SPAQR: SAMPLING-AWARE PATCH ALLOCATION AND QUERY-CONDITIONED RETRIEVAL FOR IMTS FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Irregular multivariate time series (IMTS) forecasting is challenging because observation intervals are uneven within variables and timestamps are unaligned across variables. Patch-based methods can preserve the original observations and their timestamps, yet they commonly determine partitions without allocating a shared patch budget according to cross-variable sampling heterogeneity. Moreover, multi-scale views are typically constructed before a target variable-time query identifies the historical evidence required for a forecast. To this end, we propose SPAQR, Sampling-aware Patch Allocation and Query-conditioned Retrieval for IMTS Forecasting. SPAQR first performs sampling-aware patch allocation, distributing a fixed per-sample patch budget across variables and feasible temporal splits according to observation density, temporal span, and internal gaps. It then applies multi-scale support encoding to retain coarse, boundary, and transition evidence within the resulting fixed supports. Finally, query-conditioned evidence retrieval uses each target variable and forecast time to read global context and softly weight the supports and their internal states without modifying the partition. Across four public IMTS benchmarks and sixteen representative baselines, SPAQR achieves the lowest mean MSE and MAE on every dataset under a common protocol. Ablations and sensitivity studies further characterize the behavior of its allocation and retrieval designs.

1 INTRODUCTION

Irregular multivariate time series (IMTS) arise in clinical monitoring (Harutyunyan et al., 2019), climate observation (Menne et al., 2015), biomechanical sensing (Kidger et al., 2020), and financial forecasting (Shukla & Marlin, 2021b). In contrast to regularly sampled series, IMTS exhibit two forms of irregularity: (1) observation intervals within a variable are uneven, and (2) timestamps and sampling rates are unaligned across variables. IMTS forecasting therefore entails estimating the value of a target variable at a queried future time from a history that is sparse, asynchronous, and non-uniformly sampled. This setting is not simply a problem of missing entries, because the timing of observations may itself reflect the underlying process or the way it is monitored. Effective forecasting models must consequently retain this timing information while characterizing dependencies across variables and temporal ranges, which is essential for early warning and prospective decision-making in data-limited settings (Harutyunyan et al., 2019; Zhang et al., 2024).

Existing IMTS forecasting methods differ primarily in how they represent an irregular history. Interpolation inserts estimated values at unobserved timestamps, creating spurious observations that may introduce estimation error (Shukla & Marlin, 2019; 2021a; Liu et al., 2026a). Time alignment maps variable-specific schedules onto a shared timeline, adding filled values or masked placeholders at unobserved variable-time pairs and potentially distorting the native sampling pattern (Zhang et al., 2024; Li et al., 2025; Liu et al., 2026a). Graph and hypergraph models instead connect events through time and variable relations, but a time-variable bipartite graph cannot directly relay information between variables that never share a timestamp (Yalavarthi et al., 2024; Li et al., 2025; Luo et al., 2024).

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

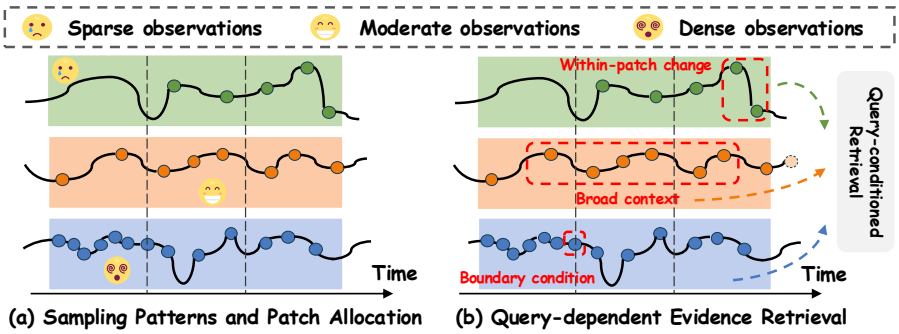


Figure 1: **Motivation of SPAQR.** *(a) Sampling Patterns and Patch Allocation*. Variables with different sampling patterns require different representation capacity. *(b) Query-dependent Evidence Retrieval*. Target variable-time queries can require broad context, boundary conditions, or within-patch change from the same fixed history.

In recent years, patch-based methods have offered a complementary representation that retains the original observations and their timestamps rather than first creating a regular grid. They also organise temporally nearby observations into compact local units, from which temporal and cross-variable dependencies can be learned. Existing designs construct such patches using fixed-duration segments or variable-wise adaptive windows (Zhang et al., 2024; Liu et al., 2025; Hu et al., 2025; Luo et al., 2025; Liu et al., 2026b). Yet these designs do not explicitly treat a fixed per-sample patch budget as a resource to be allocated across variables according to their individual sampling patterns. As illustrated in Figure 1a, for an IMTS history, variables may differ substantially in observation density, temporal span, and the locations of long gaps. Applying the same initial patch configuration can therefore devote insufficient capacity to densely observed variables, while reserving unnecessary capacity for sparsely observed ones. The first challenge is consequently *how to allocate a bounded number of patches across variables and temporal regions* in a manner that reflects their distinct sampling patterns.

In addition, patch allocation only determines what evidence can be retained from the history, but cannot decide which retained evidence should inform a specific forecasting query. Multi-level IMTS models establish representations at several temporal ranges to retain broad context and local temporal information. For example, temporal-warping approaches generate and combine representations at different warped granularities (Zhang et al., 2023). Hierarchical patch models aggregate fine patch representations through successive layers (Luo et al., 2025), while recursive approaches encode subsamples over a sequence of time periods and fuse global-to-local representations (Li et al., 2026). However, in these cases, the temporal views or levels are established during encoding, before a particular target variable-time query determines the evidence needed for prediction. In fact, as shown in Figure 1b, a forecast query is defined by both a target variable and a future time, and different queries can require different temporal information from the same fixed history: broad context may be useful for one query, whereas a local condition near a patch boundary or the change within a patch may be more pertinent for another. Therefore, the second challenge is *how to retrieve information from a fixed, compact patch history adaptively* according to the temporal evidence demanded by each query, instead of solely relying on pre-specified temporal scales.

To address these interrelated challenges, we propose Sampling-aware Patch Allocation and Query-conditioned Retrieval for IMTS Forecasting (**SPAQR**). **To account for cross-variable sampling heterogeneity, we design sampling-aware patch allocation**. Each non-empty variable receives one initial patch, after which the remaining patch slots compete across variables and feasible temporal splits under a fixed per-sample budget. This allocation assigns every observation to exactly one patch while allowing representation capacity to follow observation density, temporal span, and internal gaps. Once the partition is fixed, SPAQR processes the observations in every patch at two learned temporal-retention scales and encodes four complementary states: a coarse summary, left- and right-boundary summaries, and a transition summary. These states distinguish broad context, local conditions at the two patch boundaries, and ordered change across the interval, rather than mixing them in one pooled vector. **To address query-dependent temporal evidence, we introduce query-conditioned evidence retrieval**. For each target variable-time query, the decoder first

retrieves global coarse context from all valid patches and then jointly weights the patches and their internal states. It can therefore read broad context, boundary conditions, or within-patch change from the same allocated partition according to temporal relations, patch geometry, and target-source variable relevance, without altering patch boundaries.

Our main contributions are as follows:

- We propose SPAQR, a query-conditioned IMTS forecasting framework that treats bounded patch construction and forecast-specific reading as two interrelated decisions: allocating representation capacity across heterogeneous variables and retrieving the temporal evidence relevant to each query.
- We develop sampling-aware patch allocation and query-conditioned evidence retrieval. The former distributes a fixed patch budget across variables and feasible temporal splits, whereas the latter uses a target variable-time query to select coarse, boundary, and transition information from the resulting fixed partition.
- We conduct extensive experiments on four IMTS benchmarks against sixteen representative baselines. The results show that SPAQR improves forecasting accuracy, while ablation and sensitivity analyses characterize the behavior of its allocation and retrieval designs.

2 Related Work

2.1 Irregular Multivariate Time Series Modeling

Time-aware LSTM and AdaCare incorporate elapsed-time or multi-scale clinical representations (Baytas et al., 2017; Ma et al., 2020), whereas GRU-D applies temporal decay to missing values (Che et al., 2018). BRITS combines bidirectional recurrent models with imputation (Cao et al., 2018). IP-Net constructs interpolation-based representations, while mTAN maps observations to learned reference times through attention (Shukla & Marlin, 2019; 2021a; Li & Marlin, 2020). These methods allow standard sequence models to process irregular inputs, but their filled or estimated values are not part of the original observation history and may introduce additional error.

Other methods preserve observations as individual events. SeFT treats observations as a set (Horn et al., 2020); Raindrop models time-varying variable relations with a graph (Zhang et al., 2022); and GraFITi represents observations using a time-variable bipartite graph (Yalavarthi et al., 2024). HyperIMTS further uses hyperedges to capture higher-order relations, whereas ASTGI constructs an adaptive interaction graph for observations (Li et al., 2025; Liu et al., 2026a). Graph models for incomplete series also reconstruct missing values before forecasting (Cini et al., 2022; Marisca et al., 2022; Chen et al., 2024). These approaches retain the original observation times and model asynchronous dependencies, but graph construction based on observed variable-time pairs cannot directly connect variables that never share a timestamp (Yalavarthi et al., 2024; Li et al., 2025).

2.2 Patch-based and Multi-scale Modeling for IMTS

Patch-based methods group nearby observations into local temporal units without first filling an entire regular grid. In regularly sampled MTS, PatchTST, Crossformer, Triformer, and iTransformer use patch or variable-specific representations to capture temporal and cross-variable dependencies (Nie et al., 2023; Zhang & Yan, 2023; Cirstea et al., 2022; Liu et al., 2024). Fixed-duration patches can contain very different numbers of observations when sampling is heterogeneous. tPatchGNN divides IMTS into temporal patches and learns relations among patch representations with graph neural networks (Zhang et al., 2024). TimeCHEAT also uses fixed-duration patches to learn temporal, variable, and observation representations (Liu et al., 2025), whereas APN learns adaptive temporal windows for patch aggregation (Liu et al., 2026b). Adaptive patching makes boundaries more flexible, but the resulting patch configuration is still determined before considering how limited representation capacity should be distributed across variables with different sampling patterns.

Several studies incorporate information at multiple temporal ranges for IMTS forecasting. Warpformer uses temporal warping to produce representations at different granularities (Zhang et al., 2023). HD-TTS combines graph-based modeling with spatiotemporal downsampling (Marisca et al., 2024), Hi-Patch aggregates fine patch representations through a hierarchical patch graph (Luo et al.,

2025), and ReIMTS recursively constructs representations from long to short time periods (Li et al., 2026). These models retain both broad and local information, but their temporal views are created during encoding, before a target variable and forecast time determine which historical evidence is most useful for a particular prediction.

3 Problem Definition

We first formalize the query-based IMTS forecasting setting so that the allocation and retrieval stages can be specified precisely. Following prior query-based formulations (Zhang et al., 2024; Luo et al., 2025; Li et al., 2026), the historical part of one sample with N variables is

$$\mathcal{X} = \bigcup_{n=1}^{N} \mathcal{X}^n, \qquad \mathcal{X}^n = \{(t_i^n, x_i^n)\}_{i=1}^{L_n}, \tag{1}$$

where $x_i^n \in \mathbb{R}$ is the i-th observed value of variable n, and the timestamps t_i^n are sorted but may be unevenly spaced and unaligned across variables. We normalize timestamps within a sample to $[0, 1]$. A forecast query is a pair $q_j = (v_j, \tau_j)$, where v_j is the target variable and τ_j is a future timestamp. The forecasting model returns

$$\hat{y}_j = f_\theta(\mathcal{X}, q_j). \tag{2}$$

For a batched implementation, observations and queries are padded only for tensor construction. Their masks exclude padding and unavailable targets from all computations; no values are interpolated at unobserved positions. Let $m_j \in \{0, 1\}$ indicate whether a target is available. We optimize the masked mean squared error

$$\mathcal{L}(\theta) = \frac{\sum_j m_j (\hat{y}_j - y_j)^2}{\sum_j m_j}. \tag{3}$$

where the summation is over valid forecast targets in a minibatch.

4 Method

4.1 Framework Overview

To connect the sampling problem with the prediction problem, SPAQR processes an IMTS history through three consecutive stages. Figure 2 provides an overview. First, **sampling-aware patch allocation** divides every non-empty variable into contiguous, non-overlapping supports under a fixed per-sample budget cap. We refer to each allocated patch as a Support. This stage determines how much representation capacity each variable and temporal region receives. Second, **multi-scale support encoding** processes the original observations in each fixed support at a fine and a coarse learned time scale, yielding four complementary within-patch states. Third, **query-conditioned evidence retrieval** uses the target variable and forecast time to read global coarse context and then softly weight relevant supports and their internal states.

Allocation determines how all valid observations are grouped into a bounded support representation and how representation capacity is distributed, whereas retrieval determines which encoded evidence is useful for a particular forecast. The query never changes a support boundary or creates another patch sequence. Sections 4.2, 4.3, and 4.4 describe these stages in turn.

4.2 Sampling-Aware Patch Allocation

To avoid forcing dense observations, sparse observations, and long unobserved gaps into representations determined only by clock time, we allocate a fixed patch budget according to the sampling pattern. For each variable, SPAQR first sorts its valid observations by timestamp. Every non-empty

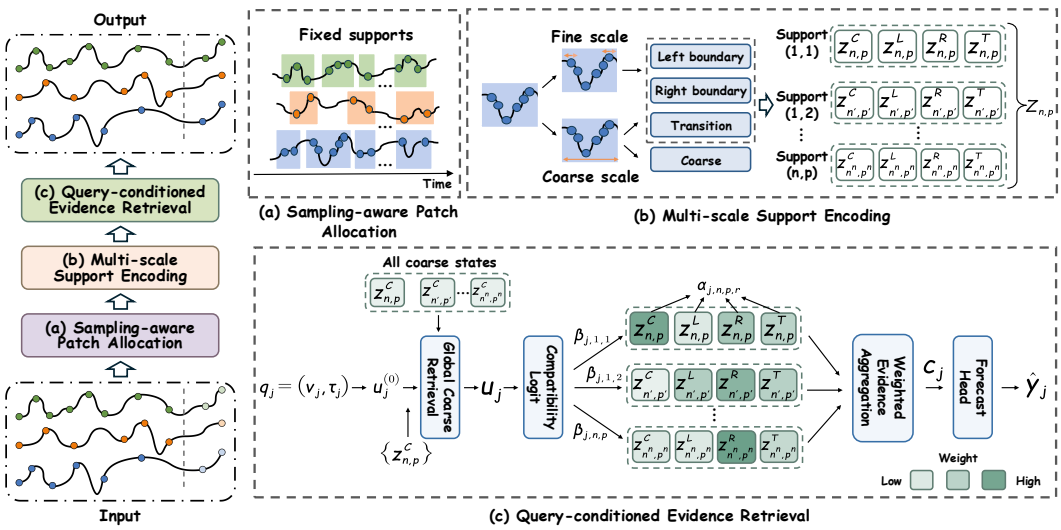


Figure 2: **Overview of SPAQR.** (a) *Sampling-aware Patch Allocation* distributes a fixed per-sample budget across variables and feasible temporal splits. (b) *Multi-scale Support Encoding* produces coarse, boundary, and transition states for each fixed support. (c) *Query-conditioned Evidence Retrieval* reads global coarse context and weights support states for each target variable-time query.

variable receives one support. Let K_n denote the current number of supports for variable n, $P_{\max}$ the per-variable cap, and $\mathcal{B}$ the fixed per-sample budget cap. The feasible allocation satisfies

$$1 \le K_n \le \min(L_n, P_{\max}) \quad \text{for } L_n > 0, \qquad \sum_{n=1}^{N} K_n \le \mathcal{B}. \tag{4}$$

The remaining slots are assigned greedily across all variables and feasible cuts. Consider a current support containing ranked observations $[a, b]$ of variable n, and a candidate cut after c, where $a \le c < b$. Let m_ℓ and m_r be the observation counts on the two sides, D, D_ℓ, and D_r their temporal spans, and $\delta_c = t^n_{c+1} - t^n_c$ the gap at the cut. We score this split by

$$G_{n,a,c,b} = \lambda_m \frac{2m_\ell m_r}{m_0^2} + \lambda_s \frac{D^2 - D_\ell^2 - D_r^2}{(D_n + \epsilon)^2} + \lambda_g \min\left\{ \left(\frac{\delta_c}{\bar{\delta}_n + \epsilon} \right)^2, 100 \right\}, \tag{5}$$

where m_0 is the mass-score scale in the count-based split term, D_n is the total observed span of variable n, and $\bar{\delta}_n = D_n / \max(L_n - 1, 1)$ is its typical gap. The three terms favor cuts that balance observation mass, reduce an excessively long temporal span, or place a boundary at a large internal gap, respectively. At every step, SPAQR selects the feasible cut with the largest $G_{n,a,c,b}$ over all variables and current supports. It stops when the budget is reached or no support can be split further.

This procedure produces a sample-specific hard partition. Each original observation is assigned to exactly one support, and the supports of a variable are contiguous and non-overlapping. The allocation uses only timestamps and observation masks, rather than observed values or forecast queries. It consequently changes neither the observed values nor their timestamps.

For later retrieval, we denote the p-th support of variable n by $\mathcal{S}_{n,p}$. Every $\mathcal{S}_{n,p}$ retains its observations and a six-dimensional geometry vector

$$\mathbf{g}_{n,p} = \left[\mu_{n,p}, s_{n,p}, e_{n,p}, \log(1 + m_{n,p}), \log(1 + 10^3 d_{n,p}), \log(1 + 10^3 \delta^{\max}_{n,p}) \right], \tag{6}$$

where $\mu_{n,p}$, $s_{n,p}$, and $e_{n,p}$ are the mean, first, and last timestamps; $m_{n,p}$ is the observation count; $d_{n,p} = e_{n,p} - s_{n,p}$ is the span; and $\delta^{\max}_{n,p}$ is the largest internal gap. We apply logarithmic scaling to

the count, span, and maximum-gap entries to control their numerical ranges. These attributes allow later stages to distinguish supports with similar content but different sampling geometry.

4.3 Multi-Scale Support Encoding

Once allocation has fixed which observations belong together, the next challenge is to preserve broad context, boundary conditions, and local change without constructing several patch sequences. We therefore encode the same observations in every support at two learned retention scales. For observation (t_i, x_i) in $\mathcal{S}_{n,p}$, its initial representation is

$$\mathbf{e}_i = \mathrm{MLP}_{\mathrm{obs}}\left[x_i \,\|\, \phi(t_i) \,\|\, \phi(t_i - \mu_{n,p})\right], \tag{7}$$

where $\phi(\cdot)$ is a shared continuous-time Fourier encoding. The absolute and support-relative time features distinguish both when an observation occurred and where it lies inside its support.

To make the two scales responsive to physical time rather than observation index, we use an order-preserving state scan. For a scale ℓ, elapsed time Δt_i, and incoming state $\mathbf{h}_{i-1}$, the scan first decays the preceding state and then gates new evidence:

$$\begin{aligned} \tilde{\mathbf{h}}_{i-1} &= \exp(-\Delta t_i/\ell)\mathbf{h}_{i-1}, \;\; \mathbf{z}_i = \tanh\Big(\mathbf{W}_z[\mathbf{e}_i \| \tilde{\mathbf{h}}_{i-1} \| \log(1+\Delta t_i/\ell)]\Big), \\ \boldsymbol{\gamma}_i &= \sigma\Big(\mathbf{W}_\gamma[\mathbf{e}_i \| \tilde{\mathbf{h}}_{i-1} \| \log(1+\Delta t_i/\ell)]\Big), \;\; \mathbf{h}_i = (1-\boldsymbol{\gamma}_i)\odot\tilde{\mathbf{h}}_{i-1} + \boldsymbol{\gamma}_i \odot \mathbf{z}_i. \end{aligned} \tag{8}$$

The scan runs in both temporal directions at a fine scale ℓ_f and a coarse scale ℓ_c. Both scales are learned and constrained as

$$\ell_f = 0.5\,\sigma(r_f) + 10^{-3}, \qquad \ell_c = \ell_f + \mathrm{softplus}(r_c), \tag{9}$$

where r_f and r_c are learned scalar parameters. This parameterization ensures $0 < \ell_f < \ell_c$. Thus, the coarse scan retains evidence across longer gaps, whereas the fine scan emphasizes local conditions. No additional supports are generated.

To turn these scans into forecast-relevant evidence, SPAQR stores four fixed-capacity states for every support. A **coarse state** combines the two coarse directional scans with support geometry and a variable embedding. A **left-boundary state** and a **right-boundary state** combine the fine and coarse scans at the corresponding boundary. A **transition state** combines both scales and directions with explicit dynamics, including the first and last values, their difference, a signed log-slope, and the transformed count, span, and maximum-gap entries of $\mathbf{g}_{n,p}$. Denoting these learned projections by ψ_C, ψ_L, ψ_R, and ψ_T, we obtain

$$\mathbf{Z}_{n,p} = \left[\mathbf{z}^C_{n,p}, \mathbf{z}^L_{n,p}, \mathbf{z}^R_{n,p}, \mathbf{z}^T_{n,p}\right] \in \mathbb{R}^{4\times d}. \tag{10}$$

The four states are computed for every valid support before decoding. They do not constitute four temporal scales; they are complementary summaries produced from two learned time scales and made available for query-conditioned reading.

4.4 Query-Conditioned Evidence Retrieval

Because the temporal evidence useful for a forecast depends on both its target variable and its future time, SPAQR performs query-conditioned evidence retrieval with a target variable-time query rather than pooling support states uniformly. For $q_j = (v_j, \tau_j)$, the initial query representation is

$$\mathbf{u}^{(0)}_j = \mathrm{MLP}_q\left[\phi(\tau_j) \| \mathbf{e}^{\mathrm{var}}_{v_j}\right], \tag{11}$$

where $\mathbf{e}^{\mathrm{var}}_{v_j}$ is a learned embedding of the target variable. The decoder uses this query in two stages.

First, global context is read from the coarse state of every valid support. Stacked multi-head attention layers update $\mathbf{u}_j$ by attending to $\{\mathbf{z}^C_{n,p}\}$. In addition to content similarity, their logits receive

a learned bias based on the query-to-support temporal distance, distance to the support interval, whether the query falls inside that interval, the count/span/gap entries of $\mathbf{g}_{n,p}$, and the target–source variable pair. This stage gives every query access to broad historical context before it chooses detailed within-support information.

Second, the updated query performs joint support–state weighting. Let (n, p) range over valid support pairs, let $r \in \{C, L, R, T\}$ index the four states of $\mathcal{S}_{n,p}$, and let $\ell_{j,n,p,r}$ be a state compatibility logit. The state distribution within a support is

$$\alpha_{j,n,p,r} = \mathrm{softmax}_r(\ell_{j,n,p,r}). \tag{12}$$

State compatibility is conditioned on content, the appropriate state anchor (support centre, start, or end), temporal and sampling geometry, and target–source variable relevance. The state evidence also contributes to the support score:

$$\beta_{j,n,p} = \mathrm{softmax}_{(n,p)}\left(\frac{\langle \mathbf{W}_q \mathbf{u}_j, \mathbf{W}_k \mathbf{z}^C_{n,p}\rangle}{\sqrt{d}} + b^{\mathrm{sup}}_{j,n,p} + \log \frac{1}{4}\sum_r \exp(\ell_{j,n,p,r})\right). \tag{13}$$

Here, $b^{\mathrm{sup}}_{j,n,p}$ is the learned support-level geometry and variable-relation bias. The resulting context is a joint weighted combination over valid support pairs,

$$\mathbf{c}_j = \sum_{n,p}\sum_r \beta_{j,n,p}\alpha_{j,n,p,r}\mathbf{W}_v \mathbf{z}^r_{n,p}, \qquad \hat{y}_j = \mathrm{MLP}_{\mathrm{out}}\big(\mathrm{GatedUpdate}(\mathbf{u}_j, \mathbf{c}_j)\big). \tag{14}$$

The query therefore changes only the soft reading weights. Patch boundaries, observation assignments, and the four support states remain fixed after encoding.

5 Experiments

5.1 Experimental Setup

Datasets and Baselines. We evaluated SPAQR on four standard IMTS forecasting benchmarks: *HumanActivity*, *USHCN*, *PhysioNet'12* (*P12*), and *MIMIC-III* (*MIMIC*). Following prior IMTS forecasting studies (Li et al., 2025; Liu et al., 2026a; Li et al., 2026), we used the standard preprocessing pipeline and divided each dataset into training, validation, and test sets in an approximately 8:1:1 ratio. Dataset statistics, forecasting windows, and preprocessing details are provided in Appendix A.1. We compared SPAQR with sixteen baselines: **GRU-D** (Che et al., 2018), **NeuralFlows** (Biloš et al., 2021), **GNeuralFlow** (Mercatali et al., 2024), **CRU** (Schirmer et al., 2022), **SeFT** (Horn et al., 2020), **mTAN** (Shukla & Marlin, 2021a), **PrimeNet** (Chowdhury et al., 2023), **Raindrop** (Zhang et al., 2022), **GraFITi** (Yalavarthi et al., 2024), **tPatchGNN** (Zhang et al., 2024), **Warpformer** (Zhang et al., 2023), **HD-TTS** (Marisca et al., 2024), **Hi-Patch** (Luo et al., 2025), **APN** (Liu et al., 2026b), **TFMixer** (Qiu et al., 2026), and **ReIMTS** equipped with the GraFITi backbone (Li et al., 2026). Their descriptions and references are deferred to the Appendix A.3.

Implementation details. All experiments used PyTorch 2.5.1+cu121 on an NVIDIA GeForce RTX 4090 GPU. We used Adam and minimized masked mean squared error (MSE) over valid forecast targets. The learning rate remained constant during the first two epochs and subsequently decayed geometrically by a factor of 0.8 per epoch. Training ran for at most 300 epochs, with early stopping after ten validation epochs without improvement. For SPAQR, the default total Support budgets were 48, 20, 144, and 384 for HumanActivity, USHCN, P12, and MIMIC, respectively; these correspond to four nominal budget slots per variable, while the allocation itself remained sample specific. Unless otherwise stated, $P_{\max} = 8$, the mass-score scale was $m_0 = 4$, and the coarse-global retrieval depth was 4. We adopted the reported hyperparameter configurations of the baselines and further tuned their principal hyperparameters on the validation set. For evaluation, we computed MSE and MAE over valid forecast targets in the dataset-specific preprocessed value space. Each setting was evaluated with five independent random seeds, and we report the mean and standard deviation.

Table 1: Forecasting performance on four IMTS benchmarks, reported as mean ± standard deviation over five runs. The best results are in **bold** and the second-best are underlined. Lower values indicate better forecasting accuracy.

Model	HumanActivity		USHCN		Physio'12		MIMIC	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PrimeNet	2.6842±0.0039	1.3264±0.0011	0.4569±0.0014	0.4780±0.0011	0.7939±0.0000	0.6847±0.0002	0.9704±0.0000	0.6609±0.0002
NeuralFlows	0.1685±0.0072	0.3113±0.0083	0.4168±0.0429	0.3353±0.0075	0.4047±0.0031	0.4459±0.0028	0.6592±0.0046	0.5314±0.0027
GRU-D	0.1787±0.0381	0.3173±0.0317	0.2690±0.0100	0.2915±0.0038	0.3418±0.0024	0.3997±0.0011	0.5177±0.0095	0.4518±0.0047
SeFT	1.3767±0.0016	0.9749±0.0006	0.4098±0.0003	0.4441±0.0020	0.7813±0.0042	0.6792±0.0026	0.9907±0.0003	0.6690±0.0009
mTAN	0.0850±0.0024	0.2069±0.0042	0.5491±0.0666	0.4975±0.0356	0.3719±0.0034	0.4234±0.0027	0.8976±0.0278	0.6457±0.0115
CRU	0.1398±0.0070	0.2630±0.0110	0.2371±0.0151	0.3170±0.0097	0.6169±0.0027	0.5776±0.0022	0.6567±0.0052	0.5248±0.0037
Raindrop	0.0945±0.0106	0.2160±0.0160	0.2135±0.0216	0.3142±0.0179	0.3453±0.0010	0.4023±0.0009	0.5142±0.0036	0.4452±0.0036
GNeuralFlow	0.3466±0.0809	0.4295±0.0437	0.2549±0.0368	0.3380±0.0227	0.8454±0.0630	0.6801±0.0243	0.8889±0.0249	0.6435±0.0052
tPatchGNN	0.0455±0.0028	0.1282±0.0072	0.2355±0.0959	0.2878±0.0206	0.3086±0.0016	0.3658±0.0021	0.4677±0.0153	0.4053±0.0065
Warpformer	0.0448±0.0003	0.1257±0.0006	0.1826±0.0048	0.2715±0.0056	0.3089±0.0055	0.3657±0.0060	0.4520±0.0033	0.3984±0.0023
HD_TTS	0.0473±0.0037	0.1325±0.0116	0.1729±0.0067	0.2776±0.0098	0.3058±0.0008	0.3633±0.0013	0.4555±0.0043	0.4076±0.0063
Hi-Patch	0.0481±0.0006	0.1256±0.0014	0.2342±0.0167	0.2830±0.0075	0.3176±0.0036	0.3758±0.0028	0.4840±0.0020	0.4189±0.0050
GraFITi	0.0419±0.0004	0.1146±0.0005	0.1848±0.0025	0.2740±0.0094	0.3032±0.0006	0.3600±0.0006	0.4109±0.0041	0.3968±0.0024
APN	0.0441±0.0006	0.1230±0.0019	0.2044±0.0278	0.2792±0.0067	0.3129±0.0006	0.3660±0.0014	0.4213±0.0017	0.4003±0.0057
TFMixer	0.2782±0.0225	0.3847±0.0191	0.1711±0.0117	0.2629±0.0051	0.3088±0.0032	0.3690±0.0024	0.4623±0.0031	0.4117±0.0028
ReIMTS	0.0425±0.0146	0.1146±0.0005	0.1652±0.0019	0.2771±0.0069	0.2916±0.0005	0.3498±0.0002	0.5065±0.0411	0.4400±0.0256
SPAQR (Ours)	**0.0409±0.0007**	**0.1054±0.0009**	**0.1582±0.0009**	**0.2549±0.0056**	**0.2899±0.0016**	**0.3489±0.0018**	**0.3956±0.0018**	**0.3780±0.0023**

Table 2: Ablation study on four IMTS benchmarks, reported in MSE (mean ± standard deviation) over five runs. The full SPAQR is shown in the first row; lower values indicate better performance.

Variant	HumanActivity	USHCN	Physio'12	MIMIC
SPAQR (full)	**0.0409 ± 0.0007**	**0.1582 ± 0.0009**	**0.2899 ± 0.0016**	**0.3956 ± 0.0018**
rp. sampling-aware allocation	0.0419 ± 0.0004	0.1979 ± 0.0225	0.3030 ± 0.0025	0.4177 ± 0.0037
w/o multi-scale encoding	0.0462 ± 0.0027	0.1780 ± 0.0050	0.2997 ± 0.0009	0.4142 ± 0.0085
w/o query-conditioned retrieval	0.0481 ± 0.0039	0.1916 ± 0.0171	0.3032 ± 0.0012	0.4217 ± 0.0073
w/o coarse-global retrieval	0.0442 ± 0.0012	0.1781 ± 0.0047	0.3010 ± 0.0007	0.4271 ± 0.0042

5.2 Main Results

Table 1 compares SPAQR with all baselines under the common forecasting protocol. SPAQR achieved the lowest mean MSE and MAE on every dataset metric combination in the table. Its gains were apparent on both high-frequency HumanActivity and the sparse, long-horizon USHCN benchmark, as well as on the higher-dimensional clinical datasets P12 and MIMIC. Relative to the best non-SPAQR mean in each column, the largest reductions were 4.24% in MSE on USHCN and 8.03% in MAE on HumanActivity. The improvement on P12 was smaller (0.58% MSE and 0.26% MAE), so we interpret this result as the best mean under the evaluated protocol rather than as evidence of a statistically significant margin.

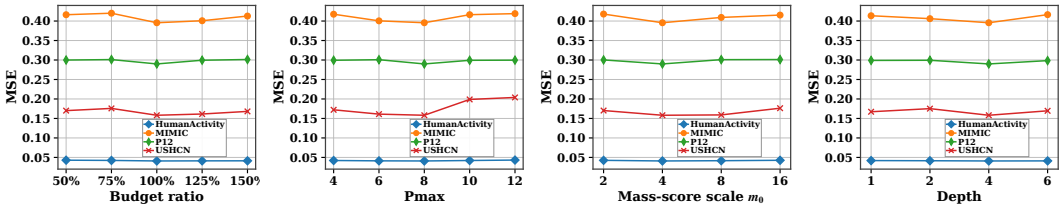
The comparison also places the result against several strong recent alternatives, including fixed- or hierarchically constructed patch methods, graph-based event models, and recursive multi-scale modeling. None matched SPAQR across all eight columns. These results are consistent with the two design decisions tested below: allocating bounded representation capacity according to the observed sampling pattern, and reading the resulting fixed partition with a target variable-time query. They do not, by themselves, isolate the contribution of either decision; that role is addressed by the controlled ablations in Section 5.3.

5.3 Ablation Study

Table 2 and Table 3 compare full SPAQR with four variants. (1) *rp. sampling-aware allocation* replaces the adaptive partition with four equal-duration bins per variable, masking empty bins. (2)

Table 3: Ablation study on four IMTS benchmarks, reported in MAE (mean ± standard deviation) over five runs. The full SPAQR is shown in the first row; lower values indicate better performance.

Variant	HumanActivity	USHCN	Physio'12	MIMIC
SPAQR (full)	**0.1054 ± 0.0009**	**0.2549 ± 0.0056**	**0.3489 ± 0.0018**	**0.3780 ± 0.0023**
rp. sampling-aware allocation	0.1179 ± 0.0008	0.2745 ± 0.0077	0.3640 ± 0.0049	0.3861 ± 0.0049
w/o multi-scale encoding	0.1277 ± 0.0077	0.2699 ± 0.0039	0.3606 ± 0.0037	0.3828 ± 0.0059
w/o query-conditioned retrieval	0.1347 ± 0.0099	0.2774 ± 0.0135	0.3649 ± 0.0055	0.3840 ± 0.0057
w/o coarse-global retrieval	0.1245 ± 0.0035	0.2758 ± 0.0070	0.3586 ± 0.0017	0.3925 ± 0.0077



(a) Budget ratio. (b) $P_{\max}$. (c) Mass-score scale (d) Coarse-global depth.

Figure 3: Sensitivity to the total Support budget, the per-variable Support cap, the mass-score scale, and the coarse-global retrieval depth. Each point is the mean MSE over the evaluated seeds; colors and markers identify the four datasets.

multi-scale encoding retains only the coarse state of each support. (3) *w/o query-conditioned retrieval* averages the four support states uniformly and removes the state-evidence term from support scores, while retaining query-dependent weighting across supports. (4) *w/o coarse-global retrieval* removes the preceding coarse-global query layers but retains the final support-state retrieval. Every variant increased mean MSE and MAE on all four benchmarks. Overall, these results indicate that SPAQR's allocation, encoding, and retrieval designs jointly contribute to effective IMTS forecasting.

5.4 Sensitivity Analysis

Figure 3 examines four hyperparameters of SPAQR. (1) The *total Support budget* controls the number of supports allocated across variables. A ratio of 100% denotes dataset-specific defaults of 48, 20, 144, and 384 for HumanActivity, USHCN, P12, and MIMIC, respectively. These defaults achieved the lowest mean MSE on all four benchmarks. (2) The *per-variable Support cap* limits the supports assigned to each variable, and $P_{\max} = 8$ performed best. (3) The *mass-score scale* controls the count-based split term, with $m_0 = 4$ yielding the lowest errors. (4) The *coarse-global retrieval depth* controls the number of global query layers, and a depth of 4 performed best. These results support the default allocation and retrieval settings used in our experiments.

6 Conclusion

We presented SPAQR for IMTS forecasting, which separates bounded support construction from forecast-specific evidence retrieval. SPAQR allocates a fixed per-sample patch budget across variables and feasible temporal splits according to their sampling patterns, then applies multi-scale support encoding to retain coarse, boundary, and transition evidence in the resulting fixed supports. For each target variable-time query, query-conditioned evidence retrieval first reads global coarse context and then jointly weights supports and their internal states without modifying patch boundaries. Across four public benchmarks and sixteen representative baselines under a common protocol, SPAQR achieved the lowest mean MSE and MAE on every reported dataset–metric combination. Ablations further showed consistent performance drops when sampling-aware allocation, complementary support states, or query-conditioned evidence retrieval is removed or simplified, indicating that a fixed, sampling-aware support partition can retain and provide the temporal evidence required by different forecasting queries.

AI USE STATEMENT

We used generative AI tools to assist with language polishing, proofreading, and formatting of the manuscript. All AI-assisted edits were reviewed by the authors and checked against the implementation, experimental records, and cited literature. No unpublished data or personally identifiable information was provided to these tools. We take full responsibility for the final content of this work, including all text, claims, results, and artifacts.

REPRODUCIBILITY STATEMENT

Our code is available at: https://anonymous.4open.science/r/SPAQR-34242SJF93JD83NF.

REFERENCES

Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jain, and Jiayu Zhou. Patient subtyping via time-aware LSTM networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 65–74, 2017.

Marin Biloš, Venkatesh Somnath, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural ODEs. In *Advances in Neural Information Processing Systems*, volume 34, pp. 21325–21337, 2021.

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. BRITS: Bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems*, 2018.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8:6085, 2018. doi: 10.1038/s41598-018-24271-9.

Xiaodan Chen, Xiucheng Li, Bo Liu, and Zhijun Li. Biased temporal convolution graph network for time series forecasting with missing values. In *International Conference on Learning Representations*, 2024.

Ranak Roy Chowdhury, Jiacheng Li, Xiyuan Zhang, Dezhi Hong, Rajesh K. Gupta, and Jingbo Shang. PrimeNet: Pre-training for irregular multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7184–7192, 2023. doi: 10.1609/aaai.v37i6.25876.

Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the gaps: Multivariate time series imputation by graph neural networks. In *International Conference on Learning Representations*, 2022.

Razvan-Gabriel Cirstea, Chenjuan Guo, Bin Yang, Tung Kieu, Xuanyi Dong, and Shirui Pan. Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 2022.

Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):96, 2019.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, Damian Roqueiro, and Karsten Borgwardt. Set functions for time series. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 4353–4363, 2020.

Zhangyi Hu, Jiemin Wu, Hua Xu, Mingqian Liao, Ninghui Feng, Bo Gao, Songning Lai, and Yutao Yue. IMTS is worth time $\times$ channel patches: Visual masked autoencoders for irregular multivariate time series prediction. In *International Conference on Machine Learning*, 2025.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):1–9, 2016.

Patrick Kidger, James Morrill, James Foster, and Terry J. Lyons. Neural controlled differential equations for irregular time series. In *Advances in Neural Information Processing Systems*, 2020.

Boyuan Li, Yicheng Luo, Zhen Liu, Junhao Zheng, Jianming Lv, and Qianli Ma. HyperIMTS: Hypergraph neural network for irregular multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 35502–35518, 2025.

Boyuan Li, Zhen Liu, Yicheng Luo, and Qianli Ma. Learning recursive multi-scale representations for irregular multivariate time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026.

Steven Cheng-Xian Li and Benjamin M. Marlin. Learning from irregularly-sampled time series: A missing data perspective. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 5937–5946, 2020.

Mengzhao Liu, Defu Cao, and Yuxuan Chen. Timecheat: An efficient approach for irregular time series modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18861–18869, 2025. doi: 10.1609/aaai.v39i18.34076.

Xvyuan Liu, Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Chenjuan Guo, Bin Yang, and Jilin Hu. ASTGI: Adaptive spatio-temporal graph interactions for irregular multivariate time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Xvyuan Liu, Xiangfei Qiu, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Jilin Hu, and Bin Yang. Rethinking irregular time series forecasting: A simple yet effective baseline. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23873–23881, 2026b. doi: 10.1609/AAAI.V40I28.39563.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024.

Yicheng Luo, Zhen Liu, Linghao Wang, Binquan Wu, Junhao Zheng, and Qianli Ma. Knowledge-empowered dynamic graph network for irregularly sampled medical time series. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yicheng Luo, Bowen Zhang, Zhen Liu, and Qianli Ma. Hi-Patch: Hierarchical patch GNN for irregular multivariate time series. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 41494–41519, 2025.

Liantao Ma, Junyi Gao, Yasha Wang, Chaohe Zhang, Jiangtao Wang, Wenjie Ruan, Wen Tang, Xin Gao, and Xinyu Ma. AdaCare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 825–832, 2020.

Ivan Marisca, Andrea Cini, and Cesare Alippi. Learning to reconstruct missing data from spatiotemporal graphs with sparse observations. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32069–32082, 2022.

Ivan Marisca, Cesare Alippi, and Filippo Maria Bianchi. Graph-based forecasting with missing data through spatiotemporal downsampling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34846–34865, 2024.

Matthew J. Menne, Claude N. Williams Jr, and Russell S. Vose. United states historical climatology network daily temperature, precipitation, and snow data. Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, 2015.

Giangiacomo Mercatali, Andre Freitas, and Jie Chen. Graph neural flows for unveiling systemic interactions among irregularly sampled time series. In *Advances in Neural Information Processing Systems*, volume 37, pp. 57183–57206, 2024. doi: 10.52202/079017-1822.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Xiangfei Qiu, Kangjia Yan, Xvyuan Liu, Xingjian Wu, and Jilin Hu. Bridging time and frequency: A joint modeling framework for irregular multivariate time series forecasting. In *International Conference on Machine Learning*, 2026.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Mona Schirmer, Ahmed Eltayeb, Stefan Lessmann, Maja Rudolph, and Marco Körner. Modeling irregular time series with continuous recurrent units. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 19388–19405, 2022.

Satya Narayan Shukla and Benjamin M. Marlin. Interpolation-prediction networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2019.

Satya Narayan Shukla and Benjamin M. Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021a.

Satya Narayan Shukla and Benjamin M. Marlin. A survey on principles, models and methods for learning from irregularly sampled time series. *arXiv preprint arXiv:2012.00168*, 2021b.

Ikaro Silva, George Moody, Daniel J. Scott, Leo A. Celi, and Roger G. Mark. Predicting in-hospital mortality of ICU patients: The PhysioNet/Computing in cardiology challenge 2012. In *Computing in Cardiology*, pp. 245–248, 2012.

Vijaya Krishna Yalavarthi, Kiran Madhusudhanan, Randolf Scholz, Nourhan Ahmed, Johannes Burchert, Shayan Jawed, Stefan Born, and Lars Schmidt-Thieme. Grafiti: Graphs for forecasting irregularly sampled time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16255–16263, 2024.

Jiawen Zhang, Shun Zheng, Wei Cao, Jiang Bian, and Jia Li. Warpformer: A multi-scale modeling approach for irregular clinical time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3273–3285, 2023. doi: 10.1145/3580305.3599543.

Weijia Zhang, Chenlong Yin, Hao Liu, Xiaofang Zhou, and Hui Xiong. Irregular multivariate time series forecasting: A transformable patching graph neural networks approach. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 60179–60196, 2024.

Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations*, 2022.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Table 4: Dataset-specific configurations for SPAQR. $\mathcal{B}$ is the total Support budget, d is the hidden dimension, and heads is the number of attention heads.

Dataset	Observation setting / forecast targets	Budget $\mathcal{B}$	d	Heads	Batch size
HumanActivity	$3000/300$	48	128	4	32
USHCN	$150/3$	20	32	1	16
P12	$36/3$	144	64	1	32
MIMIC	$72/3$	384	64	1	32

A Detailed Experimental Setup

A.1 Dataset Details

We provide detailed descriptions of the four public IMTS benchmarks used in our experiments, including their sources, characteristics, and forecasting protocols. The sample counts and variable counts follow the common PyOmniTS-style benchmark protocol used by recent IMTS forecasting studies (Zhang et al., 2024; Yalavarthi et al., 2024; Li et al., 2026).

MIMIC-III (MIMIC). MIMIC-III is a de-identified critical-care database containing measurements collected during intensive-care stays (Johnson et al., 2016). We used the clinical time series from the first 48 hours of each stay, comprising 21,250 IMTS samples and 96 variables. Each forecast uses a 36-hour lookback window and predicts the next three observed timestamps.

PhysioNet'12 (P12). P12 contains multivariate records from the first 48 hours of ICU stays released for the PhysioNet/Computing in Cardiology Challenge 2012 (Goldberger et al., 2000; Silva et al., 2012). The benchmark comprises 11,981 samples with 36 physiological variables, including laboratory measurements and vital signs. We use a 36-hour lookback window and predict the next three observed timestamps.

HumanActivity. HumanActivity contains biomechanical trajectories recorded by body-mounted sensors while individuals perform different activities (Rubanova et al., 2019). It includes 1,359 samples and 12 variables describing three-dimensional positions. The observations are naturally irregular because individual sensors follow slightly different sampling schedules. We use a 3000-millisecond history to predict the subsequent 300 milliseconds.

USHCN. The United States Historical Climatology Network (USHCN) contains long-term climate records collected by weather stations across the United States (Menne et al., 2015). Following the established IMTS protocol, we use a four-year subset from 1996 to 2000, yielding 1,114 samples and five climate variables. Missing records make the resulting series irregular. Each forecast uses a three-year lookback window and predicts the next three observed timestamps.

Preprocessing and splits. We followed the standard preprocessing pipeline and the 8:1:1 train–validation–test split used in prior IMTS forecasting work (Zhang et al., 2024; Yalavarthi et al., 2024; Li et al., 2026). The original irregular event locations and validity masks were retained without interpolation or regular-grid construction. Value and timestamp scaling followed the dataset-specific benchmark preprocessing, and all reported errors were computed in that preprocessed model space. Padding and validity masks were used only to batch variable-length instances; SPAQR does not interpolate observations or construct a regular time grid.

A.2 Implementation Details

Table 4 summarizes the dataset-specific configurations used for SPAQR. History and prediction durations are reported in milliseconds for HumanActivity, years for USHCN, and hours for P12 and MIMIC. The total Support budget $\mathcal{B}$ limits the number of supports allocated to each IMTS instance; the resulting allocation remains sample specific.

All experiments were implemented in PyTorch 2.5.1+cu121 and run on an NVIDIA GeForce RTX 4090 GPU. We trained models with Adam to minimize masked MSE over valid forecast targets. The initial learning rate was 10^{-3}, held constant for the first two epochs, and then multiplied by 0.8 after each epoch. Training lasted for at most 300 epochs and used early stopping after ten consecutive validation epochs without improvement. Each configuration was evaluated with five random seeds, and we report the mean and standard deviation.

Unless otherwise stated, SPAQR used a per-variable Support cap of $P_{\max} = 8$ and the mass-score scale of $m_0 = 4$. The mass, span, and gap terms in the allocation score used equal coefficients, $\lambda_m = \lambda_s = \lambda_g = 1.0$. The coarse-global retrieval depth was $D = 4$. We adopted the reported configurations for the baseline methods and tuned their principal hyperparameters on the validation split.

A.3 Baseline Details

We compare SPAQR with sixteen IMTS baselines. We use the architectures and primary hyperparameter configurations reported by the original papers, and tune principal hyperparameters on the validation split. For a reference implementation originally designed for classification, we replace only its task-specific output layer with a linear forecasting head.

GRU-D. GRU-D is a recurrent model for multivariate time series with missing values. It uses temporal decay to account for the elapsed time since previous observations (Che et al., 2018).

NeuralFlows. NeuralFlows models continuous-time latent dynamics with neural flows as an alternative to explicit neural ODE integration (Biloš et al., 2021).

GNeuralFlow. GNeuralFlow extends neural-flow modeling with graph neural networks to represent systemic interactions among irregularly sampled variables (Mercatali et al., 2024).

CRU. CRU uses continuous recurrent units to update a latent representation between irregular observations (Schirmer et al., 2022).

SeFT. SeFT represents a time series as a set of observation points and applies permutation-invariant set functions for prediction (Horn et al., 2020).

mTAN. mTAN uses multi-time attention to map irregular observations to a fixed set of learned reference points (Shukla & Marlin, 2021a).

PrimeNet. PrimeNet is a pre-training approach for irregular multivariate time series that learns transferable representations before downstream prediction (Chowdhury et al., 2023).

Raindrop. Raindrop is a graph-guided model that represents time-varying dependencies among variables in irregularly sampled time series (Zhang et al., 2022).

GraFITi. GraFITi uses a time–variable bipartite graph to represent irregular observations and forecast future values (Yalavarthi et al., 2024).

tPatchGNN. tPatchGNN first forms transformable temporal patches and then applies graph neural forecasting over their representations (Zhang et al., 2024).

Warpformer. Warpformer constructs multi-scale representations through temporal warping for irregular clinical time series (Zhang et al., 2023).

HD-TTS. HD-TTS combines graph-based forecasting with spatiotemporal downsampling to handle missing observations (Marisca et al., 2024).

Hi-Patch. Hi-Patch is a hierarchical patch graph neural network that aggregates information across patch levels (Luo et al., 2025).

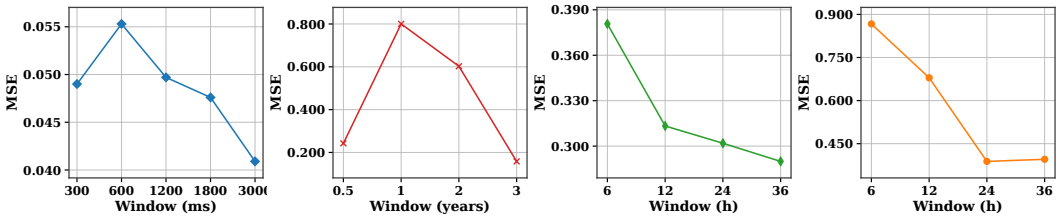
APN. APN is an adaptive patching baseline for IMTS forecasting that learns flexible temporal windows for aggregation (Liu et al., 2026b).

TFMixer. TFMixer jointly models time-domain and frequency-domain information for IMTS forecasting (Qiu et al., 2026).

ReIMTS. ReIMTS recursively constructs multi-scale representations. We use its reported configuration with GraFITi as the backbone, denoted as **ReIMTS** in the main-results table (Li et al., 2026).

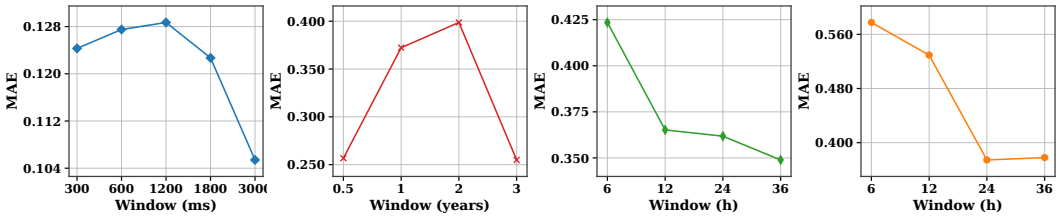
Table 5: Peak training GPU memory and trainable parameter counts on USHCN. All models were measured under the same training configuration.

Model	Peak training memory (GiB)	Parameters (M)
tPatchGNN	0.5611	0.451
Warpformer	0.6852	0.104
GraFITi	0.4805	0.249
APN	0.1855	0.019
SPAQR	**0.1745**	0.137



(a) HumanActivity (b) USHCN (c) Physio'12 (d) MIMIC

Figure 4: **Sensitivity to observation-window duration measured by MSE.** Each point reports mean test MSE over five seeds; lower is better. Panels show HumanActivity, USHCN, P12, and MIMIC, respectively. Durations are reported in milliseconds for HumanActivity, years for USHCN, and hours for P12 and MIMIC. Horizontal positions denote discrete evaluated settings rather than elapsed time.



(a) HumanActivity (b) USHCN (c) Physio'12 (d) MIMIC

Figure 5: **Sensitivity to observation-window duration measured by MAE.** Each point reports mean test MAE over five seeds; lower is better. Panels correspond to HumanActivity, USHCN, P12, and MIMIC, respectively. The dataset-specific forecasting protocol is fixed, but valid target locations change with the observation cutoff. Horizontal positions denote discrete evaluated settings rather than elapsed time.

Table 6: Additional support-allocation ablations on four IMTS benchmarks, reported in MSE (mean ± standard deviation) over five runs. Lower values indicate better performance.

Allocation	HumanActivity	USHCN	Physio'12	MIMIC
SPAQR	**0.0409 ± 0.0007**	**0.1582 ± 0.0009**	**0.2899 ± 0.0016**	**0.3956 ± 0.0018**
Equal time	0.0419 ± 0.0004	0.1979 ± 0.0225	0.3030 ± 0.0025	0.4177 ± 0.0037
Matched equal count	0.0425 ± 0.0001	0.1965 ± 0.0178	0.3010 ± 0.0015	0.4207 ± 0.0039
No splitting	0.0423 ± 0.0004	0.1784 ± 0.0079	0.3021 ± 0.0014	0.4158 ± 0.0020

Table 7: Additional support-allocation ablations on four IMTS benchmarks, reported in MAE (mean ± standard deviation) over five runs. Lower values indicate better performance.

Allocation	HumanActivity	USHCN	Physio'12	MIMIC
SPAQR	**0.1054 ± 0.0009**	**0.2549 ± 0.0056**	**0.3489 ± 0.0018**	**0.3780 ± 0.0023**
Equal time	0.1179 ± 0.0008	0.2745 ± 0.0077	0.3640 ± 0.0049	0.3861 ± 0.0049
Matched equal count	0.1185 ± 0.0016	0.2895 ± 0.0185	0.3610 ± 0.0018	0.3940 ± 0.0039
No splitting	0.1170 ± 0.0012	0.2716 ± 0.0052	0.3632 ± 0.0025	0.3843 ± 0.0023

B ADDITIONAL EXPERIMENTS

B.1 Training Memory and Parameter Count

Table 5 reports peak training GPU memory and the number of trainable parameters on USHCN, measured under the same training configuration as the main experiments. SPAQR used 0.1745 GiB of peak training memory, the lowest value among the compared models, while containing 0.137 M trainable parameters. Thus, on USHCN, the sampling-aware allocation and query-conditioned retrieval design maintained a small training-memory footprint without requiring a large parameter count. This comparison concerns training memory and model size only.

B.2 Sensitivity to Observation-Window Duration

We varied the observation-window duration while keeping all model hyperparameters and each dataset's forecasting protocol fixed. Figures 4 and 5 report the resulting test MSE and MAE. Both metrics decreased as the P12 window increased. HumanActivity achieved its lowest MSE and MAE with a 3000-millisecond window, despite variation among shorter windows. On MIMIC, the 24-hour window slightly outperformed the 36-hour window on both metrics. USHCN showed a non-monotonic pattern: MSE was lowest with a three-year window, whereas MAE was lowest with a 0.5-year window. Forecast targets are selected after the observation cutoff. Changing the window therefore changes both the available history and the locations of valid targets. These curves consequently compare complete window settings rather than isolate the effect of extending the observed history.

B.3 Additional Ablations on Support Allocation

Tables 6 and 7 compare SPAQR with three alternative support-construction rules. (1) *Equal time*, corresponding to *rp. sampling-aware allocation* in the main text, assigns each variable $\mathcal{B}/N$ equal-duration windows and masks empty windows. (2) *Matched equal count* first retains the sample- and variable-specific support count K_n produced by SPAQR and then divides the valid observations of each variable into K_n contiguous supports with approximately equal observation counts. (3) *No splitting* assigns all valid observations of each non-empty variable to a single support. SPAQR achieved the lowest mean MSE and MAE on all four benchmarks. Overall, these results indicate that sampling-aware allocation provides a more effective support construction strategy for IMTS forecasting.