# **Agentic Retrieval-Augmented Generation for Time Series Analysis**

Chidaksh Ravuru IIT Dharwad India 200010046@iitdh.ac.in Sagar Srinivas Sakhinana TCS Research India sagar.sakhinana@tcs.com Venkataramana Runkana TCS Research India venkat.runkana@tcs.com

#### **ABSTRACT**

Time series modeling is crucial for many applications, however, it faces challenges such as complex spatio-temporal dependencies and distribution shifts in learning from historical context to predict task-specific outcomes. To address these challenges, we propose a novel approach using an agentic Retrieval-Augmented Generation (RAG) framework for time series analysis. The framework leverages a hierarchical, multi-agent architecture where the master agent orchestrates specialized sub-agents and delegates the end-user request to the relevant sub-agent. The sub-agents utilize smaller, pre-trained language models (SLMs) customized for specific time series tasks through fine-tuning using instruction tuning and direct preference optimization, and retrieve relevant prompts from a shared repository of prompt pools containing distilled knowledge about historical patterns and trends to improve predictions on new data. Our proposed modular, multi-agent RAG approach offers flexibility and achieves state-of-the-art performance across major time series tasks by tackling complex challenges more effectively than task-specific customized methods across benchmark datasets.

#### **KEYWORDS**

Time Series Analysis, Retrieval Augmented Generation

# 1 INTRODUCTION

Time series modeling underpins a vast spectrum of real-world applications, including demand planning [21], anomaly detection [54], inventory management [52], energy load forecasting [24], weather modeling [31], and many others. However, it is not without its challenges. High dimensionality, non-linearity, sparsity, and distribution shifts all pose significant hurdles. Successfully navigating these challenges in time series analysis applications necessitates both considerable domain knowledge and the design of neural network architectures tailored to address task-specific goals, leading to better performance. In contrast to task-specific approaches, which employ different architecture designs for time series analysis, foundational pretrained large language models (LLMs), such as OpenAI's GPT-4 [29] and Google's Gemini [34, 39], with their strong generalization and logical reasoning capabilities, have shown remarkable versatility across a broad spectrum of natural language processing (NLP) tasks, requiring minimal fine-tuning[17] or only a few demonstrations[2] for adaptation to niche tasks. Open-source, small-scale pretrained language models (SLMs), such as Google Gemma ([40]) and Meta LLaMA ([1, 41]), offer cost-effective domain customization through Parameter Efficient Fine-Tuning (PEFT) ([15, 16]) techniques using task-specific labeled datasets. Additionally, these smaller models can be further aligned with human preferences using Direct Preference Optimization (DPO) [8], a fine-tuning technique that utilizes paired preference data, such as datasets of preferred and dispreferred responses. However, SLMs may lack the reasoning and generalization capabilities of large-scale proprietary language

models. The potential of foundational SLMs designed for universal time series applications (a single-model-fits-all approach), such as diverse time series tasks like classification, anomaly detection, forecasting, imputation, and others, remains largely unexplored but holds great promise. This approach contrasts sharply with the traditional approach of using customized, task-specific methods ([43, 49, 50]) for time series modeling for various applications. Adapting SLMs designed for NLP tasks for time series modeling to capture trends and patterns within the complex data, though unconventional, offers a clear possibility for providing unique insights. However, this is a challenging task as SLMs are trained primarily on text corpora, which operates on discrete tokens, while time series data is inherently continuous. Furthermore, SLMs may lack the inherent ability to detect and interpret time series patterns and trends like seasonality, cyclicity, or outliers, due to the absence of related pretraining knowledge. Moreover, current LMs designed for time series analysis ([14, 20, 56]) rely on a fixed-length window of past observations to generate predictions, which may be inadequate for capturing complex patterns and trends present in time series data, thus hindering accurate modeling. Smaller window sizes may capture local patterns but miss broader trends, while larger window sizes can capture more context but may overlook finer details. In recent times, Retrieval-Augmented Generation (RAG) or Retrieval-Augmented Language Modeling (RALM)[23, 33, 37] combines pre-trained language models with information retrieval from external knowledge bases to augment text generation capabilities for open-ended question-answering(ODQA)[38] tasks or for improved language modeling for text summarization, completion with improved accuracy. While regular RAG methods augment generation with retrieved knowledge for ODQA tasks, Agentic RAGs take this further by being instruction-following agents that can tackle complex goals through multi-step reasoning and iterative refinement cycles using repeated retrievals over a knowledge base to ensure the final response aligns with the end user request. In this work, we propose an Agentic RAG framework for time series analysis to improve task-specific outcomes by addressing challenges like distributional shifts, fixed window limitations in time series data. Figure 1 illustrates the framework. Our Agentic RAG framework presents a hierarchical, multi-agent architecture composed of a master (top-level) agent and specialized sub-agents customized for specific time series tasks. The top-level agent acting as the orchestrator analyzes the incoming user request, determines its nature and complexity, and then routes (or delegates) it to the corresponding task-specific sub-agent to produce the desired output. Similarly to how regular RAG frameworks retrieve relevant information from external knowledge bases like documents, databases, or access the real world through APIs, this Agentic RAG framework leverages distinct prompt pools as internal knowledge bases for each sub-agent focused on specific time series tasks. As specialized

knowledge repositories tailored to each sub-agent's time series task, the prompt pools store both domain and task-specific knowledge as key-value pairs. This facilitates easy reuse and sharing within and across datasets, promoting knowledge sharing and transfer, reducing the need to relearn or rediscover patterns from scratch. Each 'key' represents a specific pattern (seasonality, cyclicality, etc.), and the 'value' contains details about that pattern. When processing new input data, the sub-agent retrieves the most relevant prompts from the pool based on similarity. These prompts provide contextual knowledge about related historical patterns and trends, improving generalization to new scenarios. This knowledge-augmentation approach, by conditioning on past patterns, allows the sub-agent access to a broad spectrum of task-specific knowledge regardless of historical occurrence, enabling it to learn and adapt to diverse trends within complex data for improved predictions. Each subagent utilizes pre-trained, SLMs like Gemma[40] and Llama 3[1]. We fine-tune each SLM using instruction-tuning on task-specific datasets and optimize them for time series tasks such as forecasting, imputation, or other related tasks. Additionally, we fine-tune using DPO[8] through a dynamic masking technique to align the SLMs task-specific outputs to preferred and non-preferred outcomes, providing adversarial feedback[47] through a binary classification task. The master agent for sub-agent orchestration utilizes the 'ReAct' prompting technique[45], encouraging the general-purpose SLM to think step-by-step and use external tools (sub-agents, each utilizing a fine-tuned SLM for specific time series tasks) to generate responses. The master agent can even chain sub-agents together to handle complex, multi-step time series analysis tasks, addressing more intricate challenges. However, in this work, the sub-agents operate in isolation, each handling only a single, specific task.

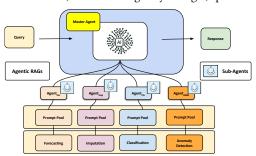


Figure 1: The figure illustrates the proposed agentic RAG framework, designed to handle diverse time series analysis tasks. The framework employs a hierarchical, multi-agent architecture. A master agent receives end-user questions and routes them to appropriate specialized sub-agents based on the specific time series task (e.g., forecasting, imputation, classification, anomaly detection). The sub-agents utilize pretrained SLMs fine-tuned on task-specific datasets using techniques like instruction tuning and direct preference optimization to capture spatio-temporal dependencies within and across the time series datasets. Each sub-agent maintains its own prompt pool as 'key-value' pairs, which stores relevant historical knowledge related to specific trends and patterns within its respective specialized domain. This allows the sub-agents to leverage related past experiences for improved task-specific predictions on new, similar data, and is then relayed back to the user through the master agent.

In summary, the master agent orchestrates sub-agents, selects the most appropriate sub-agent, and allocates the task to the specialized sub-agent. The sub-agent retrieves relevant information from a shared knowledge base of prompt pools and generates an output based on the retrieved information. The differentiable prompt pools for each sub-agent, acting as specialized dynamic knowledge repositories, provide the necessary historical context and understanding to effectively analyze new input data for their designated tasks. The master agent gathers responses from the chosen subagent and synthesize these responses to produce a comprehensive answer for the end-user query. The hierarchical, multi-agent architecture for time series analysis offers key advantages. It enables modularity, flexibility, and accuracy by allowing specialized subagents to focus on specific tasks, be updated independently, and be dynamically allocated by the meta-agent to generate comprehensive results. Extensive empirical studies demonstrate that the Agentic-RAG framework achieves performance on par with, or even surpassing, state-of-the-art methods across multiple time series analysis tasks for both univariate and multivariate datasets. The multi-agent approach tackles the diverse and complex challenges of time series analysis, unlike a single, universal agent that attempts to be a jack-of-all-trades for all time series tasks.

#### 2 PROBLEM FORMULATION

Consider a time series dataset characterized by N univariate time series, with sequential data collected over T timestamps, represented as a data matrix  $\mathbf{X} \in \mathbb{R}^{N \times T}$ . Each row in this matrix represents a univariate time series, and each column corresponds to data collected at a specific timestamp. To refer to data from a specific time series or timestamp, we use subscripts and superscripts, respectively. For instance,  $X_i = \mathbf{X}_{i,:}$  denotes the data from the i-th time series, and  $X^t = \mathbf{X}_{::t}$  denotes the data at timestamp t.

#### 2.1 Forecasting

We utilize a sliding window[10, 46] of size  $\tau$ , to construct time series subsequences  $S^t = X^{t-\tau+1:t} \in \mathbb{R}^{N \times \tau}$ , which have been observed over previous  $\tau$ -steps prior to current time step t to predict about the future values for the next v-steps,  $S^{t+1} = X^{t+1:t+\nu} \in \mathbb{R}^{N \times \nu}$ .

# 2.2 Missing Data Imputation

We utilize a binary mask matrix  $\mathbf{M} \in \{0,1\}^{N \times T}$ , where  $M_{i,t} = 0$  indicates that the value  $X_{i,t}$  is missing, and  $M_{i,t} = 1$  indicates that the value is observed in the data matrix  $\mathbf{X} \in \mathbb{R}^{N \times T}$ . Missing data can follow random or block patterns[4, 26, 27] across the N univariate time series and T timestamps. We utilize observed values  $\mathbf{X}_{\text{obs}} = \mathbf{X} \odot \mathbf{M}$  to estimate the missing values  $\mathbf{X}_{\text{miss}} = \mathbf{X} \odot (1 - \mathbf{M})$ .  $\odot$  denotes element-wise multiplication. We utilize a sliding window of size  $\tau$  over the observed samples  $\mathbf{X}_{\text{obs}}$ , to construct subsequences  $S_{\text{obs}}^t = \mathbf{X}_{\text{obs}}^{t-\tau+1:t} \in \mathbb{R}^{N \times \tau}$ , which have been observed over previous  $\tau$ -steps prior to the current time step t. These observed samples are used to predict the missing values for the next  $\nu$ -steps,  $S_{\text{miss}}^{t+1} = X_{\text{miss}}^{t+1:t+\nu} \in \mathbb{R}^{N \times \nu}$  by leveraging spatio-temporal dependencies within the data.

# 2.3 Anomaly Detection

Assuming the time series dataset exhibits normal behavior during the initial  $T_{\text{train}}$  timestamps, any pattern deviating from the normal behavior in subsequent timestamps  $t > T_{\text{train}}$  is anomalous. Data observed after  $T_{\text{train}}$  is considered the test dataset. We use a sliding window to construct samples from previous time steps  $S^t \in \mathbb{R}^{N \times \tau}$  to predict future values of multiple time series  $S^{t+1} \in \mathbb{R}^{N \times \tau}$ . The

framework predictions are denoted by  $\hat{S}^{t+1} \in \mathbb{R}^{N \times \nu}$ . In the unsupervised anomaly detection task, it computes the robust normalized anomaly scores  $(A_i^{t+1})$  for each variable i across the time steps in the training set  $\mathcal{T}_{train}$ . This information regarding the variables helps in accurately localizing the anomalies within the test set.

$$A_i^{t+1} = |S_i^{t+1} - \hat{S}_i^{t+1}|$$

 $A_i^{t+1} = \left| \mathbf{S}_i^{t+1} - \hat{\mathbf{S}}_i^{t+1} \right|$  We compute the simple moving average of the maximum value of anomalousness  $score(A_i^{t+1})$  across the multiple variables at time

point 
$$t + 1$$
 over the validation set as given,  

$$Th = \max_{t \in \mathcal{T}_{val}} A^{t+1}; A^{t+1} = \frac{1}{w_a} \sum_{t - (w_a + 1)}^{t+1} \max_{i \in |N|} (A_i^{t+1})$$
(1)

where  $w_a$  denotes the number of time points in the moving average calculation.  $\mathcal{T}_{val}$  denotes the time points in the validation set. We set the anomaly detection threshold(Th) as the moving averaged maximum anomaly value for time t + 1,  $A^{t+1}$  over the validation data. During inference, time points with an anomaly score above the threshold were flagged as anomalies.

#### 2.4 Classification

We perform unsupervised K-means clustering, identifying (K) optimal clusters or regimes and assigning cluster labels  $C \in \mathbb{R}^T$  to each time point in the data matrix  $X \in \mathbb{R}^{N \times T}$ . Then, a sliding window approach is employed to predict the cluster labels for the next  $\nu$  steps  $S^{t+1} = X^{t+1:t+\nu} \in \mathbb{R}^{N \times \nu}$  based on the observed sample  $S^t = X^{t-\tau+1:t} \in \mathbb{R}^{N \times \tau}$  over the previous  $\tau$  time steps.

#### PROPOSED METHOD

The proposed framework offers a novel approach to time series analysis by leveraging a hierarchical, multi-agent architecture. It comprises a master agent that coordinates specialized sub-agents, each dedicated to a specific time series task such as forecasting, anomaly detection, or imputation. These sub-agents employ pre-trained language models and utilize prompt pools as internal knowledge bases, storing key-value pairs representing historical patterns and trends. By retrieving relevant prompts from these pools, the sub-agents can augment their predictions with contextual knowledge about related past patterns, enabling them to adapt to diverse trends within complex time series data. The framework's modular design, combined with the strengths of individual sub-agents, allows for improved performance across various time series analysis tasks, surpassing the limitations of traditional fixed-window methods.

# **Dynamic Prompting Mechansim**

Current time series methods typically utilize past data within a predefined window length to understand historical trends and predict task-specific outcomes. However, this approach may not be optimal because there is no universally ideal window length for all time series data. A larger window length might obscure shortrange dependencies, while a smaller window length might fail to capture long-range dependencies . Existing methods fail to capture the full complexity of diverse trends and patterns within the complex data required for accurate time series modeling. Adjusting the window length in real-world scenarios can be challenging and computationally expensive. Achieving this goal is an ambitious task, given the current state of research in this field. To address the challenges of non-stationarity and distributional shifts in real-world data, we utilize a differentiable dynamic prompting mechanism[3]. This mechanism allows traditional time series methods to access

related past knowledge by retrieving the same group of prompts from the prompt pool for effective adaptive learning on new, similar input data. The dynamic prompting approach utilizes a shared pool of prompts stored as key-value pairs. For time series applications, each prompt is represented by a key vector encoding the essential global characteristics associated with that prompt. The corresponding value matrix contains specific knowledge related to those trends or patterns, such as seasonality, cyclicality, irregularities, and other effects. The key vector acts as an identifier or query vector to retrieve relevant prompts from the pool based on similarity to the input new data, providing a form of conditioning or context about historical patterns to enhance the predictions. This allows the time series methods to effectively leverage encoded knowledge from past experiences, enhancing their predictions by recognizing and applying learned patterns from the shared prompt pool to the new input data. The pool of prompts  $\mathcal{P}$  contains a set of M distinct key-value pairs as follows:

$$\mathcal{P} = (k_1, v_1), (k_2, v_2), \dots, (k_M, v_M)$$

 $\mathcal{P} = (k_1, v_1), (k_2, v_2), \dots, (k_M, v_M)$  Here, M is the total number of prompts in the pool,  $k_m \in \mathbb{R}^d$ is the key vector of the *m*-th prompt, and  $v_m \in \mathbb{R}^{l \times d}$  is the corresponding prompt value matrix with length l and dimensionality d. In order to retrieve the most relevant prompts for a given input time series  $S_i^t = X_i^{t-\tau+1:t} \in \mathbb{R}^{\tau}$ , we first linearly project it into ddimensional embeddings  $S_i^t \in \mathbb{R}^d$ . We then utilize a score-matching function  $\gamma$  to measure the similarity between the input and each prompt key:  $\gamma\left(S_{i}^{t}, k_{m}\right) = \frac{S_{i}^{t} \cdot k_{m}}{|S_{i}^{t}||k_{m}|}$ 

where  $\gamma$  computes the cosine similarity between the input embedding  $S_i^t$  and the prompt key  $\mathbf{k}_m$ . The top-K prompts with the highest similarity scores are selected, where  $1 \le K \le M$ . Let  $\mathcal{J} = j_1, j_2, \dots, j_K$  be the set of indices corresponding to the top-Kmost relevant prompts retrieved from the pool  ${\mathcal P}$  for the given input time series  $S_i^t$ . The selected prompts, along with the original input, are concatenated to form the input embedding  $S_i^t$  as follows:

$$S_i^t = \left[v_{j_1}; \dots; v_{j_K}; S_i^t\right]$$

 $S_i^t = \left[v_{j_1}; \dots; v_{j_K}; S_i^t\right]$  where  $\mathbf{s}_i^t \in \mathbb{R}^{(Kl+1) \times d}$ . We linearly project  $\mathbf{s}_i^t$  to d-dimensional representation as follows:

where  $W \in \mathbb{R}^{d \times (Kl+1)d}$  is a learnable weight matrix. In summary, it aims to improve time series modeling efficiency on the task-specific performance by allowing the framework to recognize and apply learned patterns across non-stationarity datasets with distributional shifts via the shared prompt representation pool.

#### 3.2 Fine-Tuning/Preference Optimization SLMs

Current pretrained SLMs, such as Google's Gemma and Meta's Llama-3 models, are designed with a context length of 8K tokens. However, they struggle to process long input sequences that exceed their pretraining context window. This is because the limited length of the context window during pretraining restricts their effectiveness during inference when dealing with longer texts. SLMs with an improved context length can better capture long-term spatio-temporal dependencies and complex patterns that unfold over extended periods, which is essential for accurate predictions and understanding seasonal or cyclic trends. We build upon recent work [19] to improve how SLMs handle long sequences without finetuning. A two-tiered attention mechanism (grouped and neighbor

attention) allows SLMs to process unseen long-range dependencies, enabling SLMs to naturally handle extended text and maintain performance. It outperforms fine-tuning methods on multiple NLP benchmarks, demonstrating a significant step forward for SLMs in managing long text sequences. Nevertheless, fine-tuning generalpurpose SLMs on task-specific data and objectives can still provide significant performance gains and allow for customization and adaptation to the unique challenges and requirements of different time series analysis tasks. Instruction-tuning of SLMs captures complex task-specific spatio-temporal dependencies and improves prediction accuracy. We perform instruction-tuning of SLMs with an improved context length [19](32K tokens) using parameter-efficient fine-tuning (PEFT) techniques on their associated specific tasks (e.g., forecasting, imputation) using the corresponding time-series datasets. This approach could significantly enhance the effectiveness of SLMs in processing extensive time-series data. We leverage Direct Preference Optimization (DPO; [32]), which involves randomly masking 50 % of the data and performing binary classification task to predict the corresponding correct task-specific outcomes. This is done to steer the predictions of the SLMs toward more reliable outcomes in the specific context of time series analysis, favoring preferred responses over dispreferred responses.

#### 4 EXPERIMENTS

Datasets: We evaluate the proposed Agentic-RAG framework on four tasks: forecasting, classification, anomaly detection, and imputation. To comprehensively evaluate the framework performance against several baselines, we conducted experiments using both univariate and multivariate benchmark datasets across multiple time series tasks. The variants include Agentic-RAG with SelfExtend-Gemma-2B-instruct, Gemma-7B-instruct, and Llama 3-8B-instruct. We utilized several real-world traffic-related datasets (PeMSD3, PeMSD4, PeMSD7, PeMSD7(M), PeMSD8) obtained from the Caltrans Performance Measurement System (PeMS) [5] for forecasting, classification, and imputation. To ensure consistency with prior research[7], these datasets are preprocessed by aggregating 30-second data points into 5-minute averages. Additionally, publicly available traffic prediction datasets (METR-LA, PEMS-BAY) [22] are utilized, with data aggregated into 5-minute intervals, resulting in 288 observations per day. Table 1 provides comprehensive details regarding the spatiotemporal multivariate datasets. For anomaly detection, we evaluate the proposed Agentic-RAG framework on publicly available multivariate datasets, conducting a comprehensive benchmark comparison against baseline methods. Table 2 provides an overview of the datasets used in this study. SWaT and WADI<sup>1</sup> are real-world datasets on water treatment facilities and distribution networks, respectively. SMAP and MSL are expert annotated opensource datasets of telemetry data sourced from NASA[18]. The Tennessee Eastman Process (TEP)<sup>2</sup> dataset is a simulated industrial benchmark designed for process monitoring and control, comprising 20 distinct fault types. The HAI<sup>3</sup> dataset comprises time-series data from an industrial testbed for detecting adversarial attacks on industrial control systems, involving steam-turbine power generation and pumped-storage hydropower generation processes, with

38 different attack scenarios. In addition, we discuss the univariate datasets for forecasting and imputation in the technical appendix.

| Dataset   | Sensors | Timesteps | Time-Range        | Data Split | Granularity |
|-----------|---------|-----------|-------------------|------------|-------------|
| PeMSD3    | 358     | 26,208    | 09/2018 - 11/2018 |            |             |
| PeMSD4    | 307     | 16,992    | 01/2018 - 02/2018 |            |             |
| PeMSD7    | 883     | 28,224    | 05/2017 - 08/2017 | 6 / 2 / 2  | Us.         |
| PeMSD8    | 170     | 17,856    | 07/2016 - 08/2016 |            | mins        |
| PeMSD7(M) | 228     | 12,672    | 05/2012 - 06/2012 |            | st          |
| METR-LA   | 207     | 34,272    | 03/2012 - 06/2012 | 7/1/2      |             |
| PEMS-BAY  | 325     | 52,116    | 01/2017 - 05/2017 | //1/2      |             |

Table 1: Summary of the spatio-temporal datasets.

| Dataset | SWaT | WADI | SMAP | MSL | TEP | HAI |
|---------|------|------|------|-----|-----|-----|
| Sensors | 51   | 123  | 25   | 55  | 52  | 59  |
| τ       | 25   | 25   | 50   | 55  | 35  | 30  |

Table 2: Statistical summary of benchmark datasets.  $\tau$  is the length of subsequences or historical window length.

**Evaluation Metrics:** For forecasting and imputation tasks, the performance of the proposed framework is evaluated using MAE, RMSE, and MAPE metrics on the original scale of the time series data. For classification tasks, we use accuracy. For anomaly detection, we utilize the standard evaluation metrics of precision (P in %), recall (R in %), and F1-score (F1 in %). We utilize a multi-metric approach for a fair and rigorous comparison with baseline models. To do this, we compute the confusion matrix: true positive (TP) for correctly detected anomalies, false negative (FN) for undetected anomalies, true negative (TN) for correctly identified normal points, and false positive (FP) for normal points mistakenly identified as anomalies. Precision (TP/(FP + TP)) represents the proportion of correctly detected anomalies among all identified anomalies, while recall (TP / (FN + TP)) represents the proportion of all true anomalies that were correctly detected. The F1-score is calculated as the harmonic mean of precision and recall. The threshold for identifying anomalies is set to the highest anomaly score(refer to Section 2.3) from the validation dataset. For the SWaT and WADI datasets, which contain contiguous anomaly segments, we adopt the point adjustment strategy [36, 51] to flag the entire subsequence as an anomaly if the model predicts one. On the Tennessee Eastman dataset, we utilize the Fault Detection Rate (FDR, in %), defined as the ratio of the number of faults detected to the total number of faults that occur, to evaluate the effectiveness of our framework.

Experimental Settings: To reduce memory footprint and computational complexity, we segment the time series datasets using a sliding window technique with a predefined historical window size to obtain time series subsequences (smaller, overlapping sequences of a fixed length). We performed instruction-tuning(fine-tuning) of the small-scale language models, such as SelfExtend-Instruct LLaMA 3-8B, Gemma-2B, and Gemma-7B models using the PEFT technique[44] such as QLoRA[12], on their specific associated time series tasks using corresponding datasets. We set the following hyperparameters: a batch size of 16, a sequence length of 32K, a learning rate of 1e-5, training for 15 epochs, 500 warmup steps, a weight decay of 0.01, and a gradient accumulation of 2 steps. We used the AdamW optimizer[25] and a linear scheduler to adjust the learning rate during training. We utilized a 4-bit quantization for QLoRA. The QLoRA hyperparameters include the low-rank(r) of

 $<sup>^{1}</sup>https://itrust.sutd.edu.sg/itrust-labs/datasets/\\$ 

<sup>&</sup>lt;sup>2</sup>https://dataverse.harvard.edu/dataverse/harvard

<sup>&</sup>lt;sup>3</sup>https://github.com/icsdataset/hai

| Methods                               |       | PeMSD | )3    |       | PeMSD | 4     |       | PeMSD | 7     |       | PeMSI | 08    | P    | eMSD7 | (M)   |
|---------------------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|------|-------|-------|
| Wethous                               | MAE   | RMSE  | MAPE  | MAE  | RMSE  | MAPE  |
| HA                                    | 31.58 | 52.39 | 33.78 | 38.03 | 59.24 | 27.88 | 45.12 | 65.64 | 24.51 | 34.86 | 59.24 | 27.88 | 4.59 | 8.63  | 14.35 |
| ARIMA                                 | 35.41 | 47.59 | 33.78 | 33.73 | 48.80 | 24.18 | 38.17 | 59.27 | 19.46 | 31.09 | 44.32 | 22.73 | 7.27 | 13.20 | 15.38 |
| VAR                                   | 23.65 | 38.26 | 24.51 | 24.54 | 38.61 | 17.24 | 50.22 | 75.63 | 32.22 | 19.19 | 29.81 | 13.10 | 4.25 | 7.61  | 10.28 |
| FC-LSTM                               | 21.33 | 35.11 | 23.33 | 26.77 | 40.65 | 18.23 | 29.98 | 45.94 | 13.20 | 23.09 | 35.17 | 14.99 | 4.16 | 7.51  | 10.10 |
| TCN                                   | 19.32 | 33.55 | 19.93 | 23.22 | 37.26 | 15.59 | 32.72 | 42.23 | 14.26 | 22.72 | 35.79 | 14.03 | 4.36 | 7.20  | 9.71  |
| TCN(w/o causal)                       | 18.87 | 32.24 | 18.63 | 22.81 | 36.87 | 14.31 | 30.53 | 41.02 | 13.88 | 21.42 | 34.03 | 13.09 | 4.43 | 7.53  | 9.44  |
| GRU-ED                                | 19.12 | 32.85 | 19.31 | 23.68 | 39.27 | 16.44 | 27.66 | 43.49 | 12.20 | 22.00 | 36.22 | 13.33 | 4.78 | 9.05  | 12.66 |
| DSANet                                | 21.29 | 34.55 | 23.21 | 22.79 | 35.77 | 16.03 | 31.36 | 49.11 | 14.43 | 17.14 | 26.96 | 11.32 | 3.52 | 6.98  | 8.78  |
| STGCN                                 | 17.55 | 30.42 | 17.34 | 21.16 | 34.89 | 13.83 | 25.33 | 39.34 | 11.21 | 17.50 | 27.09 | 11.29 | 3.86 | 6.79  | 10.06 |
| DCRNN                                 | 17.99 | 30.31 | 18.34 | 21.22 | 33.44 | 14.17 | 25.22 | 38.61 | 11.82 | 16.82 | 26.36 | 10.92 | 3.83 | 7.18  | 9.81  |
| GraphWaveNet                          | 19.12 | 32.77 | 18.89 | 24.89 | 39.66 | 17.29 | 26.39 | 41.50 | 11.97 | 18.28 | 30.05 | 12.15 | 3.19 | 6.24  | 8.02  |
| ASTGCN(r)                             | 17.34 | 29.56 | 17.21 | 22.93 | 35.22 | 16.56 | 24.01 | 37.87 | 10.73 | 18.25 | 28.06 | 11.64 | 3.14 | 6.18  | 8.12  |
| MSTGCN                                | 19.54 | 31.93 | 23.86 | 23.96 | 37.21 | 14.33 | 29.00 | 43.73 | 14.30 | 19.00 | 29.15 | 12.38 | 3.54 | 6.14  | 9.00  |
| STG2Seq                               | 19.03 | 29.83 | 21.55 | 25.20 | 38.48 | 18.77 | 32.77 | 47.16 | 20.16 | 20.17 | 30.71 | 17.32 | 3.48 | 6.51  | 8.95  |
| LSGCN                                 | 17.94 | 29.85 | 16.98 | 21.53 | 33.86 | 13.18 | 27.31 | 41.46 | 11.98 | 17.73 | 26.76 | 11.20 | 3.05 | 5.98  | 7.62  |
| STSGCN                                | 17.48 | 29.21 | 16.78 | 21.19 | 33.65 | 13.90 | 24.26 | 39.03 | 10.21 | 17.13 | 26.80 | 10.96 | 3.01 | 5.93  | 7.55  |
| AGCRN                                 | 15.98 | 28.25 | 15.23 | 19.83 | 32.26 | 12.97 | 22.37 | 36.55 | 9.12  | 15.95 | 25.22 | 10.09 | 2.79 | 5.54  | 7.02  |
| STFGNN                                | 16.77 | 28.34 | 16.30 | 20.48 | 32.51 | 16.77 | 23.46 | 36.60 | 9.21  | 16.94 | 26.25 | 10.60 | 2.90 | 5.79  | 7.23  |
| STGODE                                | 16.50 | 27.84 | 16.69 | 20.84 | 32.82 | 13.77 | 22.59 | 37.54 | 10.14 | 16.81 | 25.97 | 10.62 | 2.97 | 5.66  | 7.36  |
| Z-GCNETs                              | 16.64 | 28.15 | 16.39 | 19.50 | 31.61 | 12.78 | 21.77 | 35.17 | 9.25  | 15.76 | 25.11 | 10.01 | 2.75 | 5.62  | 6.89  |
| STG-NCDE                              | 15.57 | 27.09 | 15.06 | 19.21 | 31.09 | 12.76 | 20.53 | 33.84 | 8.80  | 15.45 | 24.81 | 9.92  | 2.68 | 5.39  | 6.76  |
| SelfExtend-Agentic-RAG W/Gemma-2B     | 14.05 | 20.53 | 11.57 | 19.14 | 27.92 | 10.54 | 20.59 | 31.89 | 9.27  | 15.53 | 22.17 | 8.09  | 2.10 | 5.06  | 6.61  |
| SelfExtend-Agentic-RAG W/Gemma-7B     | 13.51 | 20.02 | 10.98 | 17.99 | 25.97 | 10.03 | 19.48 | 30.53 | 8.47  | 14.52 | 21.49 | 7.46  | 2.38 | 4.79  | 6.02  |
| SelfExtend-Agentic-RAG W/Llama 3 - 8B | 13.01 | 19.48 | 10.53 | 17.46 | 25.54 | 9.52  | 19.02 | 29.97 | 8.03  | 14.03 | 20.98 | 7.04  | 2.33 | 4.68  | 5.88  |

Table 3: The table compares various methods for 12-sequence-to-12-sequence forecasting tasks on benchmark datasets using multiple evaluation metrics. These methods use 12 past sequences to predict the next 12 sequences.

| Datasata | M-41-1-                             |       | orizon |       |       | orizon( |       |       | rizon( | <b>9</b> 12 |
|----------|-------------------------------------|-------|--------|-------|-------|---------|-------|-------|--------|-------------|
| Datasets | Methods                             | RMSE  | MAE    | MAPE  | RMSE  | MAE     | MAPE  | RMSE  | MAE    | MAPE        |
|          | HA                                  | 10.00 | 4.79   | 11.70 | 11.45 | 5.47    | 13.50 | 13.89 | 6.99   | 17.54       |
|          | VAR                                 | 7.80  | 4.42   | 13.00 | 9.13  | 5.41    | 12.70 | 10.11 | 6.52   | 15.80       |
|          | SVR                                 | 8.45  | 3.39   | 9.30  | 10.87 | 5.05    | 12.10 | 13.76 | 6.72   | 16.70       |
|          | FC-LSTM                             | 6.30  | 3.44   | 9.60  | 7.23  | 3.77    | 10.09 | 8.69  | 4.37   | 14.00       |
|          | DCRNN                               | 5.38  | 2.77   | 7.30  | 6.45  | 3.15    | 8.80  | 7.60  | 3.60   | 10.50       |
|          | STGCN                               | 5.74  | 2.88   | 7.62  | 7.24  | 3.47    | 9.57  | 9.40  | 4.59   | 12.70       |
| METR-LA  | Graph WaveNet                       | 5.15  | 2.69   | 6.90  | 6.22  | 3.07    | 8.37  | 7.37  | 3.53   | 10.01       |
|          | ASTGCN                              | 9.27  | 4.86   | 9.21  | 10.61 | 5.43    | 10.13 | 12.52 | 6.51   | 11.64       |
|          | STSGCN                              | 7.62  | 3.31   | 8.06  | 9.77  | 4.13    | 10.29 | 11.66 | 5.06   | 12.91       |
|          | MTGNN                               | 5.18  | 2.69   | 6.88  | 6.17  | 3.05    | 8.19  | 7.23  | 3.49   | 9.87        |
|          | GMAN                                | 5.55  | 2.80   | 7.41  | 6.49  | 3.12    | 8.73  | 7.35  | 3.44   | 10.07       |
|          | DGCRN                               | 5.01  | 2.62   | 6.63  | 6.05  | 2.99    | 8.02  | 7.19  | 3.44   | 9.73        |
|          | SelfExtend-Agentic-RAG W/Gemma-2B   | 4.52  | 2.29   | 5.55  | 5.82  | 2.91    | 7.33  | 6.81  | 3.32   | 9.03        |
|          | SelfExtend-Agentic-RAG W/Gemma-7B   | 4.28  | 2.17   | 5.35  | 5.63  | 2.75    | 7.02  | 6.53  | 3.23   | 8.71        |
|          | SelfExtend-Agentic-RAG W/Llama 3-8B | 4.03  | 2.02   | 5.05  | 5.43  | 2.61    | 6.75  | 6.23  | 3.12   | 8.53        |
|          | HA                                  | 4.30  | 1.89   | 4.16  | 5.82  | 2.50    | 5.62  | 7.54  | 3.31   | 7.65        |
|          | VAR                                 | 3.16  | 1.74   | 3.60  | 4.25  | 2.32    | 5.00  | 5.44  | 2.93   | 6.50        |
|          | SVR                                 | 3.59  | 1.85   | 3.80  | 5.18  | 2.48    | 5.50  | 7.08  | 3.28   | 8.01        |
|          | FC-LSTM                             | 4.19  | 2.05   | 4.80  | 4.55  | 2.20    | 5.20  | 4.96  | 2.37   | 5.70        |
|          | DCRNN                               | 2.95  | 1.38   | 2.90  | 3.97  | 1.74    | 3.90  | 4.74  | 2.07   | 4.90        |
|          | STGCN                               | 2.96  | 1.36   | 2.90  | 4.27  | 1.81    | 4.17  | 5.69  | 2.49   | 5.79        |
| PEMS-BAY | Graph WaveNet                       | 2.74  | 1.30   | 2.73  | 3.70  | 1.63    | 3.67  | 4.52  | 1.95   | 4.63        |
|          | ASTGCN                              | 3.13  | 1.52   | 3.22  | 4.27  | 2.01    | 4.48  | 5.42  | 2.61   | 6.00        |
|          | STSGCN                              | 3.01  | 1.44   | 3.04  | 4.18  | 1.83    | 4.17  | 5.21  | 2.26   | 5.40        |
|          | MTGNN                               | 2.79  | 1.32   | 2.77  | 3.74  | 1.65    | 3.69  | 4.49  | 1.94   | 4.53        |
|          | GMAN                                | 2.91  | 1.34   | 2.86  | 3.76  | 1.63    | 3.68  | 4.32  | 1.86   | 4.37        |
|          | DGCRN                               | 2.69  | 1.28   | 2.66  | 3.63  | 1.59    | 3.55  | 4.42  | 1.89   | 4.43        |
|          | SelfExtend-Agentic-RAG W/Gemma-2B   | 1.81  | 0.91   | 1.82  | 2.71  | 1.31    | 2.71  | 3.31  | 1.72   | 3.32        |
|          | SelfExtend-Agentic-RAG W/Gemma-7B   | 1.72  | 0.86   | 1.68  | 2.61  | 1.26    | 2.63  | 3.21  | 1.67   | 3.23        |
|          | SelfExtend-Agentic-RAG W/Llama 3-8B | 1.62  | 0.81   | 1.63  | 2.52  | 1.21    | 2.51  | 3.12  | 1.62   | 3.14        |

Table 4: The table compares the performance of various forecasting methods on the METR-LA and PEMS-BAY benchmark datasets using multiple evaluation metrics. All methods use 12 past sequences to predict 3, 6, or 12 future sequences.

Table 5: Experimental results on the anomaly detection benchmark datasets in terms of precision, recall, and F1-score

| Methods                |       | SWaT  |       |       | WADI  |       |       | SMAP  |       |       | MSL   |       |       | HAI   |       |
|------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|                        | P(%)  | R(%)  | F1(%) | P(%)  | R(%)  | F1    | P(%)  | R(%)  | F1(%) | P(%)  | R(%)  | F1(%) | P(%)  | R(%)  | F1(%) |
| GAN-Li                 | 81.03 | 84.97 | 77.32 | 76.25 | 80.33 | 77.95 | 67.10 | 87.06 | 75.19 | 71.02 | 87.06 | 78.23 | 19.83 | 18.36 | 17.45 |
| LSTM-NDT               | 79.12 | 75.08 | 78.75 | 81.25 | 78.64 | 75.18 | 89.65 | 88.46 | 89.05 | 59.44 | 53.74 | 56.40 | 22.46 | 23.45 | 20.32 |
| MTAD-GAT               | 82.01 | 76.84 | 72.47 | 82.58 | 84.94 | 80.25 | 89.06 | 91.23 | 90.41 | 87.54 | 94.40 | 90.84 | 24.75 | 21.78 | 20.14 |
| MAD-GAN                | 98.97 | 63.74 | 77.0  | 41.44 | 33.92 | 37.0  | 80.49 | 82.14 | 81.31 | 85.17 | 89.91 | 87.47 | 25.27 | 23.34 | 21.87 |
| GDN                    | 99.35 | 68.12 | 81.0  | 97.50 | 40.19 | 57.0  | 86.62 | 84.27 | 83.24 | 89.92 | 87.24 | 86.84 | 43.41 | 46.27 | 44.59 |
| GTA                    | 74.91 | 96.41 | 84.0  | 74.56 | 90.50 | 82.0  | 89.11 | 91.76 | 90.41 | 91.04 | 91.17 | 91.11 | 44.91 | 41.63 | 40.29 |
| LOF                    | 72.15 | 65.43 | 68.62 | 57.02 | 61.17 | 53.46 | 58.93 | 56.33 | 57.60 | 47.72 | 85.25 | 61.18 | 31.27 | 29.93 | 26.48 |
| Deep-SVDD              | 80.42 | 84.45 | 82.39 | 74.18 | 70.82 | 73.43 | 89.93 | 56.02 | 69.04 | 91.92 | 76.63 | 83.58 | 34.81 | 31.26 | 30.94 |
| DAGMM                  | 89.92 | 57.84 | 70.4  | 54.44 | 26.99 | 36.0  | 86.45 | 56.73 | 68.51 | 89.60 | 63.93 | 74.62 | 35.56 | 37.12 | 33.77 |
| MMPCACD                | 82.52 | 68.29 | 74.73 | 74.29 | 75.01 | 71.48 | 88.61 | 75.84 | 81.73 | 81.42 | 61.31 | 69.95 | 31.58 | 29.46 | 27.33 |
| VAR                    | 81.59 | 60.29 | 69.34 | 75.59 | 69.36 | 66.21 | 81.38 | 53.88 | 64.83 | 74.68 | 81.42 | 77.90 | 34.42 | 36.28 | 31.97 |
| LSTM                   | 86.15 | 83.27 | 84.69 | 68.73 | 62.47 | 65.74 | 89.41 | 78.13 | 83.39 | 85.45 | 82.50 | 83.95 | 35.61 | 32.84 | 31.92 |
| CL-MPPCA               | 76.78 | 81.50 | 79.07 | 69.72 | 65.23 | 67.32 | 86.13 | 63.16 | 72.88 | 73.71 | 88.54 | 80.44 | 33.82 | 31.74 | 30.05 |
| ITAD                   | 63.13 | 52.08 | 57.08 | 71.95 | 69.39 | 65.76 | 82.42 | 66.89 | 73.85 | 69.44 | 84.09 | 76.07 | 36.72 | 33.42 | 32.47 |
| LSTM-VAE               | 76.00 | 89.50 | 82.20 | 87.79 | 14.45 | 25.0  | 92.20 | 67.75 | 78.10 | 85.49 | 79.94 | 82.62 | 38.25 | 37.94 | 35.04 |
| BeatGAN                | 64.01 | 87.46 | 73.92 | 74.46 | 70.71 | 76.52 | 92.38 | 55.85 | 69.61 | 89.75 | 85.42 | 87.53 | 39.41 | 38.03 | 35.47 |
| OmniAnomaly            | 81.42 | 84.30 | 82.83 | 78.18 | 80.13 | 77.24 | 92.49 | 81.99 | 86.92 | 89.02 | 86.37 | 87.67 | 46.29 | 43.75 | 42.73 |
| InterFusion            | 80.59 | 85.58 | 83.01 | 81.78 | 84.37 | 80.21 | 89.77 | 88.52 | 89.14 | 81.28 | 92.70 | 86.62 | 45.72 | 43.15 | 42.55 |
| THOC                   | 83.94 | 86.36 | 85.13 | 84.24 | 81.32 | 80.09 | 92.06 | 89.34 | 90.68 | 88.45 | 90.97 | 89.69 | 43.72 | 45.82 | 43.67 |
| GRELEN                 | 95.60 | 83.50 | 89.10 | 77.30 | 61.30 | 68.20 | 94.45 | 98.16 | 97.29 | 94.36 | 94.04 | 91.58 | 47.31 | 43.12 | 40.58 |
| Agentic-RAG W/Gemma-2B | 99.35 | 98.00 | 92.45 | 98.50 | 91.85 | 89.95 | 98.10 | 98.85 | 98.90 | 97.95 | 97.25 | 96.90 | 58.10 | 56.00 | 53.10 |
| Agentic-RAG W/Gemma-7B | 99.42 | 98.08 | 92.53 | 98.58 | 91.93 | 90.03 | 98.18 | 98.93 | 98.98 | 98.03 | 97.33 | 96.98 | 58.18 | 56.08 | 53.18 |
| Agentic-RAG W/Llama-8B | 99.47 | 98.15 | 92.59 | 98.63 | 91.97 | 90.08 | 98.24 | 98.97 | 99.04 | 98.11 | 97.37 | 97.04 | 58.27 | 56.13 | 53.24 |

Best performance in bold. Second-best with underlines(except Agentic-RAG framework Variants).

Table 6: Experimental results on simulated Tennessee Eastman dataset in terms of fault detection rate (FDR(%))

|                        |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       | •     | •     |       |
|------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Base Model             | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    | 16    | 17    | 18    | 19    | 20    |
| Transformer            | 99.64 | 98.45 | 5.00  | 99.96 | 28.86 | 100   | 100   | 96.43 | 5.19  | 17.48 | 77.51 | 98.20 | 94.01 | 99.97 | 5.39  | 13.43 | 91.53 | 93.76 | 25.13 | 48.05 |
| TCN                    | 99.61 | 97.93 | 5.12  | 100   | 26.46 | 100   | 100   | 94.68 | 5.19  | 35.57 | 80.51 | 96.63 | 93.48 | 99.97 | 5.36  | 21.10 | 96.14 | 93.90 | 23.39 | 47.92 |
| FNet                   | 99.67 | 98.64 | 4.86  | 99.18 | 25.82 | 100   | 100   | 96.76 | 18.87 | 18.87 | 76.08 | 98.11 | 94.07 | 99.96 | 5.48  | 13.74 | 91.05 | 93.70 | 24.43 | 45.59 |
| GTA                    | 98.12 | 99.35 | 5.88  | 98.04 | 55.82 | 100   | 100   | 97.34 | 20.18 | 34.33 | 79.81 | 98.72 | 96.03 | 98.21 | 7.64  | 16.69 | 92.25 | 94.78 | 26.57 | 47.31 |
| GDN                    | 99.81 | 99.27 | 6.72  | 99.56 | 41.07 | 100   | 100   | 95.04 | 16.46 | 41.22 | 79.57 | 99.64 | 95.71 | 97.58 | 7.83  | 15.64 | 92.79 | 95.27 | 27.17 | 48.81 |
| MTAD-GAT               | 99.78 | 98.91 | 8.92  | 99.81 | 39.33 | 100   | 100   | 98.57 | 20.37 | 43.93 | 82.47 | 99.51 | 96.84 | 99.74 | 10.13 | 16.98 | 94.47 | 94.60 | 30.79 | 58.90 |
| GRELEN                 | 99.67 | 98.64 | 10.86 | 99.18 | 51.82 | 100   | 100   | 96.76 | 18.87 | 48.87 | 76.08 | 98.11 | 94.07 | 99.96 | 5.48  | 13.74 | 91.05 | 93.70 | 24.43 | 62.59 |
| Agentic-RAG W/Gemma-2B | 99.60 | 99.75 | 16.10 | 99.85 | 75.20 | 99.85 | 99.85 | 99.30 | 28.90 | 68.00 | 87.00 | 99.30 | 98.50 | 99.60 | 13.80 | 29.20 | 99.70 | 98.05 | 41.10 | 79.20 |
| Agentic-RAG W/Gemma-7B | 99.66 | 99.82 | 16.18 | 99.90 | 75.28 | 99.90 | 99.90 | 99.40 | 29.00 | 68.12 | 87.10 | 99.35 | 98.58 | 99.68 | 13.88 | 29.30 | 99.78 | 98.13 | 41.18 | 79.28 |
| Agentic-RAG W/Llama-8B | 99.72 | 99.89 | 16.23 | 100   | 75.38 | 100   | 100   | 99.47 | 29.04 | 68.16 | 87.15 | 99.46 | 98.64 | 99.75 | 13.96 | 29.37 | 99.83 | 98.21 | 41.23 | 79.35 |

Best performance in bold. Second-best with underlines(except Agentic-RAG framework Variants).

16, an  $\alpha$  of 32, and a dropout of 0.05 to ensure efficient parameter updates. We performed preference tuning on the SLMs using Direct Preference Optimization(DPO[32]) along with QLoRA, minimizing the binary cross-entropy (BCE) loss with the following hyperparameters: a learning rate of 5.0e-7 with a cosine scheduler and a gradient accumulation of 2 steps.  $\beta$  was set to 0.2 to better align SLMs with the desired preferences. We conducted training for 3 epochs using the AdamW optimizer, with a batch size of 8 for both the training and evaluation phases. These hyperparameters were chosen to balance the trade-off between SLMs' performance on the specific time series task and computational resources. Optimal hyperparameter values are highly task-specific and depend on the dataset and language model architecture. Extensive experimentation are crucial to find the best configurations. We discuss the hyperparameter optimization results in appendix. To ensure efficient and consistent framework training, we preprocess timeseries data by standardizing each variable (zero mean, unit variance) and calculate evalution metric on the original scale. We leverage NVIDIA GPUs and PyTorch for accelerated training, enabling the use of small-scale models and datasets. For robust evaluation, we conduct multiple independent runs and report ensemble averages.

#### 5 RESULTS

Tables 3-4 present a performance comparison of the Agentic-RAG framework variants with baseline methods on seven benchmark

datasets (PeMSD3, PeMSD4, PeMSD7, PeMSD7M, PeMSD8, METR-LA, and PEMS-BAY) on the forecasting task. We report experimental results from a previous study [7] for a fair and rigorous comparison. Tables 5-6 show the performance of Agentic-RAG framework variants on time-series anomaly detection on benchmark datasets. We present experimental results of baseline methods from earlier studies [6, 11, 13, 43]. Our proposed framework outperforms baseline methods across the benchmark datasets, showing significant improvements on the forecasting and anomaly detection tasks. We present experimental results on missing data imputation and classification tasks in the appendix. Experimental results on univariate datasets across all time series tasks are discussed in the appendix.

#### 6 CONCLUSION

In this work, we propose an Agentic RAG framework to address the challenges of distribution shifts, and fixed-length subsequences in time series analysis. The framework overcomes these challenges by leveraging a hierarchical, multi-agent architecture with specialized sub-agents for various time series tasks. Each sub-agent utilizes a prompt pool as its internal knowledge base to store historical patterns and trends. The sub-agent retrieves relevant prompts and utilizes the corresponding knowledge to improve predictions on new, unseen data. This modular design with task-specific sub-agents and knowledge augmentation outperforms traditional methods in handling complex time series analysis tasks.

#### REFERENCES

- AI@Meta. 2024. Llama 3 Model Card. (2024). https://github.com/meta-llama/llama3/blob/main/MODEL\_CARD.md
- [2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. Advances in Neural Information Processing Systems 33 (2020), 1877–1901.
- [3] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. 2024. TEMPO: Prompt-based Generative Pre-trained Transformer for Time Series Forecasting. In The Twelfth International Conference on Learning Representations. https://openreview.net/forum?id=YH5w12OUuU
- [4] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. Brits: Bidirectional recurrent imputation for time series. Advances in neural information processing systems 31 (2018).
- [5] Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. 2001. Freeway performance measurement system: mining loop detector data. *Transportation Research Record* 1748, 1 (2001), 96–102.
- [6] Zekai Chen, Dingshuo Chen, Xiao Zhang, Zixuan Yuan, and Xiuzhen Cheng. 2021. Learning graph structures with transformer for multivariate time series anomaly detection in iot. IEEE Internet of Things Journal (2021).
- [7] Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. 2022. Graph neural controlled differential equations for traffic forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36. 6367–6374.
- [8] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. Advances in neural information processing systems 30 (2017).
- [9] Andrea Cini, Ivan Marisca, and Cesare Alippi. 2021. Multivariate Time Series Imputation by Graph Neural Networks. arXiv e-prints (2021), arXiv-2108.
- [10] Andrea Cini, Ivan Marisca, Daniele Zambon, and Cesare Alippi. 2024. Taming local effects in graph-based spatiotemporal forecasting. Advances in Neural Information Processing Systems 36 (2024).
- [11] Ailin Deng and Bryan Hooi. 2021. Graph neural network-based anomaly detection in multivariate time series. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. 4027–4035.
- [12] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. Advances in Neural Information Processing Systems 36 (2024).
- [13] Yiwei Fu and Feng Xue. 2022. MAD: Self-Supervised Masked Anomaly Detection Task for Multivariate Time Series. arXiv preprint arXiv:2205.02100 (2022).
- [14] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2024. Large language models are zero-shot time series forecasters. Advances in Neural Information Processing Systems 36 (2024).
- [15] Han Guo, Philip Greengard, Eric P Xing, and Yoon Kim. 2023. Lq-lora: Low-rank plus quantized matrix decomposition for efficient language model finetuning. arXiv preprint arXiv:2311.12023 (2023).
- [16] Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. arXiv preprint arXiv:2403.14608 (2024).
- [17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021).
- [18] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using 1stms and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 387–395.
- [19] Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. arXiv preprint arXiv:2401.01325 (2024).
- [20] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. 2023. Time-Ilm: Time series forecasting by reprogramming large language models. arXiv preprint arXiv:2310.01728 (2023).
- [21] Michael Leonard. 2001. Promotional analysis and forecasting for demand planning: a practical time series approach. with exhibits 1 (2001).
- [22] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In ICLR.
- [23] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. 2023. Ra-dit: Retrieval-augmented dual instruction tuning. arXiv preprint arXiv:2310.01352 (2023).
- [24] Hengbo Liu, Ziqing Ma, Linxiao Yang, Tian Zhou, Rui Xia, Yi Wang, Qingsong Wen, and Liang Sun. 2023. SADI: A Self-Adaptive Decomposed Interpretable Framework for Electric Load Forecasting Under Extreme Events. In IEEE International Conference on Acoustics, Speech and Signal Processing.
- [25] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017).

- [26] Ivan Marisca, Cesare Alippi, and Filippo Maria Bianchi. 2024. Graph-based Forecasting with Missing Data through Spatiotemporal Downsampling. arXiv preprint arXiv:2402.10634 (2024).
- [27] Ivan Marisca, Andrea Cini, and Cesare Alippi. 2022. Learning to reconstruct missing data from spatiotemporal graphs with sparse observations. Advances in Neural Information Processing Systems 35 (2022), 32069–32082.
- [28] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In The Eleventh International Conference on Learning Representations. https://openreview.net/forum?id=Jbdc0vTOcol
- [29] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]
- [30] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In International Conference on Learning Representations.
- [31] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. 2022. Fourcastnet: A global data-driven highresolution weather model using adaptive fourier neural operators. arXiv preprint arXiv:2202.11214 (2022).
- [32] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems 36 (2024).
- [33] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. Transactions of the Association for Computational Linguistics 11 (2023), 1316–1331.
- [34] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530 (2024).
- [35] Andreas Roth and Thomas Liebig. 2022. Forecasting Unobserved Node States with spatio-temporal Graph Neural Networks. arXiv preprint arXiv:2211.11596 (2022)
- [36] Lifeng Shen, Zhuocong Li, and James Kwok. 2020. Timeseries anomaly detection using temporal hierarchical one-class network. Advances in Neural Information Processing Systems 33 (2020), 13016–13026.
- [37] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. arXiv preprint arXiv:2301.12652 (2023).
- [38] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (RAG) models for open domain question answering. Transactions of the Association for Computational Linguistics 11 (2023), 1–17.
- [39] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023).
- [40] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. arXiv preprint arXiv:2403.08295 (2024).
- [41] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023).
- [42] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In The Eleventh International Conference on Learning Representations. https://openreview.net/forum?id=ju\_Uqw384Oq
- [43] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2021. Anomaly transformer: Time series anomaly detection with association discrepancy. arXiv preprint arXiv:2110.02642 (2021).
- [44] Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. arXiv preprint arXiv:2312.12148 (2023).
- [45] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629 (2022).
- [46] Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Long-bing Cao, and Zhendong Niu. 2024. FourierGNN: Rethinking multivariate time series forecasting from a pure graph perspective. Advances in Neural Information Processing Systems 36 (2024).
- [47] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. 2019. Time-series generative adversarial networks. Advances in neural information processing systems 32 (2019).

- [48] Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. 2022. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. arXiv preprint arXiv:2207.01186 (2022).
- [49] Weiqi Zhang, Chen Zhang, and Fugee Tsung. 2022. GRELEN: Multivariate Time Series Anomaly Detection from the Perspective of Graph Relational Learning.. In IJCAI. 2390–2397.
- [50] Yunhao Zhang and Junchi Yan. 2022. Crossformer: Transformer utilizing crossdimension dependency for multivariate time series forecasting. In The eleventh international conference on learning representations.
- [51] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2020. Multivariate timeseries anomaly detection via graph attention network. In 2020 IEEE International Conference on Data Mining (ICDM). IEEE, 841–850.
- [52] Helen Zhou, Sercan O Arik, and Jingtao Wang. 2023. Business Metric-Aware Forecasting for Inventory Management. arXiv preprint arXiv:2308.13118 (2023).
- [53] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35. 11106–11115.
- [54] Qihang Zhou, Shibo He, Haoyu Liu, Jiming Chen, and Wenchao Meng. 2024. Label-free multivariate time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering* (2024).
- [55] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Proc. 39th International Conference on Machine Learning (ICML 2022) (Baltimore, Maryland).
- [56] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2024. One fits all: Power general time series analysis by pretrained lm. Advances in neural information processing systems 36 (2024).
- [57] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. In Thirty-seventh Conference on Neural Information Processing Systems. https://openreview.net/ forum?id=gMS6FVZvmF

# A MULTIVARIATE SPATIO-TEMPORAL DATASETS

# A.1 Missing Data Imputation

Time series imputation is a critical step in time series analysis. It addresses a common issue in this field: missing values within datasets. These missing values can arise from sensor failures, data transmission errors, or incomplete records. By imputing these gaps, time series imputation ensures the quality and reliability of subsequent analyses. The Agentic-RAG framework achieves this by handling seasonality, trends and capturing the inherent spatio-temporal dependencies within the data. Ultimately, imputation improves data quality, enabling more accurate analysis, modeling, and decisionmaking. In essence, it plays a vital role by maintaining data integrity and enabling reliable analysis. To evaluate the Agentic-RAG framework's ability to handle missing data, we simulated two types of missingness patterns: point missing and block missing[9, 35]. These patterns represent varying degrees of data availability. To achieve this, we introduced synthetic missingness into time series datasets following these patterns. For point missing, individual values were randomly omitted with a probability threshold (p), controlling the overall percentage of missing data. The block missing pattern involves removing contiguous, multi-period, multi-time series segments. This is done by randomly selecting start and end times, as well as start and end time series, to define uniform blocks with an average length of (1). All data points within each block are then omitted. Furthermore, two block missing patterns are considered: temporal and spatial. For temporal block missing, contiguous multi-period segments are removed from a given time series. This is done by randomly selecting start and end times, creating stretches of unavailable temporal data. For spatial block missing, contiguous blocks are removed across multiple related time series at specific time points. This involves randomly selecting the start and end time series, resulting in missing spatial data at the chosen time points. Both patterns show varying levels of missing information in the time series data. In summary, point missing refers to sporadic gaps in the data, while block missing involves the absence of entire contiguous multi-period and multi-series segments. Block missing can further be categorized into two types: temporal block missing, where contiguous segments are removed within a single time series, and spatial block missing, where contiguous blocks are removed across multiple related time series, mimicking realistic scenarios of faulty data collection. In the context of time series imputation, "in-sample" and "out-of-sample" imputation refer to distinct evaluation settings. In-sample imputation involves the imputation method reconstructing missing values within a given fixed input sequence,  $S^t$ , using all available observed data within that sequence. Out-of-sample imputation involves training the imputation method using the fixed sequence  $S^t$  to impute missing points in a future sequence,  $S^{t+1}$ . In this work, we utilize out-of-sample settings, as this approach mimics real-world scenarios and rigorously assesses the Agentic-RAG framework's robustness and generalizability by evaluating its ability to handle new, unseen data. The simulated datasets with missing values were then used to evaluate the missing data handling capabilities of the proposed Agentic-RAG framework. We split multiple benchmark datasets in chronological order with

a ratio of 7:1:2 for the METR-LA and PEMS-BAY datasets and a ratio of 6:2:2 for the other datasets into training, validation, and test sets. We evaluated the Agentic-RAG framework's performance on simulated data using multiple imputation metrics (e.g., RMSE, MAE, and MAPE). This analysis helps us understand how well the framework handles time series data with missing values, particularly how its performance changes as the percentage of missing data increases. We establish the Agentic-RAG framework, trained on complete data (no missing values), as a strong performance benchmark. This benchmark allows us to evaluate the framework's effectiveness in imputing missing data under different conditions of data incompleteness. Tables 7 and 8 present the imputation results on standard benchmark datasets with different missingness patterns, while the framework performs slightly worse than the baseline for minimal missing data. Its accuracy degrades more significantly as the data becomes more incomplete, regardless of the specific missingness pattern. Our proposed Agentic-RAG framework demonstrates robustness to missing data by focusing on the available observations for imputing missing values, thereby avoiding the introduction of potentially inaccurate estimates that could obscure the underlying trends and patterns within the time series data. Additionally, the Agentic-RAG framework effectively captures the complex non-linear intra- and inter-time series dependencies and this leads to more reliable imputation. The experiments show that our framework can learn the spatiotemporal dependencies from partially observed data with various missingness patterns, resulting in lower imputation errors.

#### A.2 Time Series Classification

Time series classification is a crucial task with applications across various domains. In time series analysis, regimes, or clusters represent distinct behavioral modes, operating conditions, or states of the system underlying the data. Identifying and characterizing these regimes is crucial for understanding the complex patterns and dynamics within the data. This allows for more accurate modeling, forecasting, and decision-making in applications where time series analysis is essential. The emergence of different regimes or clusters can stem from changes in the data generation process, external conditions, or the inherent non-stationarity and multivariate nature of the time series. This reflects the rich information content and complexity often encountered in real-world time series data. To evaluate the proposed Agentic-RAG framework's ability to handle time series classification tasks, an unsupervised clustering approach was employed for data labeling. We first applied k-means clustering to the original time series datasets, determining the optimal number of clusters (k) using established techniques such as the elbow method or silhouette analysis. The optimal clusters were treated as class labels, representing distinct regimes within the time series, and each time series was assigned the corresponding cluster label, creating a labeled classification dataset. We adopted a time-based division strategy to split multiple benchmark datasets into training, validation, and testing sets. The METR-LA and PEMS-BAY datasets were split at a 7:1:2 ratio, while other datasets used a 6:2:2 split. We evaluated the framework's performance on the held-out test set using standard classification metrics: accuracy, precision, recall. This methodology allowed us to assess the framework's ability to learn the underlying patterns and relationships associated with

| Missing Scheme                    | Missing Rate  | ]     | PeMSD | 3     | ]     | PeMSD | 4     | 1     | PeMSD | 7     | M     | ETR-I | ĹA    |
|-----------------------------------|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| wissing scheme                    | Wilssing Kate |       | MAE   | MAPE  | RMSE  | MAE   | MAPE  | RMSE  | MAE   | MAPE  | RMSE  | MAE   | MAPE  |
| SelfExtend-Agentic-RAG w/Llama-8B | 0%            | 19.48 | 13.01 | 10.53 | 25.54 | 17.46 | 9.52  | 29.97 | 19.02 | 8.03  | 6.23  | 3.12  | 8.53  |
|                                   | 10%           | 21.12 | 14.07 | 12.15 | 28.23 | 19.18 | 11.04 | 32.11 | 20.06 | 10.12 | 7.05  | 4.01  | 10.13 |
| Point                             | 30%           | 22.55 | 15.23 | 13.32 | 30.61 | 20.62 | 12.63 | 34.62 | 21.58 | 11.64 | 7.82  | 4.51  | 11.02 |
|                                   | 50%           | 24.14 | 16.39 | 14.29 | 33.17 | 22.21 | 14.08 | 37.24 | 23.15 | 13.21 | 8.57  | 5.03  | 12.18 |
|                                   | 10%           | 25.07 | 17.14 | 15.25 | 35.18 | 23.14 | 15.18 | 39.21 | 25.19 | 14.13 | 9.04  | 5.53  | 13.12 |
| Block                             | 30%           | 27.21 | 18.45 | 16.48 | 38.28 | 25.12 | 17.23 | 42.32 | 27.07 | 16.27 | 10.09 | 6.02  | 14.57 |
|                                   | 50%           | 29.18 | 20.09 | 18.19 | 41.23 | 27.11 | 19.16 | 45.27 | 29.03 | 18.12 | 11.11 | 6.53  | 16.07 |
| Block                             | 10%           | 23.04 | 15.59 | 13.42 | 31.19 | 21.23 | 13.09 | 35.18 | 22.14 | 12.61 | 8.02  | 4.53  | 11.59 |
| (Only Spatial)                    | 30%           | 25.09 | 17.23 | 15.18 | 34.26 | 23.15 | 15.12 | 38.25 | 24.19 | 14.21 | 9.11  | 5.02  | 13.13 |
| (Only Spatial)                    | 50%           | 27.15 | 18.52 | 16.59 | 37.23 | 25.18 | 17.19 | 41.16 | 26.13 | 16.17 | 10.14 | 5.57  | 14.52 |
| Block                             | 10%           | 22.57 | 15.12 | 13.18 | 30.62 | 20.53 | 13.07 | 34.53 | 21.48 | 11.64 | 7.81  | 4.52  | 11.19 |
| (Only Temporal)                   | 30%           | 24.62 | 16.48 | 14.53 | 33.72 | 22.48 | 15.27 | 37.58 | 23.41 | 13.58 | 8.89  | 5.08  | 12.59 |
| (Omy Temporar)                    | 50%           | 26.48 | 18.19 | 16.32 | 36.53 | 24.31 | 18.02 | 40.42 | 25.38 | 15.43 | 9.76  | 5.53  | 14.07 |

Table 7: The table presents the Agentic-RAG framework's evaluation results on various metrics for missing data imputation across PeMSD3, PeMSD4, PeMSD7, and METR-LA benchmark datasets with diverse missing data patterns.

| Missing Scheme                    | Missing Rate  | P    | eMSD7( | M)   |       | PeMSD | 8    | I    | PEMS-BA | 4Y   |
|-----------------------------------|---------------|------|--------|------|-------|-------|------|------|---------|------|
| wiissing Scheme                   | Wiissing Kate | MAE  | RMSE   | MAPE | MAE   | RMSE  | MAPE | MAE  | RMSE    | MAPE |
| SelfExtend-Agentic-RAG w/Llama-8B | 0%            | 2.33 | 4.68   | 5.88 | 14.03 | 20.98 | 7.04 | 1.62 | 3.12    | 3.14 |
|                                   | 10%           | 2.46 | 4.75   | 6.12 | 15.14 | 22.12 | 7.58 | 1.72 | 3.26    | 3.28 |
| Point                             | 30%           | 2.68 | 5.02   | 6.43 | 16.27 | 23.18 | 8.12 | 1.83 | 3.41    | 3.42 |
|                                   | 50%           | 2.89 | 5.27   | 6.73 | 17.32 | 24.29 | 8.69 | 1.94 | 3.56    | 3.57 |
|                                   | 10%           | 2.61 | 4.89   | 6.37 | 15.75 | 22.98 | 7.89 | 1.79 | 3.34    | 3.34 |
| Block                             | 30%           | 2.84 | 5.21   | 6.68 | 16.92 | 23.99 | 8.42 | 1.89 | 3.48    | 3.48 |
|                                   | 50%           | 3.07 | 5.53   | 7.03 | 18.12 | 25.08 | 8.98 | 2.01 | 3.63    | 3.63 |
| Block                             | 10%           | 2.55 | 4.81   | 6.23 | 15.49 | 22.68 | 7.75 | 1.75 | 3.31    | 3.31 |
| (Spatial Only)                    | 30%           | 2.78 | 5.12   | 6.56 | 16.67 | 23.74 | 8.28 | 1.86 | 3.46    | 3.46 |
| (Spatial Offly)                   | 50%           | 3.00 | 5.41   | 6.88 | 17.89 | 24.89 | 8.83 | 1.97 | 3.60    | 3.60 |
| Block                             | 10%           | 2.52 | 4.78   | 6.18 | 15.37 | 22.58 | 7.72 | 1.74 | 3.29    | 3.29 |
| (Temporal Only)                   | 30%           | 2.75 | 5.09   | 6.51 | 16.52 | 23.62 | 8.24 | 1.85 | 3.44    | 3.44 |
| (Temporal Only)                   | 50%           | 2.98 | 5.38   | 6.83 | 17.75 | 24.76 | 8.80 | 1.96 | 3.58    | 3.58 |

Table 8: The table presents the performance of the Agentic-RAG framework in imputing missing data on the PeMSD7(M), PeMSD8, and PEMS-BAY benchmark datasets with the various synthetic missing data patterns.

each cluster/class and its overall effectiveness in classifying time series data based on inherent complex spatio-temporal regimes, paving the way for its practical application in real-world scenarios. The experimental results, presented in Tables 9 and 10, show a comparison with the simple baselines.

#### **B** UNIVARIATE DATASETS

We conducted several experiments to evaluate the proposed Agentic-RAG framework variants: SelfExtend-Agentic-RAG with Gemma-2B, SelfExtend-Agentic-RAG with Gemma-7B, and SelfExtend-Agentic-RAG with Llama-8B, on the univariate datasets for multiple time series analysis tasks such as forecasting and imputation.

# **B.1** Forecasting and Imputation

The ETT (Electricity Transformer) datasets[53], ETTh1, ETTh2, ETTm1, and ETTm2, are popular benchmarks used for evaluating and benchmarking univariate time series forecasting methods. They provide a challenging benchmark due to the presence of complex patterns, such as trends, seasonality, and irregularities, which are commonly found in real-world time series data. ETTh1 and

ETTh2 are two hourly time series datasets containing observations of electricity transformers from two different locations. ETTm1 and ETTm2 are two monthly time series datasets containing observations of electricity transformers from two different locations. In this work, we utilize the ETT datasets[53] to evaluate the Agentic-RAG framework for both forecasting and missing data imputation tasks. The Table 11 shows the performance of various methods on the multi-horizon forecasting task using a lookback window of size 512. It presents mean squared error (MSE) and mean absolute error (MAE) for nine models (GPT4TS[57], PatchTST[28], TimesNet[42], FEDFormer[55], LightTS[48], N-BEATS[30], Agentic-RAG w/Gemma-2B, Agentic-RAG w/Gemma-7B, and Agentic-RAG w/Llama-8B) across four datasets (ETTh1, ETTh2, ETTm1, ETTm2) at different time horizons (96, 192, 336, 720). This allows for a comprehensive analysis of forecasting accuracy and robustness of Agentic-RAG framework across varying prediction lengths. The performance of various methods for imputing missing data (point and block missing) and their effectiveness in out-of-sample imputation settings are compared in Tables 12 and 13. The evaluated methods

| Dataset                           |          | PeMSD3    |        |          | PeMSD4    |        |          | PeMSD7    |        | N        | METR-LA   |        |
|-----------------------------------|----------|-----------|--------|----------|-----------|--------|----------|-----------|--------|----------|-----------|--------|
| Dataset                           | Accuracy | Precision | Recall |
| SelfExtend-Agentic-RAG W/Gemma-2B | 91.23%   | 89.54%    | 90.87% | 92.51%   | 91.34%    | 92.08% | 93.04%   | 92.21%    | 92.83% | 94.15%   | 93.51%    | 93.81% |
| SelfExtend-Agentic-RAG W/Gemma-7B | 92.12%   | 90.79%    | 91.53% | 93.23%   | 92.04%    | 92.72% | 94.01%   | 93.01%    | 93.52% | 95.05%   | 94.33%    | 94.58% |
| SelfExtend-Agentic-RAG W/Llama-8B | 93.01%   | 91.56%    | 92.31% | 94.02%   | 92.82%    | 93.56% | 95.03%   | 94.02%    | 94.21% | 95.82%   | 95.02%    | 95.24% |
| LSTM                              | 85.01%   | 83.24%    | 84.05% | 86.56%   | 85.02%    | 85.57% | 87.04%   | 86.01%    | 86.54% | 88.01%   | 87.53%    | 87.81% |
| MLP                               | 82.01%   | 80.54%    | 81.02% | 83.01%   | 81.84%    | 82.02% | 84.51%   | 83.52%    | 84.01% | 85.03%   | 84.21%    | 84.52% |

Table 9: The table shows the evaluation results of the Agentic-RAG framework variants performance on various metrics for time series classification on the PeMSD3, PeMSD4, PeMSD7, and METR-LA benchmark datasets.

| Dataset                           | P        | eMSD7(M)  |        |          | PeMSD8    |        | P        | EMS-BAY   |        |
|-----------------------------------|----------|-----------|--------|----------|-----------|--------|----------|-----------|--------|
| Dataset                           | Accuracy | Precision | Recall | Accuracy | Precision | Recall | Accuracy | Precision | Recall |
| SelfExtend-Agentic-RAG W/Gemma-2B | 92.03%   | 90.52%    | 91.21% | 93.54%   | 92.35%    | 92.84% | 94.01%   | 93.02%    | 93.51% |
| SelfExtend-Agentic-RAG W/Gemma-7B | 93.02%   | 91.51%    | 92.03% | 94.03%   | 93.01%    | 93.53% | 95.01%   | 94.01%    | 94.53% |
| SelfExtend-Agentic-RAG W/Llama-8B | 94.02%   | 92.54%    | 93.02% | 95.04%   | 94.03%    | 94.52% | 96.01%   | 95.01%    | 95.53% |
| LSTM                              | 85.54%   | 84.01%    | 84.52% | 87.01%   | 85.52%    | 86.01% | 88.02%   | 87.01%    | 87.54% |
| MLP                               | 83.01%   | 81.52%    | 82.02% | 84.52%   | 83.01%    | 83.51% | 86.01%   | 85.01%    | 85.53% |

Table 10: The table presents a comparative evaluation of the Agentic-RAG framework variants performance on three benchmark datasets: PeMSD7(M), PeMSD8, and PEMS-BAY, across various metrics for time series classification.

include GPT4TS[57], PatchTST[28], TimesNet[42], FEDFormer[55], LightTS[48], N-BEATS[30], Agentic-RAG with Gemma-2B, Agentic-RAG with Gemma-7B, and Agentic-RAG with Llama-8B. The evaluation employs a 512-step historical window for imputing 96-step-ahead (short-term prediction) and 720-step-ahead (long-term prediction) missing values in future data. The tables show results for four datasets (ETTh1, ETTh2, ETTm1, ETTm2) under three missing data scenarios: 0% missing (no missing data), 20% point missing, and 20% block missing. The proposed Agentic-RAG framework variants demonstrate strong performance on the benchmark datasets for both forecasting and imputation tasks, with lower errors.

#### C ENVIRONMENTAL IMPACT

Our Agentic-RAG framework training process, involving multiple variants running for extended periods, increases our energy consumption and carbon footprint. Accurate quantification of the carbon footprint of deep learning experiments is essential for promoting sustainable practices in artificial intelligence research and development. A crucial aspect of this endeavor is estimating the energy consumption and associated greenhouse gas emissions during the computationally intensive training processes. This is calculated by determining the Total Graphics Power (TGP), which represents the maximum power draw of the GPU, including the GPU chip itself and other components like memory and additional circuitry. For example, the NVIDIA P100 GPU has a TGP of 300 watts, while the NVIDIA T4 GPU has a TGP of 70 watts. By multiplying the TGP by the training time, we can estimate the energy consumption, which is then converted to carbon emissions using a region-specific carbon intensity factor. This factor accounts for the energy mix (coal, natural gas, renewables, etc.) used to generate electricity in the geographic area where the computations are performed. Considering a 725-GPU hours training experiment and using an estimated carbon intensity factor of 0.0007 metric tons CO2e per kWh for the year 2024 (for more information on the carbon intensity of electricity, you can visit CO2 Intensity - Our World in Data), the calculated carbon footprint would be 152.25 kg CO2e for the NVIDIA P100

GPU and 35.525 kg CO2e for the NVIDIA T4 GPU. Note: kg CO2e stands for kilograms of carbon dioxide equivalent. The average person in the United States emits approximately 43.8 kg of carbon dioxide equivalent (CO2e) per day. Given the emissions of 152.25 kg CO2e for the NVIDIA P100 GPU and 35.525 kg CO2e for the NVIDIA T4 GPU, it would take a single person's emissions approximately 3.5 days to match the emissions of the P100 GPU and approximately 0.8 days (or 19 hours) to match the emissions of the T4 GPU. While the calculated carbon footprint provides valuable insight, the actual energy consumption and resulting emissions may vary due to factors like GPU utilization and regional energy sources. Nonetheless, quantifying the carbon footprint is a crucial step towards understanding and mitigating the environmental impact of deep learning research, paving the way for more sustainable and responsible practices in artificial intelligence.

#### **D** HYPERPARAMETER OPTIMIZATION

Hyperparameter optimization involves training the Agentic-RAG framework variants multiple times with different hyperparameter settings. This can be computationally expensive, especially for complex pre-trained language models or large datasets. We optimized the hyperparameters for the best-performing Agentic-RAG w/Llama-8B variant. For simplicity and in the interest of time, we have utilized the same settings for evaluating the performance of Agentic-RAG with w/Gemma-2B and w/Gemma-7B variants for both multivariate and univariate datasets across all tasks. In our experiments, we optimized the training process for supervised fine-tuning using a batch size from  $\{16, 32, 64\}$ , learning rate from  $\{1e-5, 5e-5, 1e-4\}$ . The training was conducted over epochs in the range of {10, 15, 20} with a warmup step count from {500, 1000, 1500} and a weight decay for regularization from {0.01, 0.05, 0.1}. We used gradient accumulation steps for stabilized training convergence from {2, 4, 8} and employed the AdamW optimizer. To manage memory and computational efficiency, we applied 4-bit quantization for QLoRA, with hyperparameters including a low-rank ( $\dot{r}$ ) from {16, 32, 64}, an (' $\alpha$ ') from {32, 64, 128}, and a dropout from {0.05, 0.1, 0.2}. For

| Metho    | ds  | GPT   | T4TS  | Patcl | hTST  | Time  | esNet | FEDF  | ormer | Ligh  | ntTS  | N-Bl  | EATS  | ARAG  | w/-2B | ARAG  | w/-7B | ARAC  | G-w/8B |
|----------|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|
| Metri    | ic  | MSE   | MAE    |
|          | 96  | 0.376 | 0.397 | 0.370 | 0.399 | 0.384 | 0.402 | 0.376 | 0.419 | 0.424 | 0.432 | 0.399 | 0.428 | 0.410 | 0.435 | 0.407 | 0.433 | 0.369 | 0.396  |
| ETTh1    | 192 | 0.416 | 0.418 | 0.413 | 0.421 | 0.436 | 0.429 | 0.420 | 0.448 | 0.475 | 0.462 | 0.451 | 0.464 | 0.448 | 0.461 | 0.445 | 0.459 | 0.412 | 0.417  |
| EIIII    | 336 | 0.442 | 0.433 | 0.422 | 0.436 | 0.491 | 0.469 | 0.459 | 0.465 | 0.518 | 0.488 | 0.498 | 0.500 | 0.487 | 0.476 | 0.484 | 0.473 | 0.421 | 0.434  |
|          | 720 | 0.477 | 0.456 | 0.447 | 0.466 | 0.521 | 0.500 | 0.506 | 0.507 | 0.547 | 0.533 | 0.608 | 0.573 | 0.496 | 0.482 | 0.491 | 0.478 | 0.446 | 0.464  |
|          | 96  | 0.285 | 0.342 | 0.274 | 0.336 | 0.340 | 0.374 | 0.358 | 0.397 | 0.397 | 0.437 | 0.327 | 0.387 | 0.345 | 0.378 | 0.342 | 0.374 | 0.273 | 0.335  |
| ETTh2    | 192 | 0.354 | 0.389 | 0.339 | 0.379 | 0.402 | 0.414 | 0.429 | 0.439 | 0.520 | 0.504 | 0.400 | 0.435 | 0.387 | 0.410 | 0.384 | 0.406 | 0.338 | 0.378  |
| E1 1112  | 336 | 0.373 | 0.407 | 0.329 | 0.380 | 0.452 | 0.452 | 0.496 | 0.487 | 0.626 | 0.559 | 0.747 | 0.599 | 0.465 | 0.468 | 0.462 | 0.465 | 0.328 | 0.379  |
|          | 720 | 0.406 | 0.441 | 0.379 | 0.422 | 0.462 | 0.468 | 0.463 | 0.474 | 0.863 | 0.672 | 1.454 | 0.847 | 0.473 | 0.472 | 0.469 | 0.469 | 0.371 | 0.420  |
|          | 96  | 0.292 | 0.346 | 0.290 | 0.342 | 0.338 | 0.375 | 0.379 | 0.419 | 0.374 | 0.400 | 0.318 | 0.367 | 0.354 | 0.369 | 0.351 | 0.366 | 0.289 | 0.340  |
| ETTm1    | 192 | 0.332 | 0.372 | 0.332 | 0.369 | 0.374 | 0.387 | 0.426 | 0.441 | 0.400 | 0.407 | 0.355 | 0.391 | 0.368 | 0.383 | 0.365 | 0.380 | 0.331 | 0.367  |
| EIIIII   | 336 | 0.366 | 0.394 | 0.366 | 0.392 | 0.410 | 0.411 | 0.445 | 0.459 | 0.438 | 0.438 | 0.401 | 0.419 | 0.396 | 0.404 | 0.392 | 0.400 | 0.365 | 0.388  |
|          | 720 | 0.417 | 0.421 | 0.416 | 0.420 | 0.478 | 0.450 | 0.543 | 0.490 | 0.527 | 0.502 | 0.448 | 0.448 | 0.435 | 0.427 | 0.431 | 0.423 | 0.411 | 0.419  |
|          | 96  | 0.173 | 0.262 | 0.165 | 0.255 | 0.187 | 0.267 | 0.203 | 0.287 | 0.209 | 0.308 | 0.197 | 0.271 | 0.190 | 0.265 | 0.187 | 0.262 | 0.164 | 0.254  |
| ETTm2    | 192 | 0.229 | 0.301 | 0.220 | 0.292 | 0.249 | 0.309 | 0.269 | 0.328 | 0.311 | 0.382 | 0.285 | 0.328 | 0.276 | 0.318 | 0.273 | 0.315 | 0.219 | 0.290  |
| E1 IIIIZ | 336 | 0.286 | 0.341 | 0.274 | 0.329 | 0.321 | 0.351 | 0.325 | 0.366 | 0.442 | 0.466 | 0.338 | 0.366 | 0.319 | 0.354 | 0.316 | 0.351 | 0.273 | 0.328  |
|          | 720 | 0.378 | 0.401 | 0.362 | 0.385 | 0.408 | 0.403 | 0.421 | 0.415 | 0.675 | 0.587 | 0.395 | 0.419 | 0.410 | 0.411 | 0.407 | 0.408 | 0.361 | 0.384  |

Table 11: The table compares various methods for the multi-horizon forecasting task with a lookback window of size 512.

| Met   | hods   | GPT   | T4TS  | Patcl | hTST  | Time  | esNet | FEDF  | ormer | Ligl  | ntTS  | N-BI  | EATS  | ARAC  | 6 w/-2B | ARAG  | w/-7B | ARAG  | G-w/8B |
|-------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|---------|-------|-------|-------|--------|
| Me    | etric  | MSE   | MAE     | MSE   | MAE   | MSE   | MAE    |
|       | 0%     | 0.376 | 0.397 | 0.370 | 0.399 | 0.384 | 0.402 | 0.376 | 0.419 | 0.424 | 0.432 | 0.399 | 0.428 | 0.410 | 0.435   | 0.407 | 0.433 | 0.369 | 0.396  |
| ETTh1 | 20% PM | 0.460 | 0.480 | 0.450 | 0.475 | 0.460 | 0.490 | 0.455 | 0.485 | 0.470 | 0.500 | 0.465 | 0.495 | 0.468 | 0.498   | 0.465 | 0.495 | 0.450 | 0.475  |
|       | 20% BM | 0.550 | 0.570 | 0.545 | 0.565 | 0.550 | 0.580 | 0.548 | 0.575 | 0.560 | 0.590 | 0.555 | 0.585 | 0.558 | 0.588   | 0.555 | 0.585 | 0.545 | 0.565  |
|       | 0%     | 0.285 | 0.342 | 0.274 | 0.336 | 0.340 | 0.374 | 0.358 | 0.397 | 0.397 | 0.437 | 0.327 | 0.387 | 0.345 | 0.378   | 0.342 | 0.374 | 0.273 | 0.335  |
| ETTh2 | 20% PM | 0.370 | 0.420 | 0.360 | 0.415 | 0.380 | 0.440 | 0.375 | 0.435 | 0.390 | 0.450 | 0.380 | 0.440 | 0.383 | 0.443   | 0.380 | 0.440 | 0.360 | 0.415  |
|       | 20% BM | 0.460 | 0.510 | 0.450 | 0.505 | 0.470 | 0.530 | 0.465 | 0.525 | 0.480 | 0.540 | 0.470 | 0.530 | 0.473 | 0.533   | 0.470 | 0.530 | 0.450 | 0.505  |
|       | 0%     | 0.292 | 0.346 | 0.290 | 0.342 | 0.338 | 0.375 | 0.379 | 0.419 | 0.374 | 0.400 | 0.318 | 0.367 | 0.354 | 0.369   | 0.351 | 0.366 | 0.289 | 0.340  |
| ETTm1 | 20% PM | 0.380 | 0.430 | 0.375 | 0.425 | 0.390 | 0.450 | 0.385 | 0.445 | 0.400 | 0.460 | 0.395 | 0.455 | 0.398 | 0.458   | 0.395 | 0.455 | 0.375 | 0.425  |
|       | 20% BM | 0.470 | 0.520 | 0.465 | 0.515 | 0.480 | 0.540 | 0.475 | 0.535 | 0.490 | 0.550 | 0.485 | 0.545 | 0.488 | 0.548   | 0.485 | 0.545 | 0.465 | 0.515  |
|       | 0%     | 0.173 | 0.262 | 0.165 | 0.255 | 0.187 | 0.267 | 0.203 | 0.287 | 0.209 | 0.308 | 0.197 | 0.271 | 0.190 | 0.265   | 0.187 | 0.262 | 0.164 | 0.254  |
| ETTm2 | 20% PM | 0.250 | 0.330 | 0.245 | 0.325 | 0.260 | 0.345 | 0.255 | 0.340 | 0.270 | 0.355 | 0.265 | 0.350 | 0.268 | 0.353   | 0.265 | 0.350 | 0.245 | 0.325  |
|       | 20% BM | 0.340 | 0.420 | 0.335 | 0.415 | 0.350 | 0.435 | 0.345 | 0.430 | 0.360 | 0.445 | 0.355 | 0.440 | 0.358 | 0.443   | 0.355 | 0.440 | 0.335 | 0.415  |

Table 12: The table compares different methods for imputing missing data, specifically for point missing (PM) and block missing (BM) scenarios, using a 512-step lookback window for forecasting 96 steps ahead.

| Met      | Methods |       | PT4TS PatchTST |       | TimesNet FE |       | FEDF  | FEDFormer LightTS |       | ntTS  | N-BEATS |       | ARAG w/-2B |       | ARAG w/-7B |       | ARAG-w/8B |       |       |
|----------|---------|-------|----------------|-------|-------------|-------|-------|-------------------|-------|-------|---------|-------|------------|-------|------------|-------|-----------|-------|-------|
| Me       | etric   | MSE   | MAE            | MSE   | MAE         | MSE   | MAE   | MSE               | MAE   | MSE   | MAE     | MSE   | MAE        | MSE   | MAE        | MSE   | MAE       | MSE   | MAE   |
| ETTh1    | 0%      | 0.477 | 0.456          | 0.447 | 0.466       | 0.521 | 0.500 | 0.506             | 0.507 | 0.547 | 0.533   | 0.608 | 0.573      | 0.496 | 0.482      | 0.491 | 0.478     | 0.446 | 0.464 |
| LIIII    | 20% PM  | 0.580 | 0.560          | 0.550 | 0.570       | 0.620 | 0.600 | 0.605             | 0.605 | 0.645 | 0.630   | 0.710 | 0.670      | 0.595 | 0.580      | 0.590 | 0.575     | 0.550 | 0.570 |
|          | 20% BM  | 0.690 | 0.670          | 0.660 | 0.680       | 0.740 | 0.720 | 0.725             | 0.725 | 0.765 | 0.750   | 0.830 | 0.790      | 0.715 | 0.700      | 0.710 | 0.695     | 0.670 | 0.680 |
| ETTh2    | 0%      | 0.406 | 0.441          | 0.379 | 0.422       | 0.462 | 0.468 | 0.463             | 0.474 | 0.863 | 0.672   | 1.454 | 0.847      | 0.473 | 0.472      | 0.469 | 0.469     | 0.371 | 0.420 |
| EIIIIZ   | 20% PM  | 0.510 | 0.545          | 0.483 | 0.526       | 0.566 | 0.572 | 0.567             | 0.578 | 0.967 | 0.776   | 1.558 | 0.947      | 0.577 | 0.576      | 0.573 | 0.573     | 0.475 | 0.524 |
|          | 20% BM  | 0.620 | 0.655          | 0.593 | 0.636       | 0.676 | 0.682 | 0.677             | 0.688 | 1.067 | 0.876   | 1.658 | 1.047      | 0.677 | 0.676      | 0.673 | 0.673     | 0.575 | 0.624 |
| ETTm1    | 0%      | 0.417 | 0.421          | 0.416 | 0.420       | 0.478 | 0.450 | 0.543             | 0.490 | 0.527 | 0.502   | 0.448 | 0.448      | 0.435 | 0.427      | 0.431 | 0.423     | 0.411 | 0.419 |
| ETIMI    | 20% PM  | 0.520 | 0.525          | 0.519 | 0.523       | 0.581 | 0.553 | 0.646             | 0.593 | 0.630 | 0.602   | 0.551 | 0.551      | 0.538 | 0.530      | 0.534 | 0.526     | 0.514 | 0.522 |
|          | 20% BM  | 0.630 | 0.635          | 0.629 | 0.633       | 0.691 | 0.663 | 0.756             | 0.703 | 0.740 | 0.712   | 0.661 | 0.661      | 0.648 | 0.640      | 0.644 | 0.636     | 0.624 | 0.632 |
| ETTm2    | 0%      | 0.378 | 0.401          | 0.362 | 0.385       | 0.408 | 0.403 | 0.421             | 0.415 | 0.675 | 0.587   | 0.395 | 0.419      | 0.410 | 0.411      | 0.407 | 0.408     | 0.361 | 0.384 |
| E1 IIII2 | 20% PM  | 0.480 | 0.503          | 0.464 | 0.487       | 0.510 | 0.505 | 0.523             | 0.517 | 0.777 | 0.689   | 0.495 | 0.519      | 0.510 | 0.511      | 0.507 | 0.508     | 0.461 | 0.484 |
|          | 20% BM  | 0.590 | 0.613          | 0.574 | 0.597       | 0.620 | 0.615 | 0.633             | 0.627 | 0.877 | 0.789   | 0.595 | 0.619      | 0.610 | 0.611      | 0.607 | 0.608     | 0.561 | 0.584 |

Table 13: The table evaluates the effectiveness of various missing data imputation techniques (including point-wise and block-wise methods) for out-of-sample imputation, using a 512-step historical window to predict missing values in subsequent 720-step future data.

preference tuning, the hyperparameter (' $\beta$ ') was set in the range of  $\{0.2, 0.4, 0.6\}$  and learning rate from  $\{5.0e-7, 1.0e-6, 5.0e-6\}$ . The optimal hyperparameters for training were chosen to achieve a balance between performance and computational efficiency. The optimal hyperparameters for supervised fine-tuning were a batch size of 16 and a learning rate of 1e-5, trained over 15 epochs with 500 warmup steps and a weight decay of 0.01, utilizing the AdamW optimizer. Gradient accumulation steps were set to 2. QLoRA quantization was applied with 4-bit precision, and its specific hyperparameters included a low-rank (r') of 16, an alpha  $(\alpha$ ') of 32, and a dropout rate of 0.05. Preference optimization was performed with a learning rate of 5.0e-7 over 3 epochs and a beta value of 0.2.

#### **E ABLATION STUDY**

To understand the contribution of each component within our proposed Agentic-RAG framework, we designed an ablation study. By systematically evaluating the impact of removing individual components, we gain valuable insights into their role in the framework's overall performance. The following ablation experiments were conducted:

- (a) Effect of dynamic prompting mechanism(DPM):
  - We compared the performance of the Agentic-RAG framework with and without the dynamic prompting mechanism.
- (b) Role of sub-agent specialization(SAS):
  - We evaluated the Agentic-RAG framework using a single, universal sub-agent for all tasks versus specialized subagents for each task.
- (c) Instruction-tuning(IT) vs. no fine-tuning(NIT):
  - We compared the performance of SLMs with instructiontuning against their performance without any fine-tuning.
- (d) Effectiveness of direct preference optimization (DPO):
  - We evaluated the framework's performance with and without DPO and assessed how aligning SLMs with preferred outcomes impacts the accuracy and reliability of predictions.

Our study investigates the impact of different components on the overall performance of the framework, 'SelfExtend-Agentic-RAG W/Llama 3 - 8B", in time series forecasting, anomaly detection, and classification tasks across various benchmark datasets. We systematically disable each component (dynamic prompting mechanism (DPM), sub-agent specialization (SAS), instruction-tuning (IT), or direct preference optimization (DPO)) and compare the results to the full framework. Tables 14 and 15 detail the forecasting performance, highlighting that the original framework consistently achieves the lowest error rates in MAE, RMSE, and MAPE across different horizons and datasets. This indicates the crucial role of each component in improving forecasting accuracy. Table 16 focuses on anomaly detection tasks, showing the original framework's superior precision, recall, and F1-score compared to its ablated variants. The original framework consistently achieves higher metrics scores across anomaly benchmark datasets such as SWaT, WADI, SMAP, MSL, and HAI. The significant performance drop observed in the ablated variants underscores the importance of the integrated components, demonstrating their synergistic contribution to enhancing anomaly

detection capabilities. For classification tasks, the original framework excels, as demonstrated in Tables 17 and 18, achieving the highest accuracy, precision, and recall across datasets like PeMSD3, PeMSD4, PeMSD7, METR-LA, PeMSD7(M), PeMSD8, and PEMSBAY. The superior performance in classification tasks, coupled with the significant drop observed in ablated variants, highlights the critical role each component plays in the original framework's success. This comprehensive analysis underscores the importance of integrating all components to maximize performance across forecasting, anomaly detection, and classification tasks. The synergistic contribution of the dynamic prompting mechanism, sub-agent specialization, instruction-tuning, and direct preference optimization is evident in the consistent superiority of the Agentic-RAG framework compared to its ablated variants.

| Methods                               | PeMSD3 |       |       | PeMSD4 |       |       | PeMSD7 |       |       | PeMSD8 |       |       | PeMSD7(M) |      |       |
|---------------------------------------|--------|-------|-------|--------|-------|-------|--------|-------|-------|--------|-------|-------|-----------|------|-------|
| Wethous                               | MAE    | RMSE  | MAPE  | MAE       | RMSE | MAPE  |
| Baseline W/O DPM                      | 15.31  | 23.37 | 12.63 | 20.10  | 30.35 | 11.42 | 22.92  | 35.96 | 9.63  | 16.13  | 25.18 | 8.45  | 2.70      | 5.61 | 6.88  |
| Baseline W/O SAS                      | 14.46  | 21.85 | 11.81 | 19.07  | 28.37 | 10.75 | 20.92  | 32.47 | 8.83  | 15.13  | 23.13 | 7.90  | 2.57      | 5.15 | 6.47  |
| Baseline W/O IT                       | 21.62  | 33.01 | 16.85 | 30.06  | 43.77 | 16.18 | 30.43  | 47.95 | 13.86 | 22.45  | 35.67 | 11.96 | 3.95      | 7.49 | 10.00 |
| Baseline W/O DPO                      | 13.53  | 20.45 | 10.97 | 18.11  | 26.89 | 10.08 | 19.82  | 31.77 | 8.44  | 14.63  | 21.82 | 7.40  | 2.42      | 4.89 | 6.23  |
| SelfExtend-Agentic-RAG W/Llama 3 - 8B | 13.01  | 19.48 | 10.53 | 17.46  | 25.54 | 9.52  | 19.02  | 29.97 | 8.03  | 14.03  | 20.98 | 7.04  | 2.33      | 4.68 | 5.88  |

Table 14: The table shows the ablation study results for 12-sequence-to-12-sequence forecasting tasks on benchmark datasets using multiple evaluation metrics. The performance of the ablated variants drops compared to the original framework.

| Datasets | Methods                             |      | orizon | @3   | Н    | rizon( | @6    | Horizon@12 |      |       |
|----------|-------------------------------------|------|--------|------|------|--------|-------|------------|------|-------|
| Datasets | Wethous                             | RMSE | MAE    | MAPE | RMSE | MAE    | MAPE  | RMSE       | MAE  | MAPE  |
|          | Baseline W/O DPM                    | 4.84 | 2.42   | 6.06 | 6.28 | 3.14   | 8.10  | 7.23       | 3.74 | 10.24 |
|          | Baseline W/O SAS                    | 4.48 | 2.23   | 5.66 | 5.97 | 2.99   | 7.77  | 6.86       | 3.43 | 9.81  |
| METR-LA  | Baseline W/O IT                     | 7.05 | 3.23   | 8.09 | 8.69 | 4.18   | 10.80 | 10.08      | 5.00 | 13.65 |
|          | Baseline W/O DPO                    | 4.19 | 2.12   | 5.36 | 5.72 | 2.74   | 7.15  | 6.49       | 3.28 | 9.04  |
|          | SelfExtend-Agentic-RAG W/Llama 3-8B | 4.03 | 2.02   | 5.05 | 5.43 | 2.61   | 6.75  | 6.23       | 3.12 | 8.53  |
|          | Baseline W/O DPM                    | 1.94 | 0.97   | 1.96 | 3.02 | 1.45   | 3.01  | 3.74       | 1.94 | 3.77  |
|          | Baseline W/O SAS                    | 1.79 | 0.90   | 1.82 | 2.79 | 1.35   | 2.86  | 3.47       | 1.75 | 3.61  |
| PEMS-BAY | Baseline W/O IT                     | 2.84 | 1.38   | 2.77 | 4.02 | 1.94   | 4.02  | 5.03       | 2.60 | 5.16  |
|          | Baseline W/O DPO                    | 1.69 | 0.85   | 1.73 | 2.62 | 1.26   | 2.64  | 3.25       | 1.68 | 3.32  |
|          | SelfExtend-Agentic-RAG W/Llama 3-8B | 1.62 | 0.81   | 1.63 | 2.52 | 1.21   | 2.51  | 3.12       | 1.62 | 3.14  |

Table 15: The table presents the ablation study results for the forecasting task performed on the METR-LA and PEMS-BAY datasets, evaluated using multiple metrics. All methods utilized 12 historical sequences to forecast 3, 6, or 12 future sequences.

Table 16: The table showcases the experimental findings from the ablation study conducted on anomaly detection benchmark datasets, reporting the precision, recall, and F1-score metrics.

| Methods                | SWaT  |       |       | WADI  |       |       | SMAP  |       |       | MSL   |       |       | HAI   |       |       |
|------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Without                | P(%)  | R(%)  | F1(%) | P(%)  | R(%)  | F1    | P(%)  | R(%)  | F1(%) | P(%)  | R(%)  | F1(%) | P(%)  | R(%)  | F1(%) |
| Baseline W/O DPM       | 79.57 | 78.52 | 74.07 | 83.49 | 78.92 | 76.32 | 83.27 | 83.13 | 84.18 | 81.98 | 82.24 | 82.48 | 46.61 | 45.14 | 42.59 |
| Baseline W/O SAS       | 88.54 | 86.84 | 83.33 | 88.77 | 82.48 | 80.37 | 87.52 | 84.12 | 84.18 | 88.30 | 84.76 | 84.49 | 52.44 | 50.52 | 48.52 |
| Baseline W/O IT        | 39.79 | 39.26 | 37.04 | 39.45 | 36.79 | 36.03 | 39.30 | 39.59 | 39.62 | 39.24 | 38.95 | 38.82 | 23.31 | 22.45 | 21.30 |
| Baseline W/O DPO       | 95.49 | 93.87 | 87.04 | 94.79 | 88.97 | 85.68 | 94.31 | 94.00 | 94.11 | 94.16 | 91.92 | 91.29 | 55.44 | 53.76 | 50.54 |
| Agentic-RAG W/Llama-8B | 99.47 | 98.15 | 92.59 | 98.63 | 91.97 | 90.08 | 98.24 | 98.97 | 99.04 | 98.11 | 97.37 | 97.04 | 58.27 | 56.13 | 53.24 |

| Dataset                           | PeMSD3   |           |        |          | PeMSD4    |        |          | PeMSD7    |        | METR-LA  |           |        |
|-----------------------------------|----------|-----------|--------|----------|-----------|--------|----------|-----------|--------|----------|-----------|--------|
| Dataset                           | Accuracy | Precision | Recall |
| Baseline W/O DPM                  | 77.12%   | 75.43%    | 76.89% | 77.25%   | 75.67%    | 76.44% | 78.32%   | 76.55%    | 77.21% | 80.14%   | 78.89%    | 80.67% |
| Baseline W/O SAS                  | 81.23%   | 79.45%    | 80.78% | 82.67%   | 80.55%    | 81.32% | 83.89%   | 81.67%    | 82.44% | 84.12%   | 83.67%    | 84.45% |
| Baseline W/O IT                   | 25.45%   | 22.78%    | 24.12% | 22.67%   | 20.56%    | 21.34% | 26.12%   | 25.34%    | 24.56% | 25.67%   | 24.12%    | 23.89% |
| Baseline W/O DPO                  | 88.67%   | 87.23%    | 88.45% | 90.12%   | 88.56%    | 89.23% | 90.78%   | 89.12%    | 88.67% | 90.45%   | 89.67%    | 90.23% |
| SelfExtend-Agentic-RAG W/Llama-8B | 93.01%   | 91.56%    | 92.31% | 94.02%   | 92.82%    | 93.56% | 95.03%   | 94.02%    | 94.21% | 95.82%   | 95.02%    | 95.24% |

Table 17: The table presents the ablation study results, evaluating the performance across various metrics for time series classification tasks on the PeMSD3, PeMSD4, PeMSD7, and METR-LA benchmark datasets.

| Dataset                           | P        | eMSD7(M)  |        |          | PeMSD8    |        | PEMS-BAY |           |        |  |
|-----------------------------------|----------|-----------|--------|----------|-----------|--------|----------|-----------|--------|--|
| Dataset                           | Accuracy | Precision | Recall | Accuracy | Precision | Recall | Accuracy | Precision | Recall |  |
| Baseline W/O DPM                  | 75.41%   | 73.21%    | 74.42% | 76.02%   | 74.81%    | 75.23% | 76.81%   | 75.42%    | 76.02% |  |
| Baseline W/O SAS                  | 82.23%   | 80.52%    | 81.14% | 83.14%   | 81.32%    | 82.01% | 83.62%   | 82.11%    | 82.73% |  |
| Baseline W/O IT                   | 37.61%   | 36.12%    | 36.54% | 38.02%   | 36.81%    | 37.23% | 38.61%   | 37.42%    | 37.92% |  |
| Baseline W/O DPO                  | 90.02%   | 88.73%    | 89.21% | 90.54%   | 89.32%    | 89.83% | 91.01%   | 89.73%    | 90.32% |  |
| SelfExtend-Agentic-RAG W/Llama-8B | 94.02%   | 92.54%    | 93.02% | 95.04%   | 94.03%    | 94.52% | 96.01%   | 95.01%    | 95.53% |  |

Table 18: This table presents the results of an ablation study comparing the performance of various Agentic-RAG framework variants. The study evaluates performance on three benchmark datasets – PeMSD7(M), PeMSD8, and PEMS-BAY – across different metrics for time series classification tasks.