}$$

A parameterised gating head $\mathcal{R}_{\text{gate}}(\cdot)$ is used to compute the routing distribution over codebook keys.

$$\mathbf{p} = \text{Softmax}(\mathcal{R}_{\text{gate}}(\bar{\mathbf{h}})) \in \Delta^{K-1}, \tag{19}$$

where $\Delta^{K-1} = \{\mathbf{p} \in \mathbb{R}_+^K \mid \sum_{k=1}^K p_k = 1\}$.

The dynamic low-rank perturbation $\boldsymbol{\delta} \in \mathbb{R}^{d_{\text{model}}}$ is calculated through basis expansion conditioned on the routing weights $\mathbf{p}$.

$$\boldsymbol{\delta} = \sum_{k=1}^{K} p_k \left(\bar{\mathbf{h}}\mathbf{A}_k\mathbf{B}_k\right) \in \mathbb{R}^{d_{\text{model}}} \tag{20}$$

In the student model, the modulated pooled embedding is projected into the distillation space via alignment head f_{proj}:

$$\mathbf{z}_{SA} = f_{\text{proj}}(\bar{\mathbf{h}} + \boldsymbol{\delta}) \in \mathbb{R}^{d_{\text{target}}}, \tag{21}$$

With the routing weights $\mathbf{p}$ from $\bar{\mathbf{h}} = \mathcal{G}(\mathbf{H}_f)$, the $\boldsymbol{\delta} \in \mathbb{R}^{d_{\text{model}}}$ is expanded across patches and incorporated through a non-linear gated residual.

$$\mathbf{H}_{\text{mod}} = \mathbf{H}_f + \gamma \cdot \tanh\left(\left[\mathbf{H}_f \parallel \mathbf{1}_P \boldsymbol{\delta}^\top\right] \mathbf{W}_{\text{comb}}^\top\right) \in \mathbb{R}^{(B\cdot C)\times P\times d_{\text{model}}}, \tag{22}$$

where $\mathbf{W}_{\text{comb}} \in \mathbb{R}^{d_{\text{model}}\times 2d_{\text{model}}}$ and $\gamma \in \mathbb{R}$ is a learnable scaling parameter, while $[\cdot \parallel \cdot]$ denotes feature concatenation.

The temporal and feature dimensions are flattened to map the patch representation to the forecasting horizon H.

$$\mathbf{v}_{\text{flat}} = \text{vec}(\mathbf{H}_{\text{mod}}) \in \mathbb{R}^{(B\cdot C)\times(P\cdot d_{\text{model}})}. \tag{23}$$

A two-layer projection head is used to flatten representations to normalised predictions.

$$\hat{\mathbf{Y}}_{\text{norm}} = \mathcal{F}_{\text{head}}(\mathbf{v}_{\text{flat}}) \in \mathbb{R}^{(B\cdot C)\times H}. \tag{24}$$

The predictions are restored to the original scale through $\text{RevIN}_{\text{denorm}}(\cdot)$ and reshaped to $\hat{\mathbf{Y}} \in \mathbb{R}^{B\times H\times C}$. The complete tensor-shape formatting is:

$$\begin{aligned}(B, L, C) &\xrightarrow{\text{Transpose}} (B\cdot C, L) \xrightarrow{\text{Patch}} (B, C, P, d) \xrightarrow{\text{Permute}} (B, P, C, d) \xrightarrow{\text{MHA}} (B\cdot C, P, d)\\ &\xrightarrow{\text{Flatten}} (B\cdot C, P\cdot d) \xrightarrow{\mathcal{F}_{\text{head}}} (B\cdot C, H) \xrightarrow{\text{Reshape}} (B, H, C).\end{aligned} \tag{25}$$

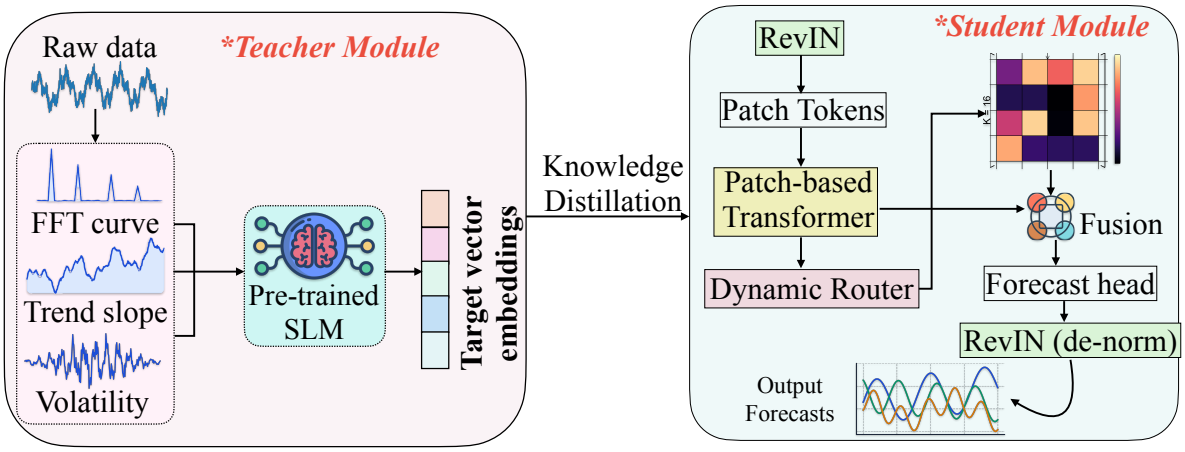


Figure 1: RAC-Net overall architecture. First, statistical features are extracted for the SLM teacher, which produces latent representations. The patch-based Transformer model encodes the normalised sequence into patch tokens, performs global average pooling and evaluates sequence-level routing probabilities $\mathbf{p} \in \Delta^{K-1}$. The fusion block calculates $\mathbf{H}_{\text{mod}}$, combining patch features $\mathbf{H}$ to the low-rank routed perturbation $\boldsymbol{\delta} = \sum_{k=1}^{K} p_k(\bar{\mathbf{h}}\mathbf{A}_k\mathbf{B}_k)$ through a parameterised combine gate.

3.5 Optimisation Objectives

The student model is trained with multi-objective functions: forecast accuracy, semantic distillation alignment and routing diversity.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}}(\hat{Y}, Y) + \lambda_{\text{MAE}}\mathcal{L}_{\text{MAE}}(\hat{Y}, Y) + \lambda_{\text{distil}}\mathcal{L}_{\text{distil}}(\boldsymbol{z_{SA}}, \boldsymbol{z_{TA}}) + \lambda_{\text{div}}\mathcal{L}_{\text{div}} \tag{26}$$

where

$$\mathcal{L}_{MSE} = \frac{1}{H\times C}\sum_{t=1}^{H}\sum_{c=1}^{C}(\hat{y}_t^{(c)} - y_t^{(c)})^2 \tag{27}$$

$$\mathcal{L}_{MAE} = \frac{1}{H \times C} \sum_{t=1}^{H} \sum_{c=1}^{C} |\hat{y}_t^{(c)} - y_t^{(c)}| \tag{28}$$

$$\mathcal{L}_{distil} = 1 - \frac{\boldsymbol{z_{SA}} \cdot \boldsymbol{z}_{TA}}{\max(||\boldsymbol{z_{SA}}||_2 ||\boldsymbol{z}_{TA}||_2, \epsilon)} \tag{29}$$

To prevent the router from delegating all inputs to the trivial subset of adapters, resulting in representation collapse, the negative Shannon entropy of the batch-mean router distribution $\bar{\boldsymbol{p}} = \frac{1}{B_{eff}} \sum_{b=1}^{B_{eff}} \boldsymbol{p}^{(b)}$ is minimised.

$$\mathcal{L}_{\text{div}} = \sum_{k=1}^{K} \bar{p}_k \log(\bar{p}_k + \epsilon) - \alpha \frac{1}{B_{eff}} \sum_{b=1}^{B_{eff}} \sum_{k=1}^{K} p_{b,k} \log(p_{b,k} + \epsilon) \tag{30}$$

where α of 0.05 is used to balance micro-level routing confidence against macro-level code balance. The optimisation objective is to evaluate across all active channel instances $B_{eff} = B \cdot C$

Before calculating the distillation loss, the student model uses a learnable linear head to map the patch transformer's hidden states to dimension d_{target}. Figure 1 describes the RAC-Net training and inference workflow.

4 Experimental setup

The proposed study evaluates RAC-Net on seven publicly available MTSF benchmark datasets: *ETTh1*, *ETTh2*, *ETTm1*, *ETTm2*, *Weather*, *Electricity*, and *Traffic*. The RAC-Net is evaluated in a multivariate setting where the model forecasts all variables. For data partitioning into training, validation, and testing sets, we follow the prior rule of 6:2:2 for *ETT* and 7:1:2 for other datasets (Zhou et al., 2021; Liu et al., 2022). To provide robust comparative analysis, RAC-Net is compared against benchmarked Transformer variants and recently reported techniques, including GTR (Cao et al., 2026), LMP-Net (Hong et al., 2026b), S-Mamba (Wang et al., 2025b), TQNet (Lin et al., 2025a), RAFT (Han et al., 2025), SOFTs (Han et al., 2024), iTransformer (Liu et al., 2024), PatchTST (Nie et al., 2023) and DLinear (Zeng et al., 2023). The mean absolute error (MAE) and mean squared error (MSE) are used as forecasting performance indicators and computed after RevIN denormalisation. All experiments were implemented using PyTorch on an NVIDIA GB10 GPU with 128 GB of unified memory (CUDA 13.0). To ensure fair comparison, we used a fixed context length of $L = 96$ and predicted for future horizons $H \in \{96, 192, 336, 720\}$. A detailed description of the dataset, RAC-Net hyperparameters, and training configuration is provided in Appendix A.

4.1 Long-Term Forecasting

Table 1 reports the average long-term forecasting results on 4 horizons. The results demonstrate that the proposed RAC-Net achieves competitive forecasting performance, outperforming state-of-the-art networks on multiple datasets. On large multivariate datasets, *Electricity* ($C = 321$) and *Traffic* ($C = 862$), RAC-Net outperforms the KD baseline LMP-Net by reporting relative improvements of 4.83% and 11.20% in MAE. With semantic distillation and dynamic codebook routing, the patch-based Transformer model in RAC-Net outperforms benchmarked PatchTST on all datasets. Apart from *ETTm2*, RAC-Net achieves superior forecasting performance and reports the lowest MAE across the *ETT*, *Traffic*, *Electricity*, and *Weather* datasets. On *ETTm2* datasets, RAC-Net maintains competitive performance with state-of-the-art techniques. Overall, RAC-Net achieves a consistently better MAE score over the L_2 loss (MSE). This behaviour is due to the joint objective loss function ($\mathcal{L}_{\text{forecast}} = \text{MSE} + \lambda_{\text{MAE}}\text{MAE}$, with $\lambda_{\text{MAE}} = 2.0$) together with semantic codebook regularisation. By aligning student representations to SLM-derived macro-temporal representations, the RAC-Net favours stable distributional-level alignment over high-frequency noise spikes and transient anomalies. Consequently, across the entire temporal sequence, the RAC-Net maintains superior pointwise fidelity, while L_2 penalties increase the MSE.

4.2 Semantic Distillation Ablation Analysis

Table 2 reports the performance of RAC-Net against the conventional benchmark PatchTST, RAC-Net variants ($\lambda_{\text{distil}} = 0$ and λ_{MSE} only), a direct-statistics baseline replacing SLM with a 2-layer MLP projection over statistical summaries ($\beta_1, \sigma^2, \mathcal{P}_{\text{dom}}, \kappa$) and random target vectors control assigning fixed isotropic random unit vectors to each centroid to demonstrate the impact of semantic distillation from the teacher model. The **RAC-Net ($\lambda_{\text{distill}} = 0$)** configuration uses an unguided codebook baseline without SLM teacher distillation, while the diversity penalty and joint L_1/L_2 forecasting objectives are included. In the MSE-only configuration, RAC-Net is trained with the L_2 forecasting objective and diversity regularisation while setting $\lambda_{\text{distill}} = 0$ and $\lambda_{\text{MAE}} = 0$. The directional alignment between student low-rank routing representations and the teacher model's projected latent representation is evaluated

Table 1: Multivariate long-term time-series forecasting results. The reported results are the average across 4 prediction horizons $\{96, 192, 336, 720\}$ with a fixed context length of 96. The best and the second-best results are highlighted in **red** and **underlined**, respectively. The count measures how many times the model ranks in the top 2. Full results on each horizon are provided in Appendix H.

Dataset	RAC-NET		GTR		LMP-Net		S-Mamba		TQNet		RAFT		SOFTs		iTransformer		PatchTST		DLinear	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.441	**0.431**	0.439	0.434	**0.430**	**0.432**	0.455	0.450	0.443	0.434	**0.428**	0.433	0.449	0.443	0.454	0.448	0.469	0.455	0.456	0.452
ETTh2	**0.369**	**0.395**	0.372	0.400	**0.367**	**0.397**	0.381	0.405	0.378	0.402	0.382	0.410	0.373	**0.400**	0.383	0.407	0.387	0.407	0.559	0.515
ETTm1	**0.375**	**0.384**	**0.367**	**0.389**	0.379	0.395	0.398	0.405	0.377	0.393	0.381	0.400	0.393	0.402	0.407	0.410	0.387	0.400	0.403	0.407
ETTm2	0.281	**0.323**	**0.268**	**0.315**	0.278	0.325	0.288	0.332	0.277	**0.323**	0.281	0.330	0.287	0.330	0.288	0.332	0.281	0.326	0.350	0.401
Weather	0.244	**0.267**	**0.239**	**0.268**	0.246	0.271	0.251	0.276	0.242	0.269	0.270	0.309	0.255	0.278	0.258	0.278	0.259	0.273	0.265	0.317
Electricity	**0.161**	**0.256**	0.166	0.260	0.179	0.269	0.170	0.265	**0.165**	**0.258**	0.175	0.272	0.174	0.264	0.178	0.270	0.205	0.290	0.212	0.300
Traffic	0.468	**0.267**	0.470	0.280	0.479	0.301	**0.414**	**0.276**	0.445	0.276	**0.414**	0.284	**0.409**	**0.267**	0.428	0.282	0.481	0.300	0.625	0.383
Count	**3**	**7**	3	4	2	2	1	1	1	3	2	0	1	2	0	0	0	0	0	0

Table 2: Evaluating the effectiveness of RAC-Net semantic distillation. The context window is fixed at 96 and the results are for a forecasting horizon of $H = 720$. **RAC-Net** denotes the full configuration with a multi-objective function. The relative reduction in MSE by semantic distillation over the **RAC-Net** ($\lambda_{\textbf{distill}} = 0$) baseline is reported in the **Improvement (%)** column.

Dataset	PatchTST		RAC-Net ($\lambda_{\text{distill}} = 0$)			RAC-Net (λ_{MSE} only)			MLP-Statistics Control			Random Embeddings			RAC-Net (Full Model)			Improvement
	MSE	MAE	MSE	MAE	CosSim	MSE	MAE	CosSim	MSE	MAE	CosSim	MSE	MAE	CosSim	MSE	MAE	CosSim	MSE (%)
ETTh1	0.500	0.488	0.492	0.476	-0.08	0.495	0.482	0.05	0.491	0.474	0.98	0.496	0.479	0.24	**0.485**	**0.468**	0.96	**1.42%**
ETTh2	0.431	0.446	0.423	**0.438**	0.02	0.425	0.445	-0.13	0.419	**0.438**	0.98	0.421	0.439	0.20	**0.417**	**0.438**	0.96	**1.42%**
ETTm1	0.454	0.439	0.450	0.434	-0.22	0.453	0.438	-0.10	0.457	0.444	0.99	0.462	0.442	0.50	**0.445**	**0.430**	0.98	**1.11%**
ETTm2	**0.402**	**0.400**	0.406	**0.400**	0.22	0.414	0.411	-0.25	0.432	0.418	0.97	0.438	0.421	0.54	0.404	**0.400**	0.96	**0.49%**
Weather	0.354	0.340	0.358	0.347	-0.17	**0.332**	0.357	-0.18	0.359	0.348	0.99	0.359	0.348	0.61	0.350	**0.342**	0.98	**2.23%**
Electricity	0.246	0.324	0.192	0.286	0.16	0.192	0.291	0.16	0.191	0.284	0.97	0.194	0.286	0.41	**0.189**	**0.283**	0.98	**1.56%**
Traffic	**0.514**	0.320	0.532	**0.286**	0.10	0.532	0.318	-0.01	0.542	**0.286**	0.98	0.559	0.294	0.49	0.546	**0.286**	0.98	-2.56%

using cosine similarity. With $\lambda_{\text{distil}} = 0$, the student model operates in unguided geometric space with directional similarity of CosSim $\in [-0.21, 0.22]$. In the full RAC-Net configuration, the student model achieves direct convergence with semantic distillation as CosSim increases to ≥ 0.96 across all datasets. With semantic distillation, RAC-Net's forecasting performance improves, with notable gains of 2.23% and 1.56% in MSE recorded for the *Weather* and *Electricity* datasets. Findings also demonstrate that the MLP-statistics baseline achieves high cosine alignment; however, the forecasting performance is degraded relative to RAC-Net. This validates that the regime rationale synthesised by the SLM improves the overall forecasting performance more than MLP-projected heuristic statistical descriptors. With the MSE-only configuration, the model achieves a gain in isolated MSE for *Weather* and *Traffic* datasets; however, a notable degradation in MAE is reported across every dataset. This confirms that RAC-Net leverages distillation with multi-task loss balancing for robust MTSF. Apart from the *Traffic* and *ETTm2* datasets, the results in Table 2 also show that RAC-Net outperforms the benchmark PatchTST. In the case of the *Traffic* dataset, the MAE of RAC-Net remains superior, whereas the MSE increases.

4.3 Model's Computational Efficiency Analysis

For real-world deployment, the model's forecasting performance must be measured alongside its computational efficiency. Table 3 and Figure 2 compare the RAC-Net MSE, inference-time parameter counts, and operations with PatchMLP, Pathformer, Crossformer, PatchTST, iTransformer, and LMP-Net across the *ETTh1* and *Electricity* datasets. The reported metrics are for the inference-time complexity of the student model, as the SLM is discarded during the inference stage. All computations are reported on a per-sample multivariate forward pass ($B = 1$, covering all channels C) and parametric matrix multiplications. In the case of the *ETTh1* dataset, the RAC-Net recorded $16\times$ fewer MAC operations than LMP-Net; however, LMP-Net outperforms RAC-Net in forecasting performance. RAC-Net's advantages emerges as the channel cardinality scales. For the high-dimensional *Electricity* dataset ($C = 321$), RAC-Net reports the inference footprint of 1.55M parameters and 0.87G MACs while achieving the lowest MSE of 0.189. Compared to LMP-Net, the RAC-Net recorded $17.72\times$ fewer inference MACs with 13.70% improvement in MSE. Moreover, for *Electricity* and *Traffic* datasets, RAC-Net achieves inference latencies of 2.9 and 13.67 ms per multivariate sample, respectively. This validates that RAC-Net distils semantic representations from the teacher model into a compact student model that executes with minimal inference computational overhead.

Additionally, we conducted sensitivity analysis on RAC-Net's offline centroid budget, codebook routing dynamics, influence of look-back window and rank, and codebook size, which are provided in the Appendices B, C, D, and E, respectively.

Table 3: RAC-Net computational efficiency and forecasting performance comparison with state-of-the-art techniques using parameter counts, MACs, and forecasting accuracy (MSE). The reported results are for a forecasting horizon of 720 with a fixed look-back window of 96. The best and the second-best reported results are highlighted in **red** and **underlined**, respectively.

Model	*Electricity* ($C = 321$)			*ETTh1* ($C = 7$)		
	Params (M)	**MACs (G)**	**MSE**	**Params (M)**	**MACs (G)**	**MSE**
PatchMLP	**1.91**	22.25	0.248	0.97	0.22	0.490
Pathformer	4.42	63.71	0.237	0.65	4.11	0.496
Crossformer	13.65	1303.92	0.230	10.82	168.29	0.678
PatchTST	10.74	411.61	0.246	13.89	26.42	0.500
iTransformer	9.88	51.32	0.225	1.79	0.63	0.503
LMP-Net	3.01	**15.42**	**0.219**	**0.18**	**0.08**	**0.472**
RAC-Net (Ours)	**1.55**	**0.87**	**0.189**	**0.50**	**0.005**	**0.485**

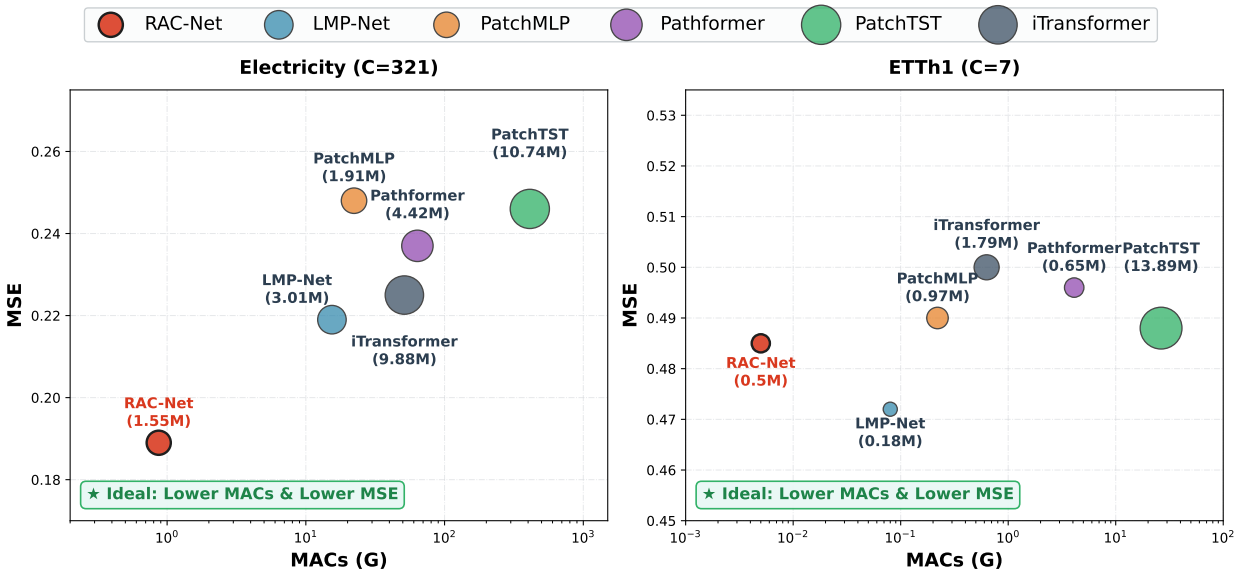


Figure 2: Forecasting performance with inference computational efficiency across the *Electricity* ($C = 321$) and *ETTh1* ($C = 7$) datasets. The y-axis represents the MSE, bubble size reflects the number of parameters, and the x-axis denotes MACs on a logarithmic scale. The RAC-Net achieves a superior trade-off between inference computational efficiency and forecasting capabilities and resides in the optimal left region.

5 Conclusions

In the proposed study, we presented a RAC-Net framework for computationally efficient MTSF. RAC-Net fills the gap between abstract semantic reasoning and statistical pattern modelling by aligning the student routing representation to the SLM latent representation while decoupling the teacher network prior to deployment to ensure practical computational efficiency. Extensive comparative analysis across seven benchmark real-world datasets illustrates that RAC-Net provided competitive forecasting performance with low inference MAC operations. On a high-dimensional benchmark dataset (*Electricity*, $C = 321$ at $H = 720$), RAC-Net recorded a relative improvement of 13.70% in MSE and $17.72\times$ fewer MAC operations than the KD-based model (LMP-Net). Moreover, ablation studies are conducted, which demonstrate the effectiveness of semantic representation on the model's forecasting performance. Future work will focus on hierarchical and spatial-graph clustering techniques to overcome representation bottlenecks and dynamic sensor channel partitioning in high-cardinality datasets such as *Traffic*.

REFERENCES

Md Atik Ahamed, Mihir Parmar, Palash Goyal, Chun-Liang Li, Qiang Cheng, Tomas Pfister, and Jinsung Yoon. Reasoning-aware training for time series forecasting. *arXiv preprint arXiv:2605.08625*, 2026.

Anonymous. TS-RAG: Retrieval-augmented generation based time series foundation models are stronger zero-shot forecaster, 2025. URL https://openreview.net/forum?id=TJuUelhGQr.

Eric L Buehler and Markus J Buehler. X-lora: Mixture of low-rank adapter experts, a flexible framework for large language models with applications in protein mechanics and molecular design. *APL Machine Learning*, 2(2), 2024.

Fanpu Cao, Lu Dai, Jindong Han, and Hui Xiong. Enhancing multivariate time series forecasting with global temporal retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=QUJBPSfyui.

Junfeng Chen, Azhu Guan, Jingjing Du, and Altangerel Ayush. Multivariate time series prediction with multi-feature analysis. *Expert Systems with Applications*, 268:126302, 2025a.

Wuqi Chen, Junjie Ye, Chunna Zhao, and Yaqun Huang. Mffcnn: multi-scale fractional fourier transform convolutional neural network for multivariate time series forecasting: W. chen et al. *The Journal of Supercomputing*, 81(2): 416, 2025b.

Mingyue Cheng, Jiqian Yang, Tingyue Pan, Qi Liu, Zhi Li, and Shijin Wang. Convtimenet: A deep hierarchical fully convolutional model for multivariate time series analysis. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 171–180, 2025.

Pavlos Floratos, Avraam Tsantekidis, Nikolaos Passalis, and Anastasios Tefas. Online knowledge distillation for financial timeseries forecasting. In *2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–6. IEEE, 2022.

Lu Han, Xu-Yang Chen, Han-Jia Ye, and De-Chuan Zhan. Softs: Efficient multivariate time series forecasting with series-core fusion. *Advances in Neural Information Processing Systems*, 37:64145–64175, 2024.

Sungwon Han, Seungeon Lee, Meeyoung Cha, Sercan O Arik, and Jinsung Yoon. Retrieval augmented time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=GUDnecJdJU.

Hu Hong, Guorong Chen, Wang Jian, Haixin Yi, Zhang Bao, Weijie Zhang, and Zhikang Ding. P2d-former for time-series forecasting with period-aware 2d representations and exogenous covariates. *Scientific Reports*, 2026a.

Junyao Hong, Zesheng Lai, Xinyi Xiao, Suyang Zhou, Aodong Shen, and Youyong Kong. Beyond point predictions: Manifold expansion and dual alignment for robust time series distillation. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=CYh8wTyRkZ.

Hongwei Jiang, Dongsheng Liu, Xinyi Ding, Yaning Chen, and Hongtao Li. Tcm: An efficient lightweight mlp-based network with affine transformation for long-term time series forecasting. *Neurocomputing*, 617:128960, 2025.

Seunghan Lee, Taeyoung Park, and Kibok Lee. Learning to embed time series patches independently. In *International Conference on Learning Representations*, volume 2024, pp. 27599–27620, 2024.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. Cyclenet: Enhancing time series forecasting through modeling periodic patterns. *Advances in Neural Information Processing Systems*, 37: 106315–106345, 2024.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=e24CueVty2.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. Segrnn: Segment recurrent neural network for long-term time series forecasting. *IEEE Internet of Things Journal*, 2025b.

Chenxi Liu, Hao Miao, Qianxiong Xu, Shaowen Zhou, Cheng Long, Yan Zhao, Ziyue Li, and Rui Zhao. Efficient multivariate time series forecasting via calibrated language models with privileged knowledge distillation. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 3165–3178. IEEE, 2025a.

Chenxi Liu, Qianxiong Xu, Hao Miao, Sun Yang, Lingzheng Zhang, Cheng Long, Ziyue Li, and Rui Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18780–18788, 2025b.

Kun Liu, Renjun Jia, Ruifeng Yang, Xirui Zeng, Yuqi Liang, and Cen Chen. Are global dependencies necessary? scalable time series forecasting via local cross-variate modeling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=CNVL194fO5.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in neural information processing systems*, 35:5816–5828, 2022.

Peiyuan Liu, Hang Guo, Tao Dai, Naiqi Li, Jigang Bao, Xudong Ren, Yong Jiang, and Shu-Tao Xia. Calf: Aligning llms for time series forecasting via cross-modal fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18915–18923, 2025c.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Zhao Liu, Jingyi Wang, Huachun Tan, and Fan Ding. Long-term traffic flow prediction via spatiotemporal sequence reconstruction on highway networks. *Expert Systems with Applications*, pp. 130845, 2025d.

Md Mahmuddun Nabi Murad, Mehmet Aktukmak, and Yasin Yilmaz. Wpmixer: Efficient multi-resolution mixing for long-term time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 19581–19588, 2025.

Reza Nematirad, Anil Pahwa, and Balasubramaniam Natarajan. Times2d: Multi-period decomposition and derivative mapping for general time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19651–19658, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Yiming Niu, Jinliang Deng, and Yongxin Tong. Phaseformer: From patches to phases for efficient and effective time series forecasting. In *International Conference on Learning Representations*, volume 2026, pp. 8262–8285, 2026.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. Duet: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD conference on knowledge discovery and data mining v. 1*, pp. 1185–1196, 2025.

MK Saravana, MS Roopa, JS Arunalatha, and KR Venugopal. Transformers for multivariate time series forecasting: Comprehensive analysis, challenges, research opportunities and future prospects. *IEEE Access*, 2026.

Artyom Stitsyuk and Jaesik Choi. xpatch: Dual-stream time series forecasting with exponential seasonal-trend decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20601–20609, 2025.

Jing Su, Dirui Xie, Yuanzhi Duan, Yue Zhou, Xiaofang Hu, and Shukai Duan. Mdcnet: Long-term time series forecasting with mode decomposition and 2d convolution. *Knowledge-Based Systems*, 299:111986, 2024.

Fan-Keng Sun and Duane S Boning. PAPer: Periodicity alignment on periodic time series for forecasting, 2026. URL https://openreview.net/forum?id=cOTAC1cT5N.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 38626–38652, 2024.

Shiyu Wang, Jiawei Li, Xiaoming Shi, Zhou Ye, Baichuan Mo, Wenze Lin, Shengtong Ju, Zhixuan Chu, and Ming Jin. Timemixer++: A general time series pattern machine for universal predictive analysis. In *International Conference on Learning Representations*, volume 2025, pp. 16980–17016, 2025a.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025b.

Wei Wei, Zi'Ang Wang, Bowen Pang, Jiannan Wang, and Xue Liu. Wavelet transformer: An effective method on multiple periodic decomposition for time series forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 36(8):14063–14077, 2025.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Zhijian Xu, Ailing Zeng, and Qiang Xu. Fits: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, volume 2024, pp. 26295–26318, 2024.

Silin Yang, Dong Wang, Haoqi Zheng, and Ruochun Jin. Timerag: Boosting llm time series forecasting via retrieval-augmented generation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Kun Yi, Jingru Fei, Qi Zhang, Hui He, Shufeng Hao, Defu Lian, and Wei Fan. Filternet: Harnessing frequency filters for time series forecasting. *Advances in neural information processing systems*, 37:55115–55140, 2024.

Haoyang Yu, Shuai Lu, Dong Pan, Zhaohui Jiang, and Weihua Gui. Industrial nonstationary multivariate time series forecasting via frequency-domain decomposition and seasonal-trend bidirectional interaction. *IEEE Transactions on Industrial Informatics*, 2026.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Michael Zhang, Khaled K Saab, Michael Poli, Tri Dao, Karan Goel, and Christopher Ré. Effectively modeling time series with simple discrete state spaces. *arXiv preprint arXiv:2303.09489*, 2023.

Mingyi Zhang, Ao Hu, Liangjian Wen, Jun Wang, Yong Dai, Dongkai Wang, and Jiang Duan. Mrdnet: Multivariable relational decomposition network for multivariate time series forecasting. *Knowledge-Based Systems*, pp. 115554, 2026.

Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*, 2022.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35:12677–12690, 2022.

A Implementation Setup Details

This section describes the implementation details, including dataset descriptions, performance evaluation metrics, and experimental setup.

A.1 Dataset Description

In the proposed study, the experimental analysis is conducted across seven widely used MTSF benchmark datasets.

ETT (Zhou et al., 2021) datasets contain load and temperature data of electricity transformers collected in two different regions of China. These datasets are recorded at 60-minute (*ETTh1*, *ETTh2*) and 15-minute (*ETTm1*, *ETTm2*) resolutions.

Weather (Wu et al., 2022) dataset is collected from meteorological stations in Germany. This dataset is recorded at 10-minute intervals and contains 21 meteorological variables.

Traffic (Wu et al., 2022) dataset is collected across the freeways of the San Francisco Bay Area. This dataset is collected with 862 detectors and includes hourly road occupancy rates.

Electricity (Wu et al., 2022) dataset is collected from the UCI Machine Learning Repository. The dataset comprises electricity consumption records of 321 consumers at hourly intervals. Table 4 reports the dataset dimensions and resolutions; dataset size regarding the training, validation, and testing ratio; and prediction horizons used in the experimental setup.

Table 4: Summary of benchmark MTSF datasets used in experimental analysis. The dataset size represents raw chronological timestamps per train/validation/test split before applying the context window L and forecasting horizon H.

Task	Dataset	Dataset Size (Train/Val/Test)	Resolution	Dimensions (C)	Prediction Horizons (H)
MTSF	ETTh1	(8,545 / 2,881 / 2,881)	60 min	7	$\{96, 192, 336, 720\}$
	ETTh2	(8,545 / 2,881 / 2,881)	60 min	7	$\{96, 192, 336, 720\}$
	ETTm1	(34,465 / 11,521 / 11,521)	15 min	7	$\{96, 192, 336, 720\}$
	ETTm2	(34,465 / 11,521 / 11,521)	15 min	7	$\{96, 192, 336, 720\}$
	Weather	(36,792 / 5,271 / 10,540)	10 min	21	$\{96, 192, 336, 720\}$
	Electricity	(18,317 / 2,633 / 5,261)	60 min	321	$\{96, 192, 336, 720\}$
	Traffic	(12,185 / 1,757 / 3,509)	60 min	862	$\{96, 192, 336, 720\}$

A.2 Performance Metrics

To evaluate the network's forecasting performance, we used MSE and MAE. The following equations define these performance metrics:

$$MSE = \frac{1}{T}\sum_{i=1}^{T}(\hat{y}_i - y_i)^2 \tag{31}$$

$$MAE = \frac{1}{T}\sum_{i=1}^{T}|\hat{y}_i - y_i| \tag{32}$$

A.3 Simulation setup

All experiments are conducted on an NVIDIA GB10 GPU with 128 GB of unified memory (CUDA 13.0) using PyTorch. Across every dataset, the AdamW optimiser is used with a weight decay of 1×10^{-4}. The teacher model leverages Qwen 0.5B SLM from the family of Qwen 2.5 Instruct. The target dimension is set to 32. RevIN is enabled across every dataset. Detailed hyperparameter settings across each dataset are provided in Table 5. The d_{model} denotes the internal representation quality of the Transformer encoder and cross-attention layers. The subspace used by the projection head of the student model to align with teacher latent representations is denoted by d_{target}. For fair comparison, all experiments are conducted with standard benchmark protocols (look-back window $L = 96$, prediction horizons $H \in \{96, 192, 336, 720\}$). For baseline comparisons, we directly report results from the published literature (Liu et al., 2026; Hong et al., 2026b; Niu et al., 2026) where the data split and experimental setup align with our configurations. The inference MACs and parameter count of the models are compared under a consistent setting of $L = 96$ and $H = 720$.

Processing the large-scale multivariate dataset (*Electricity* $C = 321$, *Traffic* $C = 862$) requires significant GPU memory. In the RAC-Net framework, to ensure computational efficiency, input batches with shape (B, L, C) are transposed and flattened into temporal instances of shape $(B{\cdot}C, L)$. This helps the Transformer model extract localised temporal features via 1D temporal patching with computational efficiency. The channel embeddings $\mathbf{E} \in \mathbb{R}^{C_{\max} \times d_{\text{model}}}$ are added, and a cross-channel attention module is applied across C dimensions for each temporal patch token to capture inter-series dependencies. This enables each variate to attend to the remaining $C - 1$ series before projection. The temporal patches of each channel are mapped to the target horizon H independently by a two-layer projection head $\mathcal{F}_{\text{head}}$, yielding shape $(B \cdot C, H)$. Subsequently, it is reshaped and transposed (B, H, C) to support multivariate evaluation.

Table 5: RAC-Net hyperparameter configuration across different datasets.

Hyperparameter	***ETT***	***Weather***	***Electricity / Traffic***
Backbone Architecture			
Patch length	12	16	16
Stride	12	4	16
Model dimension (d_{model})	64	128	128
FFN hidden dimension (d_{ffn})	256	512	512
Transformer layers (N)	2	2	2
Attention heads (h)	4	8	8
Dropout rate	0.1	0.05	0.05
Activation function	GELU	GELU	GELU
Instance Normalization	RevIN	RevIN	RevIN
Weight initialization	Kaiming Uniform	Kaiming Uniform	Kaiming Uniform
Codebook & Distillation Parameters			
Clustering centroids (M)	128	128	128
Codebook capacity (K)	16	16	16
Rank (r)	4	4	4
Target projection dimension (d_{target})	32	32	32
Teacher SLM	Qwen-2.5-0.5B	Qwen-2.5-0.5B	Qwen-2.5-0.5B
Max rationale tokens	48	48	48
Distillation weight (λ_{distill})	1.0	1.0	0.5
MAE loss weight (λ_{mae})	2.0	2.0	2.0
Entropy loss weight (λ_{div})	1.0	1.0	1.0
Clustering (k-means) Settings			
Algorithm	MiniBatch k-means	MiniBatch k-means	MiniBatch k-means
Centroid initialization	k-means++	k-means++	k-means++
Maximum iterations	100	100	100
Random seed	42	42	42
Optimisation & Training Schedule			
Batch size	32	32	16
Base learning rate	1.0×10^{-4}	1.0×10^{-4}	5.0×10^{-4}
Learning-rate schedule	Cosine Annealing	Cosine Annealing	Cosine Annealing
Minimum learning rate	1.0×10^{-6}	1.0×10^{-6}	1.0×10^{-6}
Maximum epochs	40	40	40
Early stopping patience	5	5	5
Gradient clipping norm	1.0	1.0	1.0
Optimiser	AdamW	AdamW	AdamW
Weight decay	1.0×10^{-4}	1.0×10^{-4}	1.0×10^{-4}

Algorithm 1 Semantic knowledge extraction and caching.

Input: Training dataset $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, context window L, centroid budget $M = 128$, teacher model $\mathcal{M}_{\text{SLM}}$, target dimension $d_{\text{target}} = 32$, $J_{\max} = 48$.
Output: Cached reasoning dictionary $\mathcal{Z}_{\text{cached}}$ and centriod indices $\{\text{idx}_i\}_{i=1}^{N_{\text{total}}}$.
Look-back window extraction across all channels: $\mathbf{X}_{\text{window}} \in \mathbb{R}^{(N_{\text{total}} \times L)}$
Applying Mini-batch K-means on $\{\bar{\mathbf{x}}_i\}_{i=1}^{N_{\text{total}}}$ to get $\{\boldsymbol{u}_m\}_{m=1}^{M}$
Assigning each sample i to the nearest centroid: $\text{idx}_i \leftarrow \arg\min_m \|\bar{\mathbf{x}}_i - \boldsymbol{u}_m\|_2^2$
for each centroid m in $\{1, 2, 3, \cdots, M\}$ **do**
    extract statistical trends using Equation (7)
    Format-grounded template $\mathcal{T}_{\text{prompt}}$ from $\Phi(\boldsymbol{u}_m)$
    Tokenise with attention mask $\mathbf{M}_{\text{in}}$
    Autoregressive rationale token generation with greedy decoding ($J_m \leq J_{\max}$, temperature $= 0$)
    Obtain top-layer hidden states $\rightarrow \{\mathbf{h}_j^{(m)}\}_{j=1}^{J_m} \subset \mathbb{R}^{d_{\text{SLM}}}$.
    Execute mean pooling for aggregating rationale states $\rightarrow \bar{\mathbf{h}}^{(m)} = \frac{1}{J_m}\sum_{j=1}^{J_m} \mathbf{h}_j^{(m)}$
    Projection and unit hypersphere normalisation $\rightarrow \mathbf{z}_{TA}^{(m)} = \frac{\mathbf{W}_{\text{proj}}\bar{\mathbf{h}}^{(m)}}{\max\left(\|\mathbf{W}_{\text{proj}}\bar{\mathbf{h}}^{(m)}\|_2, \epsilon\right)} \in \mathbb{S}^{d_{\text{target}}-1}$
end for
Save dictionary $\mathcal{Z}_{\text{cached}}$ and instance assignments $\{\text{idx}_i\}$ to disk.
Offload SLM to CPU memory
return $\mathcal{Z}_{\text{cached}}$, $\{\text{idx}_i\}$

Algorithm 2 RAC-NET training and Inference.

Input: Training dataset $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, student model $\mathcal{M}_{\text{student}}$, Cached reasoning dictionary $\mathcal{Z}_{\text{cached}}$, $\mathcal{C} = \{(\mathbf{A}_k, \mathbf{B}_k)\}_{k=1}^{K}$, max epochs $e_{\max}$, loss hyperparameters $(\lambda_{\text{MAE}}, \lambda_{\text{distil}}, \lambda_{\text{div}})$
Output: Trained $\mathcal{M}^*_{\text{student}}$ and test prediction $\hat{\mathbf{Y}}$.
— Stage 1: Student Model Training —
for epoch $= 1$ to e_{max} **do**
    for each $\{(\mathbf{x}_b, \mathbf{y}_b, \text{idx}_b)\}_{b=1}^{B} \subset \mathcal{D}_{\text{train}}$ **do**
        Extract teacher targets: $\mathcal{Z}_{\text{cached}}[\text{idx}_b] \rightarrow \boldsymbol{z}_{TA}$
        RevIN normalisation of every input sequence
        Temporal patch representation extraction: $\text{PatchEncoder}(\tilde{\mathbf{x}_b}) \in \mathbb{R}^{(B\cdot C)\times P \times d_{\text{model}}} \rightarrow \boldsymbol{H}$
        Compute cross-channel spatial representations: $\text{CrossChannelAttn}(\mathbf{H}, \mathbf{E}_C) \rightarrow \mathbf{H}_f$
        Calculate routing distribution: $\text{Softmax}(\mathcal{R}_{\text{gate}}(\mathcal{G}(\mathbf{H}_f))) \in \Delta^{K-1} \rightarrow \boldsymbol{p}$
        Dynamic low-rank perturbation: $\sum_{k=1}^{K} p_k(\mathcal{G}(\mathbf{H}_f)\mathbf{A}_k\mathbf{B}_k) \rightarrow \boldsymbol{\delta}$
        Modulated patch tokens: $\text{CombineGate}(\mathbf{H}_f, \mathbf{1}_P\boldsymbol{\delta}^\top) \rightarrow \mathbf{H}_{\text{mod}}$
        Student rationale embedding projection: $f_{\text{proj}}(\mathcal{G}(\mathbf{H}_f) + \boldsymbol{\delta}) \in \mathbb{R}^{d_{\text{target}}} \rightarrow \mathbf{z}_{SA}$
        Generate normalised forecast $\hat{\mathbf{Y}}_{norm}$
        Compute multi-objective loss using Equation (27)
    end for
end for
— Stage 2: Inference —
for each test window $x_{\text{test}} \in \mathbb{R}^{L\times C}$ **do**
    Compute predictions: $\mathcal{M}^*_{\text{student}}(x_{\text{test}}) \rightarrow \hat{Y}$
end for
return Trained student model $\mathcal{M}^*_{\text{student}}$ and predictions $\hat{Y}$.

For deterministic and consistent results across simulation runs, we used a fixed random seed (seed $= 0$, `torch.manual_seed(0)` and `numpy.random.seed(0)`) to initialise the student network's weights. Moreover, the frozen-parameter ($\nabla_\theta = 0$) SLM is used with disabled gradient tracking (`requires_grad=False`) for reasoning and semantic cached representations for deterministic distillation across independent runs.

Algorithms 1 and 2 describe the complete workflow of semantic knowledge extraction, the caching process, RAC-Net training, and inference.

B OFFLINE CENTROID BUDGET

We evaluated the trade-off between the offline SLM teacher caching overhead, clustering reasoning fidelity, and marginal gains across the training splits for reasoning centroid budget M. Let the 1D look-back context window extracted from all multivariate channels be denoted by $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, where $L = 96$ and $\mathbf{x}_i \in \mathbb{R}^L$. For a budget M, each sequence is assigned to cluster assignment $c(i) \in \{1, \ldots, M\}$ by k-mean partitioning with prototype centriod $\{\mathbf{u}_m\}_{m=1}^{M} \in \mathbb{R}^L$.

Total within-cluster variance measures the empirical quantisation error across the dataset.

$$\text{WCSS}(M) = \sum_{i=1}^{N} \min_{m \in \{1,\ldots,M\}} \|\mathbf{x}_i - \mathbf{u}_m\|_2^2 \tag{33}$$

The representation gain by $M_{j-1} \to M_j$ is calculated as follows:

$$\Delta_{M_j} = \text{WCSS}(M_{j-1}) - \text{WCSS}(M_j) \tag{34}$$

The cumulative variance $\mathcal{R}_{\text{cumul}}(M_j)$ and relative marginal percentage reduction $\mathcal{R}_{\text{rel}}(M_j)$ with the baseline budget are calculated as follows:

$$\mathcal{R}_{\text{cumul}}(M_j) = \frac{\text{WCSS}(M_0) - \text{WCSS}(M_j)}{\text{WCSS}(M_0)} \times 100\%, \mathcal{R}_{\text{rel}}(M_j) = \frac{\Delta_{M_j}}{\text{WCSS}(M_{j-1})} \times 100\%, \tag{35}$$

The inference time is measured for SLM prompting and reason extraction across all M centroids with forward step latency τ_{fwd} and batch size B_{teacher}.

$$T_{\text{cache}}(M) = \left\lceil \frac{M}{B_{\text{teacher}}} \right\rceil \times \tau_{\text{fwd}} \tag{36}$$

Table 6: Offline reasoning centroid budget analysis across *Electricity* and *Weather* datasets with a context length of $L = 96$. Selected operating points for experimental analysis are bold ($M = 128$).

	Electricity					***Weather***				
M	WCSS	Δ_M	Rel. (%)	Cumul. (%)	T_{cache} (s)	WCSS	Δ_M	Rel. (%)	Cumul. (%)	T_{cache} (s)
16	181,666,064.0	—	—	0.00%	0.025	21,461,428.0	—	—	0.00%	0.025
32	150,022,480.0	31,643,584.0	17.42%	17.42%	0.025	18,666,346.0	2,795,082.0	13.02%	13.02%	0.025
64	130,602,440.0	19,420,040.0	12.94%	28.11%	0.049	16,610,837.0	2,055,509.0	11.01%	22.60%	0.049
128	**116,422,664.0**	**14,179,776.0**	**10.86%**	**35.91%**	**0.098**	**14,968,511.0**	**1,642,326.0**	**9.89%**	**30.25%**	**0.098**
256	106,273,160.0	10,149,504.0	8.72%	41.50%	0.196	13,576,585.0	1,391,926.0	9.30%	36.74%	0.196
512	97,176,984.0	9,096,176.0	8.56%	46.51%	0.392	12,492,866.0	1,083,719.0	7.98%	41.79%	0.392

Results from 6 report that the majority of cumulative variance reduction is accounted by scaling M from 16 to 128: 30.25% on *Weather* and 35.91% on *Electricity*. Across both datasets, the caching time with $M = 128$ is 0.098 seconds. By increasing the M to 256 and 512, the steady drop in Δ_M is reported with a linear increase in latency ($T_{\text{cache}} \propto M$). Selecting $M = 128$ provides an optimal trade-off between clustering representation fidelity and offline preparation overhead.

C CODEBOOK ROUTING DYNAMICS

In codebook- and routing-mechanism-based architectures, a codebook collapse problem is often reported where the router collapses into a trivial subset of entries, ignoring most representations. To verify that RAC-Net's low-rank codebook $\mathbf{C} = \mathbf{AB}$ maintains diverse and active representations, we used two evaluation protocols: **Routing Perplexity** and **Codebook Utilisation**.

Let the codebook capacity be denoted by K and the total number of sequence windows processed across the test split be $N_{\text{seq}} = N_{\text{test}} \cdot C$. For each sequence instance $i \in \{1, 2, \cdots, N_{\text{seq}}\}$, through softmax temperature scaling, the router computes the assignment probabilities $\mathbf{p}_i \in \Delta^{K-1}$ over the sequence representation $\bar{\mathbf{h}}_i = \mathcal{G}(\mathbf{H}_i)$.

$$\mathbf{p}_i = \mathrm{Softmax}\left(\mathcal{R}\mathrm{gate}(\bar{\mathbf{h}}_i)\right) \tag{37}$$

Across the test set, the codebook utilisation measures the percentage of codebook vectors that are actively assigned to at least one sequence window.

$$\mathcal{U} = \frac{1}{K}\sum_{k=1}^{K}\mathbb{I}\left(\sum_{i=1}^{N_{\text{seq}}}\mathbb{I}(k_i = k) > 0\right) \times 100\% \tag{38}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. The absence of dead codebook entries is verified with the utilisation of $\mathcal{U} = 100\%$.

Perplexity measures whether the tokens are distributed across codebook entries or collapsed to a single dominant entry. Let the assignment frequency of tokens routed to entry k averaged over the test set, be denoted by $\bar{p}_k$.

$$\bar{p}_k = \frac{1}{N_{\text{seq}}}\sum_{i=1}^{N_{\text{total}}} p_{j,k} \quad \text{where} \sum_{k=1}^{K}\bar{p}_k = 1 \tag{39}$$

The routing entropy and perplexity are calculated as:

$$\mathcal{H}(\bar{\mathbf{p}}) = -\sum_{k=1}^{K}\bar{p}_k \log\left(\bar{p}_k + \epsilon\right), \qquad \mathcal{P} = \exp\left(\mathcal{H}(\bar{\mathbf{p}})\right) \tag{40}$$

The $\epsilon > 0$ is introduced for numerical stability. For $K = 16$, the maximum perplexity for the uniform token assignment across all discrete indices is $\mathcal{P}_{\max} = \exp(\log 16) = 16.0$.

Across all experimental analyses, the soft probability distribution achieved a maximum perplexity $\mathcal{P} \geq 15.98$. Moreover, at the assignment level ($k_j = \arg\max p_{j,k}$), RAC-Net avoids representation collapse by selecting all 16 discrete vectors.

Figure 3 demonstrates that top-1 discrete selection frequencies are non-uniformly distributed across codebook indices. It also highlights that the codebook captures diverse time series patterns.

D Influence of context length

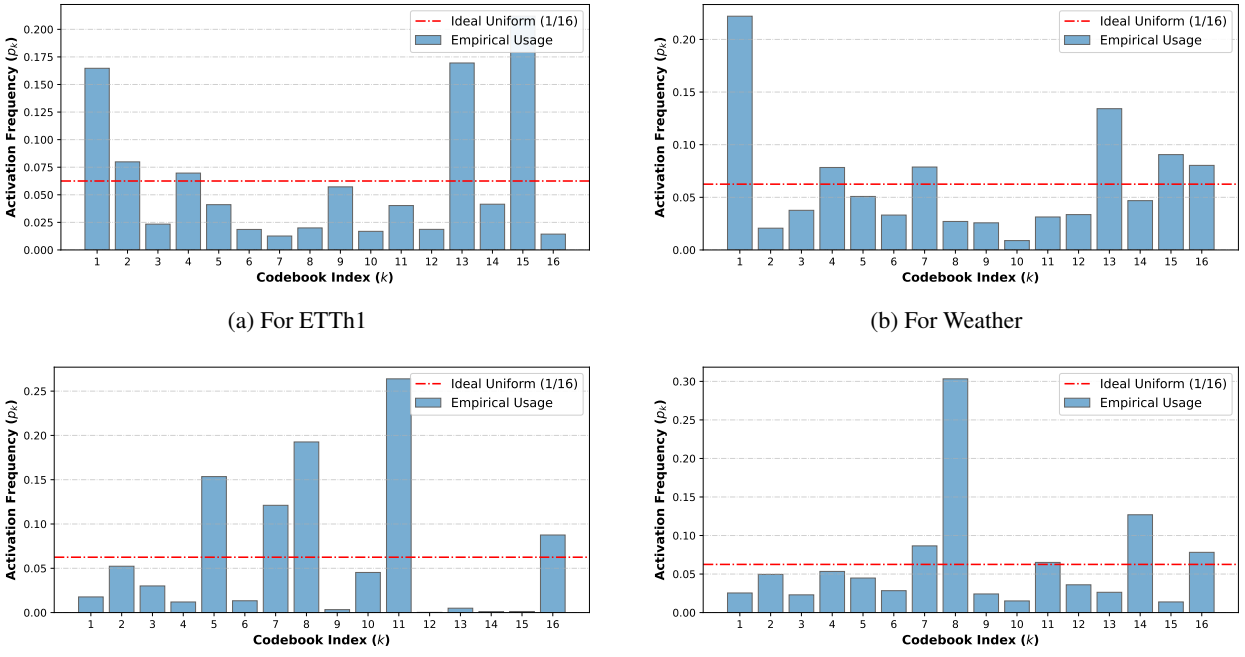
Figure 4 illustrates the performance variation by changing the context window. The results demonstrate that RAC-Net performance consistently improves across both datasets by increasing the context window length. The average MSE of RAC-Net drops from 0.441 to 0.414 (for *ETTh1*) and 0.244 to 0.223 (for *Weather*) by increasing the context window from 96 to 720. RAC-Net's performance gain aligns with the literature findings (Zeng et al., 2023), validating that with the expansion of the context window, the models learn long-term dependencies effectively without overfitting.

Table 7 compares the performance of RAC-Net with PatchTST and CycleNet (Lin et al., 2024) with varying look-back windows.

E Rank and codebook ablation analysis

To analyse the effect of rank and codebook size on the model's performance, we evaluate a grid combination of $r \in \{2, 4, 8\}$ and $K \in \{8, 16, 32\}$. Table 8 reports the average error metric across four forecasting horizons and MAC counts for the *ETTh1* and *Weather* datasets. The configuration selection criterion is set to maximise multi-objective efficiency in achieving optimal forecasting accuracy without incurring high computational overhead. Figure 5 and Table 8 demonstrate that setting $r = 4/K = 16$ provides the most balanced trade-off across both datasets. With these settings, RAC-Net achieves the lowest error for *ETTh1* (0.441 MSE, 0.431 MAE) with moderate MAC counts (5.04M). On *Weather* dataset, increasing K to 32 provides marginal improvement in MSE; however, the MAC count increases by up to 38.9%. With a smaller rank and codebook-size configurations $r = 2, K \in \{8, 16\}$, similar performance is reported for *Weather* dataset with low MACs. However, these configurations result in forecasting performance degradation across *ETTh1* dataset (0.441–0.444/0.447 MSE). These findings validate that $r = 4/K = 16$ offers an optimal configuration for compact temporal routing without bloating the computational complexity.

Table 9 evaluates the effect of r and K on the forecasting performance with $H \in \{96, 192, 336, 720\}$.



(a) For ETTh1

(b) For Weather

(c) For Electricity

(d) For Traffic

Figure 3: Routing activation frequencies for codebook entries across *ETTh1*, *Weather*, *Electricity* and *Traffic* datasets ($L = 96, H = 720$). Meaningful adaptive specialisation across dynamic and transient regimes is demonstrated with a non-uniform activation distribution.

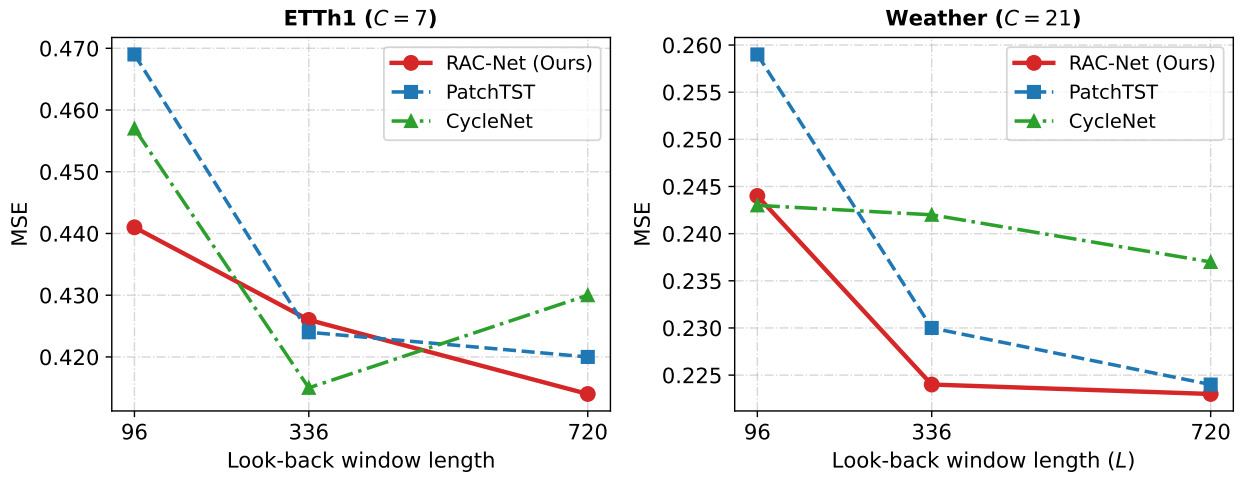


Figure 4: Effect of look-back window on models' forecasting performance. The average MSE across $H \in \{96, 196, 336, 720\}$ is displayed against the look-back windows of $L \in \{96, 336, 720\}$.

Table 7: Performance evaluation with look-back window $L \in \{96, 336, 720\}$ on *ETTh1* and *Weather* datasets. The best and second-best are highlighted with **red** and underlined, respectively.

ETTh1								*Weather*							
Look-back	Horizon	RAC-NET		PatchTST		CycleNet		Look-back	Horizon	RAC-NET		PatchTST		CycleNet	
		MSE	MAE	MSE	MAE	MSE	MAE			MSE	MAE	MSE	MAE	MSE	MAE
$L = 96$	96	0.382	0.392	0.414	0.419	0.375	0.395	$L = 96$	96	0.153	0.194	0.177	0.210	0.158	0.203
	192	0.430	0.422	0.460	0.445	0.436	0.428		192	0.204	0.241	0.225	0.250	0.207	0.247
	336	0.468	0.442	0.501	0.466	0.496	0.455		336	0.268	0.289	0.278	0.290	0.262	0.289
	720	0.485	0.468	0.500	0.488	0.520	0.484		720	0.350	0.342	0.354	0.340	0.344	0.344
	Average	0.441	0.431	0.469	0.455	0.457	0.441		Average	0.244	0.267	0.259	0.273	0.243	0.271
$L = 336$	96	0.375	0.392	0.385	0.405	0.374	0.396	$L = 336$	96	0.146	0.188	0.155	0.204	0.167	0.221
	192	0.418	0.418	0.414	0.421	0.406	0.415		192	0.192	0.234	0.195	0.242	0.221	0.258
	336	0.440	0.435	0.440	0.440	0.431	0.430		336	0.241	0.274	0.249	0.283	0.260	0.293
	720	0.471	0.473	0.456	0.470	0.450	0.464		720	0.318	0.328	0.321	0.334	0.328	0.339
	Average	0.426	0.430	0.424	0.434	0.415	0.426		Average	0.224	0.256	0.230	0.266	0.242	0.278
$L = 720$	96	0.368	0.398	0.377	0.408	0.379	0.403	$L = 720$	96	0.146	0.191	0.149	0.199	0.164	0.220
	192	0.405	0.424	0.413	0.431	0.416	0.425		192	0.193	0.237	0.193	0.243	0.209	0.258
	336	0.434	0.444	0.436	0.444	0.447	0.445		336	0.244	0.275	0.240	0.281	0.255	0.292
	720	0.448	0.465	0.455	0.475	0.477	0.483		720	0.308	0.323	0.312	0.334	0.320	0.338
	Average	0.414	0.432	0.420	0.434	0.430	0.439		Average	0.223	0.257	0.224	0.264	0.237	0.277

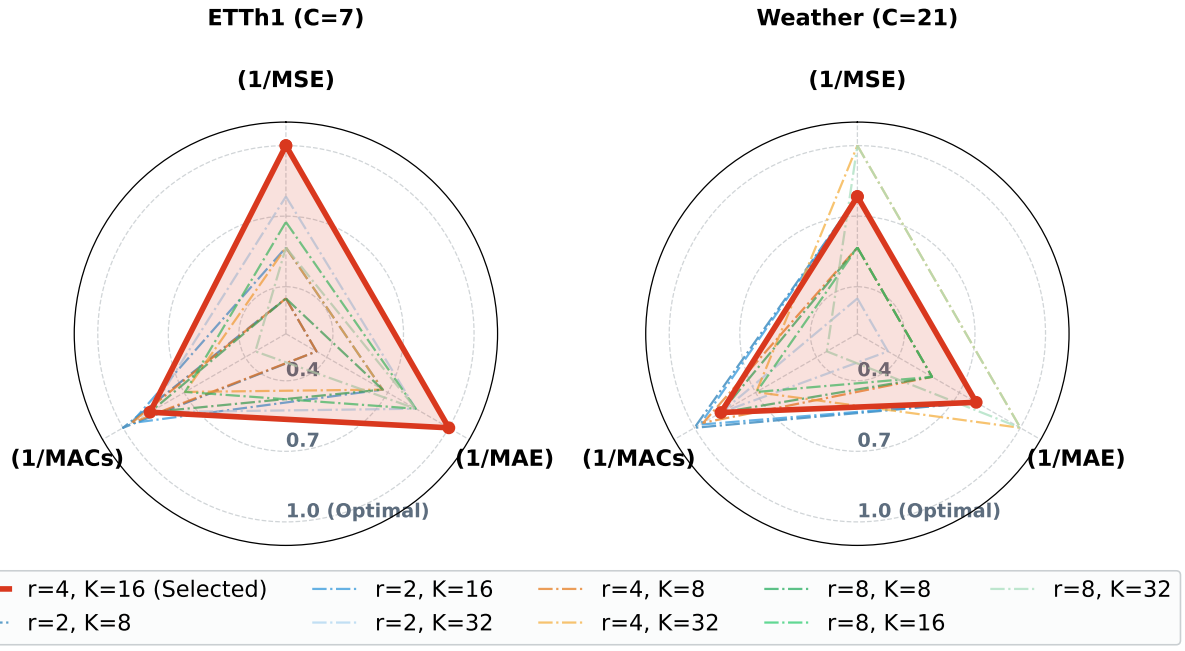


Figure 5: RAC-Net performance metrics across the grid combination of $r \in \{2, 4, 8\}$ and $K \in \{8, 16, 32\}$. The error metrics and MAC operations are normalised between 0.35 and 1, where the optimal trade-off is signified by the larger area. The $r = 4/K = 16$ setting provides the optimal balance between forecasting performance and computational efficiency across both datasets.

Table 8: Effect of rank r and codebook K on RAC-Net forecasting accuracy and computational complexity. The MSE and MAE are the averages across the $H \in \{96, 192, 336, 720\}$ with a fixed context length of 96. The MACs are reported for the long horizon 720.

ETTh1 ($C = 7$)				***Weather*** ($C = 21$)			
r/K	**MSE**	**MAE**	**MACs (M)**	r/K	**MSE**	**MAE**	**MACs (M)**
2 / 8	0.447	0.435	3.64	2 / 8	0.244	0.267	40.14
2 / 16	0.445	0.433	4.06	2 / 16	0.244	0.267	43.65
2 / 32	0.443	0.432	5.04	2 / 32	0.246	0.269	56.68
4 / 8	0.447	0.435	4.06	4 / 8	0.245	0.268	45.64
4 / 16 (Selected)	**0.441**	**0.431**	5.04	**4 / 16 (Selected)**	0.244	0.267	56.66
4 / 32	0.445	0.433	6.86	4 / 32	**0.243**	**0.266**	78.7
8 / 8	0.447	0.433	5.04	8 / 8	0.245	0.268	56.65
8 / 16	0.444	0.432	6.86	8 / 16	0.245	0.268	78.68
8 / 32	0.445	0.432	10.50	8 / 32	**0.243**	**0.266**	122.74

F Error bar analysis

We conducted RAC-Net error bar analysis across four distinct random seeds $\{0, 2024, 2025, 2026\}$ over forecasting horizons $H \in \{96, 192, 336, 720\}$ with varying look-back windows $L \in \{96, 336, 720\}$. The findings in Table 10 verify the multi-seed stability of RAC-Net, as the proposed architecture maintains consistent performance with negligible variance, highlighting that the routing mechanisms reliably converge to semantic codebook assignments. .

Table 11 compares RAC-Net with no distillation, λ_{MSE} only MLP-statistic and random target vector techniques on long horizon ($L = 720$) for the *ETTh1*, *Weather* and *Traffic* datasets across 4 random seed values. The findings demonstrate that the RAC-Net with full configuration achieves the lowest error with stability. The baseline models, MLP-statistic and random target vector, demonstrate high divergence across the *ETTh1* dataset.

G Topology Preservation Under Target Projection

To validate semantic regime geometry preservation with dimensional compression, we examine the pairwise cosine similarities across all $\binom{128}{2} = 8{,}128$ centroid pairs for *ETTh1* and *Electricity/ECL* datasets. Figure 6 demonstrates linear preservation by reporting $r > 0.984$, $\rho > 0.981$. This confirms that the low-dimensional projection preserves regime geometry without topological distortion. The pairwise cosine similarity for both datasets occupies an elevated range ($S \in [0.65, 1.00]$). This behaviour is due to the LLM's directional anisotropy and invariant prompt template tokens. More importantly, the non-linear semantics remain preserved with the rank-order correlation across the $\binom{128}{2} = 8{,}128$ pairs are $\rho = 0.9858$ for ETTh1 and $\rho = 0.9814$ for ECL.

H Per Horizon Forecasting Results

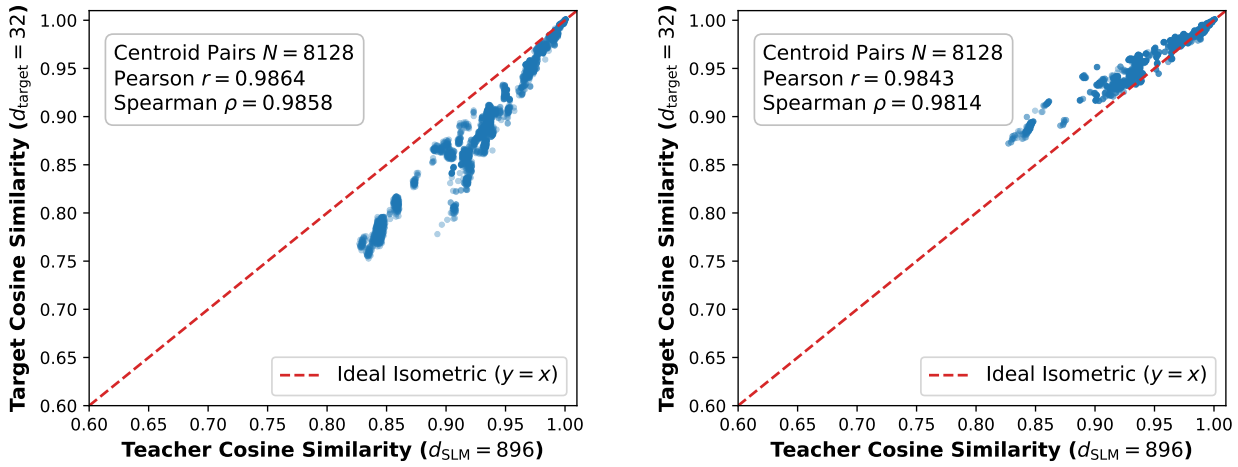
Table 12 compares the forecasting performance of RAC-Net with state-of-the-art techniques across horizons $\{96, 192, 336720\}$ with look-back window of $L = 96$.

I Prompt template and rationale Tokens

Table 13 provides the prompt specification and the reasoning rationale token generated by the teacher SLM (Qwen-2.5-0.5B). The teacher produces a rationale sequence with $N_{\text{tok}} = 48$. The semantic distillation provides better regularisation than numeric features alone. Validated from Table 2, raw scaler feature injection via (**MLP-Statistics Control**) achieves high cosine similarity; however, report higher forecasting errors than RAC-Net with SLM distillation.

Table 9: Effect codebook capacity (K) and low-rank dimension (r) on forecasting performance and MAC operations across *ETTh1* and *Weather* datasets with fixed $L = 96$).

ETTh1 **(r/K)**					***Weather*** **(r/K)**				
Ratio (r/K)	**Horizon**	**MSE**	**MAE**	**MACs (M)**	**Ratio (r/K)**	**Horizon**	**MSE**	**MAE**	**MACs (M)**
2/8	96	0.383	0.393	1.40	2/8	96	0.156	0.196	36.78
	192	0.433	0.423	1.75		192	0.205	0.243	37.30
	336	0.479	0.448	2.24		336	0.263	0.287	38.07
	720	0.494	0.474	3.64		720	0.350	0.341	40.14
	Average	0.447	0.435	–		Average	0.244	0.267	–
2/16	96	0.380	0.392	1.82	2/16	96	0.157	0.195	42.30
	192	0.432	0.423	2.17		192	0.205	0.243	42.81
	336	0.479	0.448	2.73		336	0.267	0.289	43.59
	720	0.487	0.471	4.06		720	0.346	0.339	43.65
	Average	0.445	0.433	–		Average	0.244	0.267	–
2/32	96	0.382	0.392	2.80	2/32	96	0.154	0.195	53.33
	192	0.431	0.422	3.15		192	0.204	0.243	53.85
	336	0.471	0.443	3.64		336	0.272	0.292	54.62
	720	0.489	0.469	5.04		720	0.353	0.344	56.68
	Average	0.443	0.432	–		Average	0.246	0.269	–
4/8	96	0.383	0.393	1.82	4/8	96	0.153	0.192	42.29
	192	0.431	0.423	2.17		192	0.209	0.247	42.80
	336	0.475	0.446	2.73		336	0.271	0.292	43.58
	720	0.499	0.479	4.06		720	0.345	0.341	45.64
	Average	0.447	0.435	–		Average	0.245	0.268	–
4/16	96	0.382	0.392	2.80	4/16	96	0.153	0.194	53.31
	192	0.430	0.422	3.08		192	0.204	0.241	53.82
	336	0.468	0.442	3.64		336	0.268	0.289	54.60
	720	0.485	0.468	5.04		720	0.350	0.342	56.66
	Average	0.441	0.431	–		Average	0.244	0.267	–
4/32	96	0.385	0.392	4.62	4/32	96	0.153	0.193	75.35
	192	0.433	0.422	4.97		192	0.204	0.242	75.87
	336	0.475	0.448	5.46		336	0.267	0.287	76.64
	720	0.488	0.468	6.86		720	0.346	0.342	78.70
	Average	0.445	0.433	–		Average	0.243	0.266	–
8/8	96	0.384	0.391	2.80	8/8	96	0.152	0.193	53.30
	192	0.438	0.423	3.08		192	0.206	0.245	53.81
	336	0.475	0.445	3.64		336	0.267	0.289	54.59
	720	0.489	0.471	5.04		720	0.353	0.345	56.65
	Average	0.447	0.433	–		Average	0.245	0.268	–
8/16	96	0.384	0.392	4.62	8/16	96	0.150	0.192	75.30
	192	0.432	0.423	4.97		192	0.205	0.243	75.84
	336	0.475	0.444	5.46		336	0.267	0.289	76.62
	720	0.486	0.467	6.86		720	0.359	0.347	78.68
	Average	0.444	0.432	–		Average	0.245	0.268	–
8/32	96	0.385	0.392	8.26	8/32	96	0.152	0.194	119.39
	192	0.434	0.423	8.61		192	0.206	0.244	119.91
	336	0.474	0.444	9.17		336	0.265	0.286	120.68
	720	0.489	0.468	10.50		720	0.348	0.341	122.74
	Average	0.445	0.432	–		Average	0.243	0.266	–



(a) For the ETTh1 dataset

(b) For the ECL dataset

Figure 6: Across all $\binom{128}{2} = 8{,}128$ centroid combinations for ETTh1 and ECL, empirical pairwise metric preservation by comparing teacher hidden states ($d_{\text{SLM}} = 896$) against projected target representations ($d_{\text{target}} = 32$).

ACKNOWLEDGMENTS

The authors acknowledge the use of Generative AI-based writing assistance for improving sentence structure, grammar and clarity. However, the intellectual contribution, analysis, and interpretations presented in the paper are solely those of the authors.

Table 10: **RAC-Net robustness and multi-seed stability analysis. The experiment is conducted on the *ETTh1* and *Weather* datasets, using varying look-back windows across multiple forecasting horizons. The mean and standard deviation of one horizon across multiple seeds' values are reported in Mean ± Std.**

L	H	ETTh1										Weather									
		Seed 0		Seed 2024		Seed 2025		Seed 2026		Mean ± Std		Seed 0		Seed 2024		Seed 2025		Seed 2026		Mean ± Std	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
96	96	0.382	0.392	0.387	0.393	0.381	0.392	0.384	0.392	$0.384_{\pm.003}$	$0.392_{\pm.001}$	0.153	0.194	0.164	0.206	0.162	0.205	0.161	0.205	$0.160_{\pm.005}$	$0.203_{\pm.006}$
	192	0.430	0.422	0.435	0.422	0.431	0.422	0.433	0.421	$0.432_{\pm.002}$	$0.422_{\pm.001}$	0.204	0.241	0.204	0.249	0.205	0.248	0.205	0.248	$0.205_{\pm.001}$	$0.247_{\pm.004}$
	336	0.468	0.442	0.472	0.442	0.470	0.443	0.473	0.442	$0.471_{\pm.002}$	$0.442_{\pm.001}$	0.268	0.289	0.254	0.286	0.255	0.288	0.255	0.290	$0.258_{\pm.007}$	$0.288_{\pm.002}$
	720	0.485	0.468	0.487	0.469	0.491	0.469	0.490	0.470	$0.488_{\pm.003}$	$0.469_{\pm.001}$	0.350	0.342	0.327	0.339	0.329	0.340	0.331	0.349	$0.334_{\pm.011}$	$0.343_{\pm.005}$
	Avg	0.441	0.431	0.445	0.432	0.443	0.432	0.445	0.431	$\mathbf{0.444}_{\pm.002}$	$\mathbf{0.431}_{\pm.000}$	0.244	0.267	0.237	0.270	0.238	0.270	0.238	0.273	$\mathbf{0.239}_{\pm.003}$	$\mathbf{0.270}_{\pm.003}$
336	96	0.375	0.392	0.372	0.391	0.379	0.399	0.377	0.394	$0.376_{\pm.003}$	$0.394_{\pm.004}$	0.146	0.188	0.152	0.192	0.150	0.195	0.149	0.196	$0.149_{\pm.003}$	$0.193_{\pm.004}$
	192	0.418	0.418	0.410	0.414	0.417	0.418	0.417	0.417	$0.416_{\pm.004}$	$0.417_{\pm.002}$	0.192	0.234	0.195	0.248	0.204	0.255	0.193	0.241	$0.196_{\pm.005}$	$0.245_{\pm.009}$
	336	0.440	0.435	0.442	0.434	0.441	0.435	0.438	0.434	$0.440_{\pm.002}$	$0.435_{\pm.001}$	0.241	0.274	0.255	0.290	0.259	0.307	0.256	0.307	$0.253_{\pm.008}$	$0.295_{\pm.016}$
	720	0.471	0.473	0.452	0.461	0.459	0.466	0.439	0.456	$0.455_{\pm.013}$	$0.464_{\pm.007}$	0.318	0.328	0.327	0.347	0.331	0.348	0.329	0.354	$0.326_{\pm.006}$	$0.344_{\pm.011}$
	Avg	0.426	0.430	0.419	0.425	0.424	0.430	0.418	0.425	$\mathbf{0.422}_{\pm.004}$	$\mathbf{0.427}_{\pm.003}$	0.224	0.256	0.232	0.269	0.236	0.276	0.232	0.275	$\mathbf{0.231}_{\pm.005}$	$\mathbf{0.269}_{\pm.009}$
720	96	0.368	0.398	0.367	0.396	0.384	0.392	0.369	0.398	$0.372_{\pm.008}$	$0.396_{\pm.003}$	0.146	0.191	0.151	0.204	0.148	0.196	0.152	0.201	$0.149_{\pm.003}$	$0.198_{\pm.006}$
	192	0.405	0.424	0.409	0.426	0.433	0.421	0.402	0.419	$0.412_{\pm.014}$	$0.423_{\pm.003}$	0.193	0.237	0.195	0.247	0.200	0.249	0.195	0.243	$0.196_{\pm.003}$	$0.244_{\pm.005}$
	336	0.434	0.444	0.435	0.444	0.473	0.442	0.448	0.453	$0.448_{\pm.018}$	$0.446_{\pm.005}$	0.244	0.275	0.251	0.293	0.249	0.288	0.277	0.366	$0.255_{\pm.015}$	$0.306_{\pm.041}$
	720	0.448	0.465	0.490	0.489	0.490	0.470	0.489	0.489	$0.479_{\pm.021}$	$0.478_{\pm.013}$	0.308	0.323	0.326	0.345	0.326	0.350	0.323	0.347	$0.321_{\pm.009}$	$0.341_{\pm.012}$
	Avg	0.414	0.433	0.425	0.439	0.445	0.431	0.427	0.440	$\mathbf{0.428}_{\pm.013}$	$\mathbf{0.436}_{\pm.004}$	0.223	0.257	0.231	0.272	0.231	0.271	0.237	0.289	$\mathbf{0.230}_{\pm.006}$	$\mathbf{0.272}_{\pm.013}$

Table 11: Multi-seed evaluation ($S \in \{0, 2024, 2025, 2026\}$) of RAC-Net with different configurations on long horizon ($L = 720$) across three benchmark datasets. The mean and standard deviation across four seed are reported.

Method	ETTh1		Weather		Electricity	
	MSE	MAE	MSE	MAE	MSE	MAE
RAC-Net (Full Model)	**0.488 ± 0.003**	**0.469 ± 0.001**	**0.334 ± 0.011**	**0.343 ± 0.005**	**0.190 ± 0.002**	**0.283 ± 0.001**
No Distillation ($\lambda_{\text{distill}} = 0$)	0.491 ± 0.007	0.473 ± 0.004	0.339 ± 0.016	0.345 ± 0.005	0.192 ± 0.001	0.286 ± 0.001
λ_{MSE} Only	0.491 ± 0.011	0.477 ± 0.006	**0.334** ± 0.002	0.366 ± 0.003	0.197 ± 0.007	0.293 ± 0.006
MLP-Statistics Control	0.503 ± 0.033	0.493 ± 0.012	0.347 ± 0.009	0.348 ± 0.001	0.193 ± 0.005	0.285 ± 0.003
Random Target Vectors	0.514 ± 0.048	0.503 ± 0.020	0.356 ± 0.010	0.346 ± 0.005	0.198 ± 0.002	0.287 ± 0.002

Table 12: Full MTSF performance analysis across prediction horizons $H \in \{96, 192, 336, 720\}$ with fixed $L = 96$. The best and second-best results are highlighted in **red**, and underlined, respectively.

Dataset	Horizon	RAC-Net (Ours)		GTR (2026)		LMP-Net (2026)		S-Mamba (2025)		TQNet (2025)		RAFT (2025)		SOFTS (2024)		iTransformer (2024)		PatchTST (2023)		DLinear (2023)	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.382	0.392	0.367	0.391	0.369	0.391	0.386	0.405	0.371	0.393	0.376	0.401	0.381	0.399	0.386	0.405	0.414	0.419	0.386	0.400
	192	0.430	0.422	0.420	0.421	0.421	0.420	0.443	0.437	0.438	0.426	0.417	0.424	0.435	0.431	0.441	0.436	0.460	0.445	0.437	0.432
	336	0.468	0.442	0.476	0.447	0.456	0.448	0.489	0.468	0.476	0.446	0.454	0.441	0.480	0.452	0.487	0.458	0.501	0.466	0.481	0.459
	720	0.485	0.468	0.493	0.476	0.472	0.467	0.502	0.489	0.487	0.470	0.465	0.465	0.499	0.488	0.503	0.491	0.500	0.488	0.519	0.516
	Average	0.441	0.431	0.439	0.434	0.430	0.432	0.455	0.450	0.443	0.434	0.428	0.433	0.449	0.443	0.454	0.448	0.469	0.455	0.456	0.452
ETTh2	96	0.285	0.334	0.290	0.342	0.284	0.337	0.296	0.348	0.295	0.343	0.289	0.342	0.297	0.347	0.297	0.349	0.302	0.348	0.333	0.387
	192	0.364	0.386	0.362	0.389	0.364	0.389	0.376	0.396	0.367	0.393	0.382	0.400	0.373	0.394	0.380	0.400	0.388	0.400	0.477	0.476
	336	0.411	0.422	0.414	0.429	0.407	0.424	0.424	0.431	0.417	0.427	0.424	0.439	0.410	0.426	0.428	0.432	0.426	0.433	0.594	0.541
	720	0.417	0.438	0.423	0.442	0.414	0.438	0.426	0.444	0.433	0.446	0.433	0.457	0.411	0.433	0.427	0.445	0.431	0.446	0.831	0.657
	Average	0.369	0.395	0.372	0.400	0.367	0.397	0.381	0.405	0.378	0.402	0.382	0.410	0.373	0.400	0.383	0.407	0.387	0.407	0.559	0.515
ETTm1	96	0.307	0.343	0.305	0.349	0.317	0.357	0.333	0.368	0.311	0.353	0.325	0.369	0.325	0.361	0.334	0.368	0.329	0.367	0.345	0.372
	192	0.362	0.372	0.349	0.375	0.361	0.382	0.376	0.390	0.356	0.378	0.364	0.388	0.375	0.389	0.377	0.391	0.367	0.385	0.380	0.389
	336	0.385	0.392	0.380	0.398	0.388	0.401	0.408	0.413	0.390	0.401	0.391	0.407	0.405	0.412	0.426	0.420	0.399	0.410	0.413	0.413
	720	0.445	0.430	0.435	0.436	0.451	0.438	0.475	0.448	0.452	0.440	0.444	0.438	0.466	0.447	0.491	0.459	0.454	0.439	0.474	0.453
	Average	0.375	0.384	0.367	0.389	0.379	0.395	0.398	0.405	0.377	0.393	0.381	0.400	0.393	0.402	0.407	0.410	0.387	0.400	0.403	0.407
ETTm2	96	0.175	0.254	0.168	0.249	0.178	0.261	0.179	0.263	0.173	0.256	0.176	0.264	0.180	0.261	0.180	0.264	0.175	0.259	0.193	0.292
	192	0.241	0.299	0.232	0.292	0.240	0.301	0.250	0.309	0.238	0.298	0.241	0.306	0.246	0.306	0.250	0.309	0.241	0.302	0.284	0.362
	336	0.304	0.330	0.287	0.330	0.297	0.339	0.312	0.349	0.301	0.340	0.302	0.345	0.319	0.352	0.311	0.348	0.305	0.343	0.369	0.427
	720	0.404	0.400	0.386	0.390	0.396	0.397	0.411	0.406	0.397	0.396	0.404	0.405	0.405	0.401	0.412	0.407	0.402	0.400	0.554	0.522
	Average	0.281	0.323	0.268	0.315	0.278	0.325	0.288	0.332	0.277	0.323	0.281	0.330	0.287	0.330	0.288	0.332	0.281	0.326	0.350	0.401
Electricity	96	0.136	0.231	0.134	0.229	0.152	0.242	0.139	0.235	0.134	0.229	0.152	0.257	0.143	0.233	0.148	0.240	0.181	0.270	0.197	0.282
	192	0.156	0.250	0.152	0.245	0.165	0.255	0.159	0.255	0.154	0.247	0.156	0.257	0.158	0.248	0.162	0.253	0.188	0.274	0.196	0.285
	336	0.163	0.261	0.171	0.264	0.181	0.272	0.176	0.272	0.169	0.262	0.171	0.272	0.178	0.269	0.178	0.269	0.204	0.293	0.209	0.301
	720	0.189	0.283	0.208	0.300	0.219	0.305	0.204	0.298	0.201	0.294	0.221	0.300	0.218	0.305	0.225	0.317	0.246	0.324	0.245	0.330
	Average	0.161	0.256	0.166	0.260	0.179	0.269	0.170	0.265	0.165	0.258	0.175	0.272	0.174	0.264	0.178	0.270	0.205	0.290	0.212	0.300
Traffic	96	0.424	0.247	0.440	0.263	0.452	0.284	0.382	0.261	0.413	0.261	0.388	0.265	0.376	0.251	0.395	0.268	0.462	0.290	0.650	0.396
	192	0.441	0.267	0.454	0.274	0.463	0.294	0.396	0.267	0.432	0.271	0.400	0.278	0.398	0.261	0.417	0.276	0.466	0.290	0.598	0.370
	336	0.461	0.266	0.472	0.282	0.483	0.304	0.417	0.276	0.450	0.277	0.411	0.287	0.415	0.269	0.433	0.283	0.482	0.300	0.605	0.373
	720	0.546	0.286	0.514	0.301	0.516	0.322	0.460	0.300	0.486	0.295	0.456	0.305	0.447	0.287	0.467	0.302	0.514	0.320	0.645	0.394
	Average	0.468	0.267	0.470	0.280	0.479	0.301	0.414	0.276	0.445	0.276	0.414	0.284	0.409	0.267	0.428	0.282	0.481	0.300	0.625	0.383
Weather	96	0.153	0.194	0.154	0.200	0.163	0.206	0.165	0.210	0.157	0.200	0.185	0.241	0.166	0.208	0.174	0.214	0.177	0.210	0.196	0.255
	192	0.204	0.241	0.202	0.244	0.210	0.249	0.214	0.252	0.206	0.245	0.237	0.287	0.217	0.253	0.221	0.254	0.225	0.250	0.237	0.296
	336	0.268	0.289	0.259	0.287	0.266	0.289	0.274	0.297	0.262	0.287	0.290	0.327	0.282	0.300	0.278	0.296	0.278	0.290	0.283	0.335
	720	0.350	0.342	0.341	0.342	0.345	0.341	0.350	0.345	0.344	0.342	0.367	0.380	0.356	0.351	0.358	0.349	0.354	0.340	0.345	0.381
	Average	0.244	0.267	0.239	0.268	0.246	0.271	0.251	0.276	0.242	0.269	0.270	0.309	0.255	0.278	0.258	0.278	0.259	0.273	0.265	0.317

Table 13: Prompt template and reasoning rationale tokens from Qwen-2.5-0.5B on *Weather* ($L = 96$) for two centroids.

Teacher Prompt Template (ChatML Specification)

```
<|im_start|>system
You are an expert time-series analyst.<|im_end|>
<|im_start|>user
Analyze temporal regime:
Trend: {Trend_Class} (slope {Slope}), Dominant period: {Period} steps, Volatility
(std): {Std}
Provide concise diagnosis.<|im_end|>
<|im_start|>assistant
```

Centroid	**Regime statistics**	N_{tok}	**Generated rationale tokens**
Centroid #0	*Trend*: Downtrend (slope -0.0247), *Period*: 96.0 steps, *Volatility* (σ): 0.914	48	"The analysis indicates a trend in the data with a **negative slope** (-0.0247) and a dominant period of 96.0 steps. The volatility is also moderate at 0.914 standard deviations."
Centroid #1	*Trend*:Stationary (slope -0.0000), *Period*: 12.0 steps, *Volatility* (σ): 0.011	48	"The analysis indicates that the time series exhibits a **stationary trend** with a dominant period of 12.0 steps and a standard deviation of 0.011. This suggests that there is no significant change in the pattern…"