RETHINKING PATCH AGGREGATION FOR IRREGULAR TIME SERIES FORECASTING WITH VORONOI QUADRATURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Patching has become the efficient backbone of irregular multivariate time series (IMTS) forecasting, yet the step that turns scattered records into a patch token has remained largely unexamined. We revisit that step by formalizing a patch operator as a pair: a support kernel delimiting its temporal scope, and an aggregation measure assigning integration mass to the observations within that support. Prior work has increasingly refined the support kernel while leaving the aggregation measure at one unit of base mass per observation, so patch content estimates an average under the observation-induced measure rather than over physical time. We formalize this discrepancy and prove that it equals the patch-local covariance between temporal content and observation intensity, divided by the mean intensity. Being deterministic, the discrepancy resists averaging: it vanishes on a regular grid, grows whenever recording density correlates with content, and is fixed before any learnable module reads the token, beyond the reach of added capacity. Guided by this identity, we propose VQPatch, which treats every record as the representative of its Voronoi temporal cell, converts cell widths into quadrature weights, and retains the observation pattern as a bounded prediction condition. The resulting token provably approximates the physical-time patch integral at a rate governed by the cell mesh and independent of the sampling intensity, at no asymptotic cost beyond sorting. Across four IMTS benchmarks and seventeen baselines, VQPatch attains the lowest MAE on every dataset, improves MSE over the strongest patch baseline by up to 9.36%, and cuts peak training memory by 61.4%. The code is available at `https://anonymous.4open.science/r/VQ-Patch_code-366D/`.

1 INTRODUCTION

Irregular multivariate time series (IMTS) arise wherever variables are recorded at non-uniform times, as in healthcare monitoring, scientific sensing, and environmental observation (Rubanova et al., 2019; De Brouwer et al., 2019; Yalavarthi et al., 2024; Zhang et al., 2024). Forecasting them underpins early warning, system monitoring, and proactive decision making. The difficulty is not missingness alone: intervals are unequal within a variable, timestamps are asynchronous across variables, and long stretches may carry no measurement at all. Taken together, these properties dissolve the shared temporal grid on which conventional multivariate forecasters rely, and they leave informative temporal and cross-variable structure considerably harder to expose.

Two broad routes have emerged in response. The first maps irregular records onto a canonical temporal representation through interpolation, imputation, or alignment (Shukla & Marlin, 2021; Zhou et al., 2026), which restores standard temporal operators at the price of a densified input and a larger volume of information passed downstream. The second keeps the records intact and models them through continuous-time dynamics or relational structure defined over individual observations (Rubanova et al., 2019; De Brouwer et al., 2019; Biloš et al., 2021; Yalavarthi et al., 2024; Li et al., 2025). Efficient attention and structured sequence models relieve part of the resulting cost (Zhou et al., 2021; Gu et al., 2022; Gu & Dao, 2023), yet dependencies continue to be resolved at the granularity of single records. Patching charts a more economical course: records are first aggregated into a small number of local temporal units, and dependency modeling then proceeds over

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.

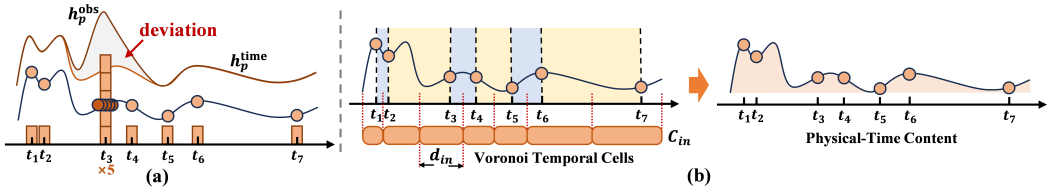


Figure 1: **Rethinking patch aggregation.** (a) With one unit of mass per record, a densely recorded interval accumulates disproportionate mass, causing point-normalized aggregation to deviate from the physical-time average; (b) Voronoi temporal cells assign each record a represented temporal coverage, whose cell width serves as a quadrature weight for estimating physical-time content.

compact tokens (Zhang et al., 2024; Liu et al., 2025; Hu et al., 2025; Liu et al., 2026b). This design delivers competitive accuracy at low cost, and it now underlies several leading IMTS forecasters.

Effort in this line has concentrated almost entirely on the temporal scope of a patch: fixed windows gave way to the transformable patches of t-PatchGNN (Zhang et al., 2024), and hard boundaries to the learned soft boundaries of APN (Liu et al., 2026b). We turn instead to an operation that has survived these advances untouched, namely the rule by which the selected records are combined into a token. A patch operator is a pair: a support kernel, which fixes the temporal scope of a token, and an aggregation measure, which fixes the mass carried by each record within that scope. Normalized aggregation leaves the measure at its default of one unit of mass per valid record. On a regular grid the choice is innocuous, since equal intervals render record count proportional to temporal coverage; under irregular sampling the two quantities part company, and a densely recorded interval comes to dominate a patch even when it spans a small fraction of the physical time that the patch covers, as illustrated in Fig. 1(a).

Insight. *Adaptive support and adaptive mass serve different ends. The support delimits the records that a token reads; the measure fixes the quantity that the token estimates. Only the second determines whether patch content describes the signal or the recording process.*

This distinction admits an exact characterization. Treating the records inside a patch as a realization of a point process, point-normalized aggregation converges to an average of the temporal content under a measure tilted by the observation intensity, and its discrepancy from the physical-time average collapses to a single quantity: a normalized covariance between the signal and the recording density. Three properties of that quantity matter. It vanishes when density carries no content-correlated structure, which is precisely why the regular-grid intuition holds there and fails elsewhere. It is deterministic rather than stochastic, so repetition across records, samples, or epochs does not dilute it. And it is fixed before any learnable module reads the token, leaving a larger model to compensate only indirectly, and only within the sampling regime seen during training.

The same analysis points to the remedy. If content mass is to reflect physical time, each record should speak for the portion of the timeline that it represents. The Voronoi principle asserts exactly this property for spatial samples, and transplanted to the one-dimensional temporal axis it yields record-centered cells whose boundaries are midpoints between neighboring timestamps. Cell widths then serve as the natural quadrature weights of the patch integral: they adapt to local recording density, they require only linear time after sorting, and they turn aggregation into an estimator of physical-time content whose error is governed by the cell mesh rather than by the sampling intensity, as illustrated in Fig. 1(b).

Building on this view, we propose **VQPatch**, a Voronoi quadrature patching framework organized around three coordinated strategies. Content–Observation Decoupling (COD) assigns separate computational roles to physical-time content and the observation process. Voronoi Temporal Quadrature (VTQ) forms the content token from cell widths, so that record count alone no longer sets content mass. Observation Process Conditioning (OPC) returns the observation pattern as a bounded prediction condition, preserving informative acquisition behavior without letting it redefine the token. Furthermore, by evaluating quadrature weights in closed form, VQPatch bypasses the costly continuous patch transformations and soft assignment matrices of existing baselines. To our knowledge, this is the first account to attribute the accuracy loss of irregular patching to the aggregation measure rather than to patch support, model capacity, or downstream dependency modeling, and VQPatch is the first IMTS patch operator whose output is provably an estimator of physical-time content. Our contributions are as follows.

- **A new design axis.** We recast a patch operator as a support kernel paired with an aggregation measure. Prior work refines the kernel; we show that the measure, left at one unit of mass per record, is where irregularity enters the representation.
- **An identity that capacity cannot undo.** We prove that the discrepancy of point-normalized aggregation from the physical-time average equals the patch-local covariance between temporal content and observation intensity, divided by the mean intensity. It is deterministic, vanishes only under content-independent recording, and is fixed before any learnable module reads the token.
- **An operator-level remedy with a guarantee.** VQPatch converts Voronoi widths into quadrature weights and applies the observation pattern as a bounded prediction condition post-aggregation. The token provably approximates the physical-time patch integral at a rate set by the cell mesh, subsumes point-normalized patching as a special case, and adds no asymptotic cost beyond sorting.
- **Mechanism-level evidence.** Against seventeen baselines on four benchmarks, VQPatch attains the lowest MAE on every dataset, improves MSE over the strongest patch baseline by up to 9.36%, and cuts peak training memory by 61.4%. Gains scale with the variability of recording density, and ablating the quadrature measure incurs the sharpest performance drop.

2 Related Work

From Record-Level Modeling to Compact Representations. Research on IMTS forecasting reads as a sequence of answers to one question of representation granularity, each resolving a limitation of the previous answer at a new cost. The earliest answer removes irregularity before modeling, by interpolating records onto reference points as in mTAN (Shukla & Marlin, 2021) or aligning them to a canonical grid as in KAFNet (Zhou et al., 2026); standard operators become applicable again, at the price of reconstruction error and a densified input. The second answer keeps the records and shifts the burden to the temporal dynamics: Latent ODE (Rubanova et al., 2019) and GRU-ODE-Bayes (De Brouwer et al., 2019) evolve latent states in continuous time, Neural Flows (Biloš et al., 2021) and CRU (Schirmer et al., 2022) replace the solver with closed-form or filtering updates, and ContiFormer (Chen et al., 2023) attends over continuous time, yet the state must still be advanced between consecutive records. The third answer replaces sequential evolution with relational propagation over records and queries, as in GraFITi (Yalavarthi et al., 2024), GNeuralFlow (Mercatali et al., 2024), HyperIMTS (Li et al., 2025), and ASTGI (Liu et al., 2026a); cross-variable dependencies become flexible, but every record remains a node. The three stages relocate rather than reduce the cost of irregularity, leaving granularity itself as the bottleneck and motivating representations that summarize records before dependency modeling.

Patch Aggregation as the Unexamined Step. Patching was popularized for regularly sampled series by PatchTST (Nie et al., 2023) and was subsequently transferred to irregular data, where the design effort has concentrated almost entirely on the temporal scope of a token. The scope was first made data dependent: t-PatchGNN (Zhang et al., 2024) forms transformable patches that hold varying numbers of records under a uniform temporal resolution. It was then enriched along other axes, with TimeCHEAT (Liu et al., 2025) separating local channel-dependent from global channel-independent structure, VIMTS (Hu et al., 2025) tokenizing time and channel jointly under masked reconstruction, and TFMixer (Qiu et al., 2026) coupling patch mixing with frequency-domain interaction. Most recently, APN (Liu et al., 2026b) replaces hard boundaries with learned soft boundaries and time-aware weights, so that patch scope follows local recording patterns, and it attains strong accuracy with a lightweight architecture. These methods refine patch support from fixed to learned boundaries while the measure stays fixed, yet aggregation still averages records with equal mass. Since support defines the temporal region and measure the estimated quantity, they are not interchangeable under irregular sampling. We retain adaptive support but replace occurrence-based mass with physical-time coverage from Voronoi cells.

3 Theoretical Analysis

Separating patch support from the aggregation measure turns the omission identified in Sec. 2 into an exact statement. For a variable n observed over $[t_{\mathrm{L}}, t_{\mathrm{R}}]$, we characterize the deviation induced by the observation process and give a Voronoi quadrature approximation of physical-time patch content. Sec. 4 instantiates every symbol, and Appendix B collects the assumptions and proofs.

Definition 1 (Temporal Patch Operator)**.** Let $K_{np}(t)$ be the support kernel of patch p and let $\mathbf{z}_n(t)$ be the continuous counterpart of the discrete representation $\mathbf{z}_{\ell n}$. Given an aggregation measure μ, the temporal patch operator is

$$\mathcal{P}_{np}(\mu; \mathbf{z}_n) = \frac{\int_{t_{\mathrm{L}}}^{t_{\mathrm{R}}} K_{np}(t)\, \mathbf{z}_n(t)\, d\mu(t)}{\int_{t_{\mathrm{L}}}^{t_{\mathrm{R}}} K_{np}(t)\, d\mu(t)}. \tag{1}$$

The kernel delimits the region where a patch aggregates and the measure distributes mass within it; point-normalized and physical-time aggregation are two choices of μ, the latter written $\mathcal{P}_{np}(dt; \mathbf{z}_n)$.

Proposition 1 (Observation-Measure Bias Identity)**.** Let $g_n(t) \geq 0$ be the local observation intensity of variable n, let μ_n^{obs} be the observation-induced measure with density proportional to $g_n(t)$, and let ν_{np} be the support-normalized physical-time probability measure induced by K_{np}. Under standard integrability and positivity conditions,

$$\mathcal{P}_{np}(\mu_n^{\mathrm{obs}}; \mathbf{z}_n) - \mathcal{P}_{np}(dt; \mathbf{z}_n) = \frac{\mathrm{Cov}_{\nu_{np}}(\mathbf{z}_n, g_n)}{\mathbb{E}_{\nu_{np}}[g_n]}. \tag{2}$$

For vector-valued $\mathbf{z}_n$, the covariance is taken component-wise.

The deviation is therefore driven not by irregular spacing as such, but by the patch-local dependence between temporal content and observation intensity: it vanishes under constant intensity, which recovers the regular-grid case, and under locally constant content. In every other regime it persists for any support kernel, since the aggregation measure remains observation-induced. Appendix B.2 gives the proof and the resulting bounds.

Proposition 2 (Voronoi Quadrature Approximation)**.** After coalescing exact duplicate timestamps, let $(\widetilde{t}_{in}, \widetilde{\mathbf{z}}_{in})_{i=1}^{\widetilde{L}_n}$ be the ordered records of variable n, and assign each observation a boundary-clipped Voronoi temporal cell of width d_{in}. The resulting physical-time content token is

$$\mathbf{u}_{np} = \frac{\sum_{i=1}^{\widetilde{L}_n} K_{np}(\widetilde{t}_{in})\, d_{in}\, \widetilde{\mathbf{z}}_{in}}{\sum_{i=1}^{\widetilde{L}_n} K_{np}(\widetilde{t}_{in})\, d_{in} + \varepsilon} \;\approx\; \mathcal{P}_{np}(dt; \mathbf{z}_n). \tag{3}$$

If $K_{np}\mathbf{z}_n$ and K_{np} are Lipschitz and the denominator is bounded away from zero, the approximation error of the unstabilized ratio is $O(h_n)$ with mesh $h_n = \max_i d_{in}$, and the rate is independent of the observation intensity g_n.

The width d_{in} supplies quadrature mass and is computed from timestamps alone rather than learned. Appendix B.3 states the constants, the consistency conditions, and the invariances of the operator.

Insight. *The two propositions describe one operator under two measures: point-normalized aggregation is consistent for an intensity-tilted average, Voronoi quadrature for the physical-time average. Correcting the measure therefore changes the quantity being estimated, not merely its variance.*

4 METHOD

Sec. 3 leaves a concrete requirement: content mass must be physical-time coverage, while the observation pattern must stay available to the predictor without re-entering the content token. VQPatch meets both by acting before, during, and after aggregation.

Problem Formulation. Consider a batch of IMTS over a historical window $[t_{\mathrm{L}}, t_{\mathrm{R}}]$, with batch size B, padded row length L, N variables, and Q forecast queries. The inputs are the value tensor $\mathbf{X} \in \mathbb{R}^{B\times L\times N}$, the timestamp tensor $\mathbf{T} \in \mathbb{R}^{B\times L\times 1}$, the observation mask $\mathbf{M} \in \{0,1\}^{B\times L\times N}$, and the query timestamps $\mathbf{Q} \in \mathbb{R}^{B\times Q\times 1}$, where $\mathbf{T}$ supplies a shared row-wise time axis and $\mathbf{M}$ selects the valid timestamps of each variable. Entry $x_{b\ell n}$ is the value of variable n at row timestamp $t_{b\ell}$, and $m_{b\ell n} = 1$ marks a valid record, so missing and padded entries carry zero mask. Targets and target masks are $\mathbf{Y} \in \mathbb{R}^{B\times Q\times N}$ and $\mathbf{M}^y \in \{0,1\}^{B\times Q\times N}$, with each query timestamp q_{bj} shared across variables. The model maps $(\mathbf{X}, \mathbf{T}, \mathbf{M}, \mathbf{Q})$ to $\widehat{\mathbf{Y}} = f_\theta(\mathbf{X}, \mathbf{T}, \mathbf{M}, \mathbf{Q}) \in \mathbb{R}^{B\times Q\times N}$. We omit the batch index below for clarity. Appendix A summarizes the notation.

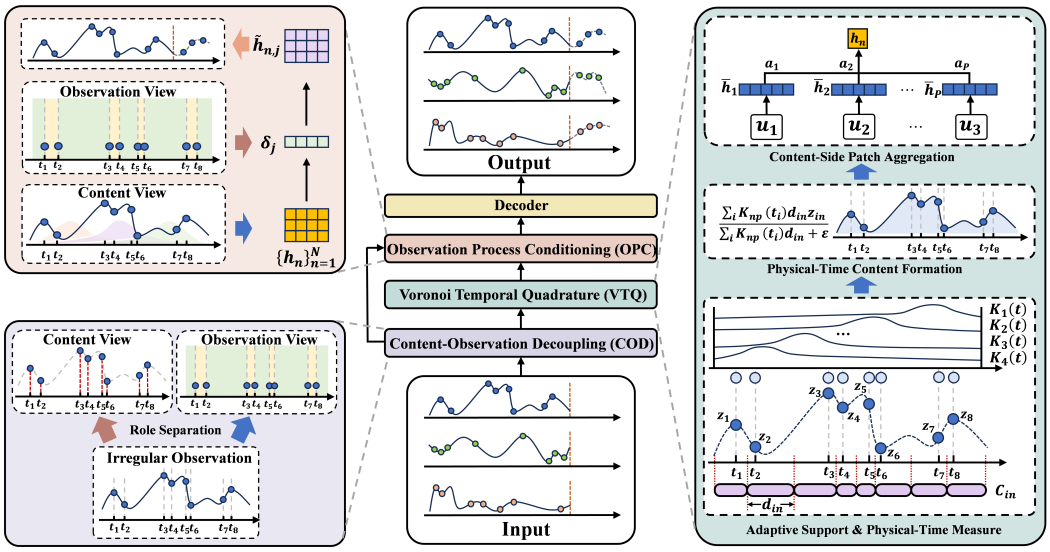


Figure 2: **Overview of VQPatch**. COD separates irregular observations into content and observation views. VTQ forms physical-time content tokens using adaptive patch support and Voronoi cell widths, and aggregates them into variable-level representations. OPC then reintroduces observation information through residual conditioning, followed by the decoder for forecasting.

As shown in Fig. 2, COD routes the input into a content path and a support path. VTQ maps the content path to variable-level physical-time content tokens, while the support path yields query-specific observation support descriptors. OPC conditions the content representation on these descriptors, and a shared predictor produces $\widehat{\mathbf{Y}}$. All parameters are optimized jointly with the mask-aware MSE objective, and batches without valid targets are skipped. No auxiliary observation-process or prediction-mixture objective is used.

4.1 Content–Observation Decoupling (COD)

Role Separation: Temporal Content and Observation Processes. The *temporal content process* is the signal that evolves over physical time, whereas the *observation process* is the mechanism that decides the timing and frequency of measurement. Each valid record combines its value with the learnable time embedding

$$\mathbf{e}_t(t) = [w_s t + b_s, \sin(\mathbf{W}_p t + \mathbf{b}_p)] \in \mathbb{R}^{d_t}, \tag{4}$$

where $w_s, b_s \in \mathbb{R}$ and $\mathbf{W}_p, \mathbf{b}_p \in \mathbb{R}^{d_t-1}$. This yields $\mathbf{z}_{\ell n} = x_{\ell n} \oplus \mathbf{e}_t(t_\ell) \in \mathbb{R}^{d_z}$ with $d_z = 1 + d_t$, where $\oplus$ denotes concatenation. The same embedding is reused for forecast queries.

COD assigns the two processes distinct computational roles by routing the input into a content path $\mathcal{C} = (\mathbf{X}, \mathbf{T}, \mathbf{M})$ and a support path $\mathcal{S} = (\mathbf{T}, \mathbf{M})$. VTQ consumes $\mathcal{C}$, whereas $\mathcal{S}$ enters only through OPC, so the support path accesses neither historical values nor targets. No statistical independence is assumed between the processes; the ordering alone keeps the covariance term of Eq. 2 from re-entering the token through a side path.

4.2 Voronoi Temporal Quadrature (VTQ)

VTQ instantiates **Definition** 1 with the physical-time measure: it retains adaptive temporal support while replacing occurrence-based mass with physical-time coverage.

Support Kernel: Learning Adaptive Temporal Scope. For variable n and patch $p \in \{1, \ldots, P\}$, the Support Kernel is

$$K_{np}(t) = \sigma\left(\frac{r_{np} - t}{\tau_n}\right) \sigma\left(\frac{t - l_{np}}{\tau_n}\right), \tag{5}$$

where l_{np} is a learned left boundary, $r_{np} > l_{np}$ is parameterized by a positive width, and the learned variable-specific $\tau_n > 0$ controls boundary softness. Conventional point normalization assigns unit

base mass to every valid record, making the resulting statistic depend on local record occurrence rather than physical-time coverage; **Proposition** 1 characterizes this Observation Measure Substitution.

Quadrature Aggregation: From Voronoi Cell Widths to Content Tokens. For each variable, valid records are sorted and exact duplicate timestamps are averaged, producing $(\widetilde{t}_{in}, \widetilde{x}_{in})_{i=1}^{\widetilde{L}_n}$. For $\widetilde{L}_n > 1$, we set $\beta_{0n} = t_{\mathrm{L}}$, $\beta_{in} = (\widetilde{t}_{in} + \widetilde{t}_{i+1,n})/2$ for $i = 1, \ldots, \widetilde{L}_n - 1$, and $\beta_{\widetilde{L}_n n} = t_{\mathrm{R}}$. The boundary-clipped Voronoi cell is $C_{in} = [\beta_{i-1,n}, \beta_{in}]$, with Voronoi Cell Width $d_{in} = \beta_{in} - \beta_{i-1,n}$. This width is the physical-time extent represented by record i, rather than a confidence or attention weight; for a single valid record, $d_{1n} = t_{\mathrm{R}} - t_{\mathrm{L}}$. With $\widetilde{\mathbf{z}}_{in} = \widetilde{x}_{in} \oplus \mathbf{e}_t(\widetilde{t}_{in})$, VTQ constructs

$$\mathbf{u}_{np}^{(\gamma)} = \frac{\sum_{i=1}^{\widetilde{L}_n} K_{np}(\widetilde{t}_{in})\, \bar{d}_{in}^{\gamma}\, \widetilde{\mathbf{z}}_{in}}{\sum_{i=1}^{\widetilde{L}_n} K_{np}(\widetilde{t}_{in})\, \bar{d}_{in}^{\gamma} + \varepsilon}, \qquad \bar{d}_{in} = \frac{d_{in}}{t_{\mathrm{R}} - t_{\mathrm{L}}}. \tag{6}$$

Here $K_{np}(\widetilde{t}_{in})$ determines temporal relevance, whereas $\bar{d}_{in}^{\gamma}$ determines physical-time mass. Normalization by the history-window length makes γ unit free. The default $\gamma = 1$ gives Voronoi temporal quadrature, while $\gamma = 0$ recovers point normalization over the coalesced records; we report its sensitivity in Sec. 5.4. Coalescing prevents duplicate timestamps from creating additional temporal mass, and equal effective widths recover point normalization. For a variable with no valid historical record, we define $\mathbf{u}_{np} = \mathbf{0}$; below, $\mathbf{u}_{np}$ denotes the default $\mathbf{u}_{np}^{(1)}$.

Let D denote the hidden dimension and $\boldsymbol{\xi}_n \in \mathbb{R}^D$ a learned query for variable n. Each token is projected as $\mathbf{h}_{np}^{(0)} = \mathbf{W}_{\mathrm{proj}}\mathbf{u}_{np} + \mathbf{b}_{\mathrm{proj}}$ and refined by $\phi(\mathbf{u}_{np}) = \mathrm{LN}(\mathbf{h}_{np}^{(0)} + \mathrm{FFN}(\mathbf{h}_{np}^{(0)}))$. Adding fixed sinusoidal patch-position encoding gives $\bar{\mathbf{h}}_{np} = \phi(\mathbf{u}_{np}) + \mathrm{PE}(p)$. The learned variable query then assigns patch weights and aggregates the tokens:

$$\alpha_{np} = \mathrm{softmax}_p\left(\frac{\boldsymbol{\xi}_n^{\top}\bar{\mathbf{h}}_{np}}{\sqrt{D}}\right), \quad \mathbf{h}_n = \mathrm{LN}\left(\sum_{p=1}^{P} \alpha_{np}\bar{\mathbf{h}}_{np}\right) \in \mathbb{R}^D. \tag{7}$$

Here $\mathbf{W}_{\mathrm{proj}}$, $\mathbf{b}_{\mathrm{proj}}$, and FFN are learned, LN is layer normalization, and PE is fixed, so the weights α_{np} depend only on content-side tokens.

4.3 Observation Process Conditioning (OPC)

VTQ prevents record occurrence from redefining content, while observation patterns may remain predictive. OPC therefore reintroduces observation support after content aggregation as a bounded condition, leaving the Support Kernel, Voronoi Cell Widths, and Physical-Time Content Tokens unchanged.

Observation Support Descriptor. For query q_j, let $\mathcal{T}_j^{\mathrm{seq}} = (t_{j1}^{\mathrm{obs}}, \ldots, t_{jK_j}^{\mathrm{obs}})$ denote the ordered historical timestamp rows no later than q_j at which at least one variable is observed, with duplicate rows retained. OPC constructs

$$\boldsymbol{\rho}_j^{\mathrm{raw}} = \left[\log\frac{1}{\Delta_j + \varepsilon}, \log(1+\kappa_j), \log(1+\kappa), s_j, \omega_j, c_j\right]. \tag{8}$$

Let $W = t_{\mathrm{R}} - t_{\mathrm{L}}$. For $K_j \geq 2$, define $\Delta_j = \max(q_j - t_{jK_j}^{\mathrm{obs}}, 0)$, $s_j = t_{jK_j}^{\mathrm{obs}} - t_{j1}^{\mathrm{obs}}$, and $\omega_j = \max_{1\leq k<K_j}(t_{j,k+1}^{\mathrm{obs}} - t_{jk}^{\mathrm{obs}})$; for $K_j \leq 1$, set $(\Delta_j, s_j, \omega_j) = (W, 0, W)$. The variable-level counts are $\kappa_j = \sum_{\ell: t_\ell \leq q_j} \sum_{n=1}^{N} m_{\ell n}$ and $\kappa = \sum_{\ell=1}^{L}\sum_{n=1}^{N} m_{\ell n}$, while $c_j = \mathrm{clip}(s_j/W, 0, 1)$ is normalized coverage.

Using training-set statistics, the descriptor is standardized as $\bar{\boldsymbol{\rho}}_j = (\boldsymbol{\rho}_j^{\mathrm{raw}} - \boldsymbol{\mu}_\rho)/(\boldsymbol{\sigma}_\rho + \varepsilon)$. Temporal geometry and support counts use $\mathcal{I}_t = \{1, 4, 5, 6\}$ and $\mathcal{I}_s = \{2, 3\}$, respectively. For $a \in \{t, s\}$, we set $\mathbf{r}_j^a = \tanh(\mathrm{clip}(\bar{\boldsymbol{\rho}}_{j,\mathcal{I}_a}, -3, 3))$; $E_t : \mathbb{R}^4 \to \mathbb{R}^R$ and $E_s : \mathbb{R}^2 \to \mathbb{R}^R$ then yield $\mathbf{e}_j^t = E_t(\mathbf{r}_j^t)$ and $\mathbf{e}_j^s = E_s(\mathbf{r}_j^s)$. Clipping at three standard deviations limits extreme conditions.

Reference-Centered Conditioning. OPC centers both embeddings at the standardized zero reference and injects a bounded condition through a low-rank residual path:

$$\begin{aligned} \boldsymbol{\delta}_j &= \frac{1}{2}\tanh\big(\mathbf{W}_t[\mathbf{e}_j^t - E_t(\mathbf{0}_4)] + \mathbf{W}_s[\mathbf{e}_j^s - E_s(\mathbf{0}_2)]\big)\,, \\ \widetilde{\mathbf{h}}_{nj} &= \mathbf{h}_n + \eta\, U(V(\mathrm{LN}(\mathbf{h}_n)) \odot \boldsymbol{\delta}_j)\,. \end{aligned} \tag{9}$$

Here $R < D$ is the low-rank dimension; $\mathbf{W}_t, \mathbf{W}_s : \mathbb{R}^R \to \mathbb{R}^R$ are learned linear maps, while $V : \mathbb{R}^D \to \mathbb{R}^R$ and $U : \mathbb{R}^R \to \mathbb{R}^D$ are linear projections. The factor $1/2$ bounds each coordinate of $\boldsymbol{\delta}_j$ within $(-0.5, 0.5)$, while $\eta = 0.1$ in the formal full-joint VQPatch configuration keeps $\mathbf{h}_n$ as the main path. Finally, the shared two-layer MLP decoder predicts $\widehat{y}_{jn} = f_{\mathrm{dec}}(\widetilde{\mathbf{h}}_{nj} \oplus \mathbf{e}_t(q_j))$ using hidden dimension $2D$, ReLU, dropout, and scalar output; OPC uses neither schedule-specific branches nor multiple predictors.

5 Experiments

5.1 Experimental Setup

Datasets: We use four standard IMTS benchmarks: MIMIC (Johnson et al., 2016) and PhysioNet (Silva et al., 2012) from intensive care, HumanActivity (Rubanova et al., 2019) from biomechanics, and USHCN (Menne et al., 2016) from climate science. They differ in dimensionality, sampling density, and the variability of that density over time, which makes them suitable for testing the account of Sec. 3. Following previous work, each dataset is split into training, validation, and test sets with a ratio of 80%, 10%, and 10%. Appendix E.1 provides descriptions and statistics.

Baselines: We compare VQPatch with seventeen representative baselines in two groups. ***1) Representation, classification, and imputation models***: PrimeNet (Chowdhury et al., 2023), SeFT (Horn et al., 2020), mTAN (Shukla & Marlin, 2021), GRU-D (Che et al., 2016), Raindrop (Zhang et al., 2022), and Warpformer (Zhang et al., 2023). ***2) IMTS forecasting models***: NeuralFlows (Biloš et al., 2021), CRU (Schirmer et al., 2022), GNeuralFlow (Mercatali et al., 2024), t-PatchGNN (Zhang et al., 2024), GraFITi (Yalavarthi et al., 2024), VIMTS (Hu et al., 2025), TFMixer (Qiu et al., 2026), KAFNet (Zhou et al., 2026), HyperIMTS (Li et al., 2025), ASTGI (Liu et al., 2026a), and APN (Liu et al., 2026b). Among them, t-PatchGNN, VIMTS, and APN are patch-based and therefore form the most direct comparison. Appendix E.2 describes each baseline.

Implementation Details: All experiments run in PyTorch 2.6.0+cu124 on a single NVIDIA RTX 4090 GPU. Models are trained with Adam for 50 epochs, with early stopping after 10 epochs without validation improvement. Each experiment uses five seeds, and we report mean$\pm$std. We do not apply the "Drop Last" trick (Qiu et al., 2024; 2025). Appendix E.3 reports the remaining protocol details.

5.2 Main Results

Table 1 compares VQPatch with seventeen baselines on four IMTS datasets. Three observations follow. ***1) VQPatch achieves the best result in seven of eight dataset–metric pairs and the lowest MAE on all four datasets.*** Against APN, the strongest adaptive-patch baseline, it reduces both metrics on every dataset, by 3.5% in MSE and 4.8% in MAE on average. Against the best baseline per dataset, the reductions are 1.45% and 0.79% on HumanActivity, 6.11% and 5.40% on USHCN, and 1.28% and 2.15% on MIMIC. No auxiliary objective or additional supervision is used, so these gains arise from the proposed patch operator. ***2) The improvement varies across benchmarks as Proposition 1 predicts.*** The largest gain over APN appears on USHCN, reaching 9.36% in MSE and 10.07% in MAE: pronounced variation in recording density makes consecutive records represent unequal amounts of physical time, increasing the covariance term in Eq. equation 2. HumanActivity is the opposite regime, with dense and relatively uniform sampling, and accordingly exhibits smaller but consistent gains. ***3) PhysioNet exhibits a metric-dependent trade-off.*** VQPatch achieves the lowest MAE while its MSE remains within 1.7% of the best result, obtained by Warpformer. Clinical series contain sparse extreme excursions that can disproportionately affect squared error and may coincide with bursts of dense recording. VQPatch reduces the influence of such recording density during content formation, yielding stronger MAE while remaining competitive in MSE. Its variability across seeds is also small, e.g., an MSE standard deviation of 0.0004 on HumanActivity.

Table 1: Forecasting performance (MSE, MAE; mean±std) on four IMTS datasets. The best and second-best results are highlighted in bold and with an underline, respectively.

Dataset	HumanActivity		USHCN		PhysioNet		MIMIC	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
PrimeNet	4.2507±0.0041	1.7018±0.0011	0.4930±0.0015	0.4954±0.0018	0.7953±0.0000	0.6859±0.0001	0.9073±0.0001	0.6614±0.0001
NeuralFlows	0.1722±0.0090	0.3150±0.0094	0.2087±0.0258	0.3157±0.0187	0.4056±0.0033	0.4466±0.0027	0.6085±0.0101	0.5306±0.0066
CRU	0.1387±0.0073	0.2607±0.0092	0.2168±0.0162	0.3180±0.0248	0.6179±0.0045	0.5778±0.0031	0.5895±0.0092	0.5151±0.0048
mTAN	0.0993±0.0026	0.2219±0.0047	0.5561±0.2020	0.5015±0.0968	0.3809±0.0043	0.4291±0.0035	0.9408±0.1126	0.6755±0.0459
SeFT	1.3786±0.0024	0.9762±0.0007	0.3345±0.0022	0.4083±0.0084	0.7721±0.0021	0.6760±0.0029	0.9230±0.0015	0.6628±0.0008
GNeuralFlow	0.3936±0.1585	0.4541±0.0841	0.2205±0.0421	0.3286±0.0412	0.8207±0.0310	0.6759±0.0100	0.8957±0.0209	0.6450±0.0072
GRU-D	0.1893±0.0627	0.3253±0.0485	0.2097±0.0493	0.3045±0.0305	0.3419±0.0029	0.3992±0.0011	0.4759±0.0100	0.4526±0.0055
Raindrop	0.0916±0.0072	0.2114±0.0072	0.2035±0.0336	0.3029±0.0264	0.3478±0.0019	0.4044±0.0020	0.6754±0.1829	0.5444±0.0868
Warpformer	0.0449±0.0010	0.1228±0.0018	0.1888±0.0598	0.2939±0.0591	**0.3056±0.0011**	0.3661±0.0016	0.4302±0.0035	0.4025±0.0014
GraFITi	0.0437±0.0005	0.1221±0.0017	0.1691±0.0093	0.2771±0.0248	0.3075±0.0015	0.3637±0.0036	0.4359±0.0455	0.4142±0.0297
t-PatchGNN	0.0482±0.0018	0.1209±0.0054	0.1817±0.0023	0.2883±0.0126	0.3133±0.0051	0.3678±0.0024	0.4661±0.0109	0.3953±0.0081
HyperIMTS	0.0429±0.0025	0.1196±0.0004	0.1622±0.0047	0.2652±0.0072	0.3154±0.0131	0.3696±0.0055	0.4301±0.0020	0.4034±0.0066
VIMTS	0.0422±0.0074	0.1163±0.0086	0.1649±0.0107	0.2607±0.0153	0.3182±0.0030	0.3683±0.0004	0.4367±0.0011	0.4018±0.0009
KAFNet	0.0441±0.0012	0.1198±0.0007	0.1607±0.0099	0.2638±0.0182	0.3197±0.0043	0.3708±0.0017	0.4325±0.0084	0.4054±0.0159
TFMixer	0.0420±0.0009	0.1145±0.0027	0.1537±0.0226	0.2482±0.0055	0.3113±0.0046	0.3573±0.0020	0.4310±0.0103	0.3965±0.0072
ASTGI	0.0413±0.0032	0.1192±0.0051	0.1606±0.0121	0.2592±0.0204	0.3120±0.0033	0.3595±0.0038	0.4342±0.0234	0.3961±0.0105
APN	0.0419±0.0002	0.1159±0.0006	0.1592±0.0137	0.2611±0.0167	0.3127±0.0021	0.3665±0.0039	0.4292±0.0029	0.4016±0.0016
VQPatch (Ours)	**0.0407±0.0004**	**0.1136±0.0003**	**0.1443±0.0096**	**0.2348±0.0105**	0.3109±0.0009	**0.3546±0.0021**	**0.4237±0.0032**	**0.3868±0.0014**

Table 2: Ablation study on the HumanActivity and PhysioNet datasets. Values in blue parentheses denote the relative increase (%) compared with the full model. Lower values are better.

Variant	HumanActivity		PhysioNet	
	MSE ↓	MAE ↓	MSE ↓	MAE ↓
Full Model	**0.042702**	**0.115622**	**0.313920**	**0.367713**
w/o COD	0.042922 (+0.52%)	0.117631 (+1.74%)	0.314184 (+0.08%)	0.368241 (+0.14%)
w/o VTQ	0.042981 (+0.65%)	0.118121 (+2.16%)	0.314552 (+0.20%)	0.368825 (+0.30%)
w/o OPC	0.042927 (+0.53%)	0.116427 (+0.70%)	0.313952 (+0.01%)	0.368193 (+0.13%)
w/o COD & VTQ	0.043049 (+0.81%)	0.119219 (+3.11%)	0.314468 (+0.17%)	0.369878 (+0.59%)
w/o COD & OPC	0.043214 (+1.20%)	0.117944 (+2.01%)	0.314379 (+0.15%)	0.368659 (+0.26%)
w/o VTQ & OPC	0.043173 (+1.10%)	0.119732 (+3.55%)	0.317436 (+1.12%)	0.370228 (+0.68%)
w/o All	0.044841 (+5.01%)	0.121062 (+4.70%)	0.319905 (+1.91%)	0.382482 (+4.02%)

Insight. *Accuracy gains track the variability of recording density rather than model size. The physical-time and counting measures nearly agree under uniform recording, and they diverge precisely where the covariance term is large.*

Two controlled studies confirm this mechanism directly. Over 1,760 schedule-window cells the covariance term of Eq. 2 reproduces the measured bias with $R^2 = 0.9999$, and on HumanActivity VQPatch matches APN under the original observation schedule yet lowers MSE by 8.03% once the schedule is shifted (Appendix C and Appendix D).

5.3 Ablation Study

Table 2 isolates COD, VTQ, and OPC on HumanActivity and PhysioNet. All variants share one configuration and a fixed seed, so absolute values differ slightly from Table 1. Removing any single component degrades both metrics on both datasets. VTQ matters most, with MAE increases of 2.16% and 0.30%; this variant restores point-normalized mass while leaving the architecture intact, so it measures the effect of **Proposition** 1 directly. COD ranks second, consistent with keeping support statistics out of content formation, and OPC costs least in isolation, as expected for a bounded low-rank correction. The three components are complementary rather than additive. Removing VTQ and OPC together raises MAE by 3.55% on HumanActivity, above the sum of the individual effects of 2.16% and 0.70%, and by 0.68% on PhysioNet against 0.30% and 0.13%. Once physical-time mass is discarded, the bounded condition has no stable content context to modulate. Disabling all three recovers a point-normalized adaptive patch operator and is worst, with MAE increases of 4.70% and 4.02%.

5.4 Parameter Sensitivity

Fig. 3 varies four method-related hyperparameters: the quadrature power γ, the OPC residual scale η, the patch number P, and the time-encoding dimension d_t. VQPatch is stable across broad ranges

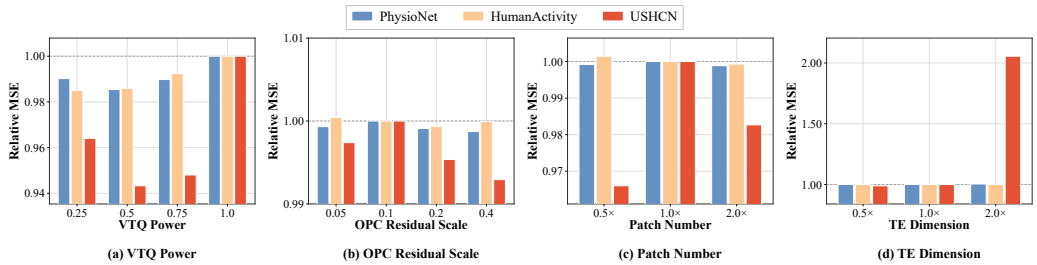


Figure 3: Sensitivity of VQPatch to its main method-related hyperparameters on PhysioNet, HumanActivity, and USHCN. MSE is normalized by the corresponding default configuration, where 1.0 denotes the default performance and lower values are better.

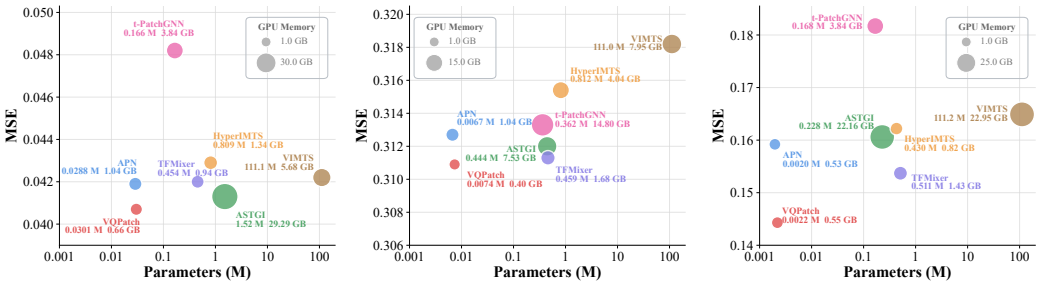


Figure 4: Accuracy–efficiency trade-offs on HumanActivity, PhysioNet, and USHCN. Bubble size indicates GPU memory; only representative baselines with resource measurements are shown.

of all four. Moderate values of γ achieve lower normalized MSE than the default $\gamma = 1$, which Eq. 6 explains as a bias-variance trade-off: exact quadrature removes the measure bias but grants the widest cells the largest influence, and those cells rest on a single distant record. Insensitivity to η and P indicates that neither the bounded conditioning path nor the support resolution requires precise tuning. An excessively large d_t degrades USHCN, where few variables and long spans favor a low-capacity temporal embedding. Appendix F.3 reports model and training hyperparameters.

5.5 Efficiency Analysis

Fig. 4 compares VQPatch with representative baselines in MSE, parameter count, and GPU memory. Among the methods shown, VQPatch achieves the lowest MSE on PhysioNet and USHCN and remains close to the best result on HumanActivity, using only 0.0301M, 0.0074M, and 0.0022M parameters with 0.66, 0.40, and 0.55 GB training memory, respectively. On PhysioNet, this is 61.4% lower than APN. Its compact patch representation avoids higher-order record-level interactions, while Voronoi weighting adds no learnable parameters. Appendix F.4 reports broader comparisons.

6 Conclusion

Irregular patching has been improved for years by sharpening the temporal region that a token reads; this work shows that the decisive choice lies in the quantity that the token averages. Treating the aggregation measure as a design variable turns the failure of point-normalized patching into a statement about estimands rather than architectures, since the gap to physical-time content is a patch-local covariance that no increase in capacity can absorb. VQPatch closes that gap with Voronoi cell widths as quadrature weights, while the observation pattern survives as a bounded condition. The correction adds no parameters and no asymptotic cost beyond sorting, yet it delivers the lowest MAE on four benchmarks against seventeen baselines, and controlled interventions confirm that the predicted covariance is the source of the gain. Two directions remain. Nearest-record weights make the approximation error scale with the widest cell, so higher-order or learned measures could sharpen it where records are sparse. The separation of support from measure is also not specific to forecasting, and it extends naturally to classification, imputation, and irregular event modeling.

AI USE STATEMENT

Generative AI tools were used solely for language editing and proofreading of this paper. The authors take full responsibility for the final content.

REPRODUCIBILITY STATEMENT

Sec. 3 states the theoretical results, with assumptions and proofs provided in Appendix B.2 and Appendix B.3. Sec. 4 specifies the model, complexity, and training objective, while Appendix E.1–E.6 details the datasets, baselines, training protocol, evaluation metrics, forecasting setup, and VQPatch configurations. Main results are reported over five seeds (2024–2028); ablation, sensitivity, and controlled studies use the fixed-seed settings specified in their respective sections. An anonymized implementation and configuration files accompany the submission.

REFERENCES

Marin Biloš, Johanna Sommer, 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.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *CoRR*, abs/1606.01865, 2016.

Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. ContiFormer: Continuous-time transformer for irregular time series modeling. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

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.

Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. In *Advances in Neural Information Processing Systems*, volume 32, pp. 7377–7388, 2019.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022.

Max Horn, Michael Moor, Christian Bock, Bastian Rieck, 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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 24805–24822, 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*, 3:160035, 2016.

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, pp. 35502–35518, 2025.

Jiexi Liu, Meng Cao, and Songcan Chen. TimeCHEAT: A channel harmony strategy for irregularly sampled multivariate time series analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18861–18869, 2025.

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 *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.

Matthew J. Menne, Claude N. Jr. Williams, and Russell S. Vose. Long-term daily and monthly climate records from stations across the contiguous united states (U.S. Historical Climatology Network). Technical report, Carbon Dioxide Information Analysis Center, 2016.

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.

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, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. TFB: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proceedings of the VLDB Endowment*, 17 (9):2363–2377, 2024.

Xiangfei Qiu, Zhe Li, Wanghui Qiu, Shiyan Hu, Lekui Zhou, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Aoying Zhou, Zhenli Sheng, Jilin Hu, Christian S. Jensen, and Bin Yang. Tab: Unified benchmarking of time series anomaly detection methods. *Proc. VLDB Endow.*, 18(9):2775–2789, 2025.

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 *ICML*, 2026.

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

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. 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. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021.

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. *Computing in Cardiology*, 39:245–248, 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.

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, 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.

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

Ziyu Zhou, Yiming Huang, Yanyun Wang, Yuankai Wu, James Kwok, and Yuxuan Liang. Revitalizing canonical pre-alignment for irregular multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29115–29123, 2026.

A NOTATION

Table 3 collects the symbols used throughout the paper, grouped by the stage of the model in which they appear. Unless stated otherwise, the batch index is omitted, indices n and p refer to a variable and a patch, index ℓ refers to a padded historical row, index i refers to a coalesced observation of a given variable, and index j refers to a forecast query.

Table 3: Summary of notation.

Symbol	Description
Problem setting	
$[t_{\mathrm{L}}, t_{\mathrm{R}}]$	historical observation window
B, L, N, Q	batch size, padded historical row length, number of variables, number of forecast queries
$\mathbf{X}$, $\mathbf{T}$, $\mathbf{M}$	value tensor, timestamp tensor, observation mask
$x_{\ell n}$, t_ℓ, $m_{\ell n}$	value of variable n, row timestamp, and observation mask entry
$\mathbf{Q}$, q_j	query timestamps and the j-th query timestamp
$\mathbf{Y}$, y_{jn}, m^y_{jn}	prediction targets, target value, and target validity mask
f_θ, $\widehat{y}_{jn}$	forecasting model and predicted value
Record representation and COD	
$\mathbf{e}_t(\cdot)$, d_t	learnable time embedding and its dimension
$\mathbf{z}_{\ell n}$, d_z	record token $x_{\ell n} \oplus \mathbf{e}_t(t_\ell)$ and its dimension $d_z = 1 + d_t$
$\mathcal{C}$, $\mathcal{S}$	content path and support path defined by COD
Temporal Patch Operator and VTQ	
$\mathcal{P}_{np}(\mu; \mathbf{z}_n)$	Temporal Patch Operator under aggregation measure μ
$\mathbf{z}_n(t)$, $\lambda_n(t)$	temporal content process and observation intensity of variable n
μ_n^{obs}, ν_{np}	observation-induced measure and support-normalized physical-time measure
K_{np}, l_{np}, r_{np}, τ_n	Support Kernel, learnable boundaries, and boundary softness
σ, P, ε	logistic function, number of patches, numerical-stability constant
$\mathbf{h}_{np}^{\mathrm{obs}}$	point-normalized patch aggregate
$\widetilde{t}_{in}$, $\widetilde{x}_{in}$, $\widetilde{\mathbf{z}}_{in}$, $\widetilde{L}_n$	coalesced timestamps, values, record tokens, and their number
β_{in}, C_{in}, d_{in}, $\bar{d}_{in}$, h_n	cell boundary, Voronoi cell, cell width, normalized width, and cell mesh
γ, $\mathbf{u}_{np}$	quadrature power and Physical-Time Content Token
ϕ, PE, $\bar{\mathbf{h}}_{np}$	patch projection with residual refinement, patch-position encoding, encoded patch token
$\boldsymbol{\xi}_n$, α_{np}, $\mathbf{h}_n$, D	variable query, patch weight, variable-level content context, hidden dimension
OPC and decoding	
$\mathcal{T}_j^{\mathrm{seq}}$, Δ_j	ordered historical timestamp rows up to q_j and query recency
κ_j, κ, s_j, g_j, c_j	query-local and total valid counts, temporal span, maximum adjacent gap, normalized coverage
$\boldsymbol{\rho}_j^{\mathrm{raw}}$, $\bar{\boldsymbol{\rho}}_j$	Observation Support Descriptor and its standardized form
$\mathcal{I}_t$, $\mathcal{I}_s$, E_t, E_s	temporal-geometry and support-count index sets with their encoders
$\mathbf{e}_j^t$, $\mathbf{e}_j^{\mathrm{s}}$, R	descriptor embeddings and the low-rank conditioning dimension
$\boldsymbol{\delta}_j$, η	bounded observation condition and residual scale
V, U, $\mathbf{v}_n$, $\widetilde{\mathbf{h}}_{nj}$	down and up projections, projected content, conditioned content state
$\mathbf{s}_{nj}$, f_{dec}	decoder input and shared predictor

B DETAILED THEORETICAL DERIVATIONS

This appendix provides the derivations omitted from the main-text theoretical analysis. We first show that point-normalized aggregation is exactly an integral under an empirical observation measure and characterize its continuous observation-induced limit. We then prove the Observation-Measure Bias Identity and establish the Voronoi quadrature approximation and consistency result for normalized physical-time patch content.

B.1 From Point-Normalized Aggregation to the Observation-Induced Measure

Fix a variable n and a patch p over the historical window $[t_{\mathrm{L}}, t_{\mathrm{R}}]$. Let $\{(t_{in}, \mathbf{z}_{in})\}_{i=1}^{L_n}$ denote the valid observations of variable n, where L_n is the number of valid observations. We analyze the unstabilized point-normalized aggregation

$$\mathbf{h}_{np,L_n}^{\mathrm{point}} = \frac{\sum_{i=1}^{L_n} K_{np}(t_{in})\mathbf{z}_{in}}{\sum_{i=1}^{L_n} K_{np}(t_{in})}, \tag{B.1}$$

whenever its denominator is positive. Each valid observation carries the same base point mass before normalization, while $K_{np}(t_{in})$ determines the extent to which that mass contributes to patch p.

Define the empirical observation measure of variable n by

$$\mu_{n,L_n} = \frac{1}{L_n}\sum_{i=1}^{L_n} \delta_{t_{in}}, \tag{B.2}$$

where $\delta_{t_{in}}$ is the Dirac measure at t_{in}. Identifying $\mathbf{z}_{in} = \mathbf{z}_n(t_{in})$, integration against μ_{n,L_n} gives

$$\int K_{np}(t)\mathbf{z}_n(t)\, d\mu_{n,L_n}(t) = \frac{1}{L_n}\sum_{i=1}^{L_n} K_{np}(t_{in})\mathbf{z}_{in}, \tag{B.3}$$

and

$$\int K_{np}(t)\, d\mu_{n,L_n}(t) = \frac{1}{L_n}\sum_{i=1}^{L_n} K_{np}(t_{in}). \tag{B.4}$$

The factors $1/L_n$ cancel in the normalized ratio. Hence

$$\mathbf{h}_{np,L_n}^{\mathrm{point}} = \frac{\int K_{np}(t)\mathbf{z}_n(t)\, d\mu_{n,L_n}(t)}{\int K_{np}(t)\, d\mu_{n,L_n}(t)} = \mathcal{P}_{np}(\mu_{n,L_n}; \mathbf{z}_n). \tag{B.5}$$

Equation (B.5) is an exact rewriting of the finite point-normalized sum, not an approximation.

To relate this finite measure to its continuous counterpart, suppose that $\mu_{n,L_n} \Rightarrow \mu_n^{\mathrm{obs}}$ as $L_n \to \infty$. Assume that the numerator and denominator integrands satisfy conditions under which their empirical integrals converge—for example, bounded continuity on the compact historical window—and that the limiting denominator is positive. Then the continuous limit of Eq. (B.5) is $\mathcal{P}_{np}(\mu_n^{\mathrm{obs}}; \mathbf{z}_n)$. If μ_n^{obs} is absolutely continuous with respect to physical time and its density is proportional to a nonnegative local observation intensity g_n, then

$$d\mu_n^{\mathrm{obs}}(t) = \frac{g_n(t)}{Z_{g,n}}\, dt, \qquad Z_{g,n} = \int_{t_{\mathrm{L}}}^{t_{\mathrm{R}}} g_n(s)\, ds. \tag{B.6}$$

Here $0 < Z_{g,n} < \infty$.

Substituting Eq. (B.6) into the Temporal Patch Operator yields

$$\begin{aligned} \mathcal{P}_{np}(\mu_n^{\mathrm{obs}}; \mathbf{z}_n) &= \frac{\int K_{np}(t)\mathbf{z}_n(t)g_n(t)\, dt/Z_{g,n}}{\int K_{np}(t)g_n(t)\, dt/Z_{g,n}} \\ &= \frac{\int K_{np}(t)\mathbf{z}_n(t)g_n(t)\, dt}{\int K_{np}(t)g_n(t)\, dt}, \end{aligned} \tag{B.7}$$

where all unmarked integrals in this appendix are over $[t_{\mathrm{L}}, t_{\mathrm{R}}]$. By contrast, physical-time aggregation is

$$\mathcal{P}_{np}(dt; \mathbf{z}_n) = \frac{\int K_{np}(t)\mathbf{z}_n(t)\, dt}{\int K_{np}(t)\, dt}. \tag{B.8}$$

Equations (B.7) and (B.8) use the same Support Kernel. Their distinction is the aggregation measure: μ_n^{obs} distributes mass according to observation intensity, whereas dt distributes mass according to physical time. Thus, the Support Kernel determines where a patch aggregates, while the aggregation measure determines how mass is distributed within that support.

B.2 Proof of the Observation-Measure Bias Identity

Assume that $K_{np}(t) \geq 0$ and

$$0 < \int K_{np}(t)\, dt < \infty. \tag{B.9}$$

Assume further that the relevant functions are integrable and that the patch-local mean observation intensity defined below is strictly positive. For vector-valued $\mathbf{z}_n$, all expectations and covariances are understood component-wise.

Define the support-normalized physical-time probability measure ν_{np} by

$$d\nu_{np}(t) = \frac{K_{np}(t)\, dt}{\int K_{np}(s)\, ds}. \tag{B.10}$$

For any integrable function f,

$$\mathbb{E}_{\nu_{np}}[f] = \frac{\int K_{np}(t) f(t)\, dt}{\int K_{np}(t)\, dt}. \tag{B.11}$$

It follows directly that

$$\mathcal{P}_{np}(dt; \mathbf{z}_n) = \mathbb{E}_{\nu_{np}}[\mathbf{z}_n]. \tag{B.12}$$

Using Eq. (B.7) and dividing both its numerator and denominator by $\int K_{np}(t)\, dt$ gives

$$\begin{aligned} \mathcal{P}_{np}(\mu_n^{\mathrm{obs}}; \mathbf{z}_n) &= \frac{\dfrac{\int K_{np}(t) \mathbf{z}_n(t) g_n(t)\, dt}{\int K_{np}(t)\, dt}}{\dfrac{\int K_{np}(t) g_n(t)\, dt}{\int K_{np}(t)\, dt}} \\ &= \frac{\mathbb{E}_{\nu_{np}}[\mathbf{z}_n g_n]}{\mathbb{E}_{\nu_{np}}[g_n]}. \end{aligned} \tag{B.13}$$

The covariance under ν_{np} is

$$\mathrm{Cov}_{\nu_{np}}(\mathbf{z}_n, g_n) = \mathbb{E}_{\nu_{np}}[\mathbf{z}_n g_n] - \mathbb{E}_{\nu_{np}}[\mathbf{z}_n]\mathbb{E}_{\nu_{np}}[g_n]. \tag{B.14}$$

Therefore,

$$\mathbb{E}_{\nu_{np}}[\mathbf{z}_n g_n] = \mathrm{Cov}_{\nu_{np}}(\mathbf{z}_n, g_n) + \mathbb{E}_{\nu_{np}}[\mathbf{z}_n]\mathbb{E}_{\nu_{np}}[g_n]. \tag{B.15}$$

Substitution into Eq. (B.13), using $\mathbb{E}_{\nu_{np}}[g_n] > 0$, yields

$$\begin{aligned} \mathcal{P}_{np}(\mu_n^{\mathrm{obs}}; \mathbf{z}_n) &= \frac{\mathrm{Cov}_{\nu_{np}}(\mathbf{z}_n, g_n) + \mathbb{E}_{\nu_{np}}[\mathbf{z}_n]\mathbb{E}_{\nu_{np}}[g_n]}{\mathbb{E}_{\nu_{np}}[g_n]} \\ &= \frac{\mathrm{Cov}_{\nu_{np}}(\mathbf{z}_n, g_n)}{\mathbb{E}_{\nu_{np}}[g_n]} + \mathbb{E}_{\nu_{np}}[\mathbf{z}_n] \\ &= \frac{\mathrm{Cov}_{\nu_{np}}(\mathbf{z}_n, g_n)}{\mathbb{E}_{\nu_{np}}[g_n]} + \mathcal{P}_{np}(dt; \mathbf{z}_n). \end{aligned} \tag{B.16}$$

Rearranging proves Proposition 1:

$$\mathcal{P}_{np}(\mu_n^{\mathrm{obs}}; \mathbf{z}_n) - \mathcal{P}_{np}(dt; \mathbf{z}_n) = \frac{\mathrm{Cov}_{\nu_{np}}(\mathbf{z}_n, g_n)}{\mathbb{E}_{\nu_{np}}[g_n]}. \tag{B.17}$$

Remark. Equation (B.17) is an exact identity under the stated conditions, rather than a first-order approximation. The deviation is determined by the patch-local dependence between temporal content and observation intensity, not by irregularity alone; in particular, the two aggregations coincide when $\mathrm{Cov}_{\nu_{np}}(\mathbf{z}_n, g_n) = 0$. Adapting K_{np} changes ν_{np} and hence the covariance, but learning patch boundaries alone does not generally force this covariance to vanish.

B.3 Voronoi Quadrature Approximation and Consistency

After coalescing exact duplicate timestamps for variable n, let

$$t_{\mathrm{L}} \le \widetilde{t}_{1n} < \cdots < \widetilde{t}_{\widetilde{L}_n n} \le t_{\mathrm{R}} \tag{B.18}$$

be the ordered timestamps, with corresponding representations $\widetilde{\mathbf{z}}_{in}$. Define the boundary-clipped midpoint boundaries

$$\beta_{0n} = t_{\mathrm{L}}, \qquad \beta_{in} = \frac{\widetilde{t}_{in} + \widetilde{t}_{i+1,n}}{2} \quad (i = 1, \ldots, \widetilde{L}_n - 1), \qquad \beta_{\widetilde{L}_n n} = t_{\mathrm{R}}. \tag{B.19}$$

The temporal cell and its width are

$$C_{in} = [\beta_{i-1,n}, \beta_{in}], \qquad d_{in} = |C_{in}| = \beta_{in} - \beta_{i-1,n}. \tag{B.20}$$

Up to shared endpoints of measure zero, the cells form a partition of the historical window. Consequently,

$$\sum_{i=1}^{\widetilde{L}_n} d_{in} = t_{\mathrm{R}} - t_{\mathrm{L}}. \tag{B.21}$$

Thus, d_{in} is the local physical-time extent represented by observation i. It is neither an attention weight, a confidence score, nor a learned importance score.

For an integrable scalar- or vector-valued function f,

$$\int_{t_{\mathrm{L}}}^{t_{\mathrm{R}}} f(t)\, dt = \sum_{i=1}^{\widetilde{L}_n} \int_{C_{in}} f(t)\, dt. \tag{B.22}$$

Approximating f within each cell by its value at the representative timestamp gives

$$\int_{C_{in}} f(t)\, dt \approx d_{in} f(\widetilde{t}_{in}), \qquad \int_{t_{\mathrm{L}}}^{t_{\mathrm{R}}} f(t)\, dt \approx \sum_{i=1}^{\widetilde{L}_n} d_{in} f(\widetilde{t}_{in}). \tag{B.23}$$

Applying Eq. (B.23) to $f_{np,z}(t) = K_{np}(t)\mathbf{z}_n(t)$ and $f_{np}(t) = K_{np}(t)$ gives

$$\int K_{np}(t)\mathbf{z}_n(t)\, dt \approx \sum_{i=1}^{\widetilde{L}_n} K_{np}(\widetilde{t}_{in}) d_{in} \widetilde{\mathbf{z}}_{in}, \tag{B.24}$$

and

$$\int K_{np}(t)\, dt \approx \sum_{i=1}^{\widetilde{L}_n} K_{np}(\widetilde{t}_{in}) d_{in}. \tag{B.25}$$

The resulting unstabilized quadrature token is

$$\mathbf{u}_{np}^{(0)} = \frac{\sum_{i=1}^{\widetilde{L}_n} K_{np}(\widetilde{t}_{in}) d_{in} \widetilde{\mathbf{z}}_{in}}{\sum_{i=1}^{\widetilde{L}_n} K_{np}(\widetilde{t}_{in}) d_{in}}. \tag{B.26}$$

Here, $K_{np}(\widetilde{t}_{in})$ specifies the temporal relevance of observation i to patch p, whereas d_{in} supplies its physical-time quadrature mass. Their product therefore follows from discretizing the dt-based numerator and denominator, rather than from introducing an additional learned attention weight.

To establish consistency, define the mesh radius

$$\delta_{n,L} = \max_{1 \le i \le \widetilde{L}_n} \sup_{t \in C_{in}} |t - \widetilde{t}_{in}|. \tag{B.27}$$

Suppose that f is Lipschitz on $[t_{\mathrm{L}}, t_{\mathrm{R}}]$ with constant $\mathrm{Lip}(f)$. For vector-valued f, the Lipschitz condition is taken with respect to the chosen norm. For every $t \in C_{in}$,

$$\|f(t) - f(\widetilde{t}_{in})\| \le \mathrm{Lip}(f)|t - \widetilde{t}_{in}| \le \mathrm{Lip}(f)\delta_{n,L}. \tag{B.28}$$

Therefore,

$$\begin{aligned}\left\| \int f(t)\,dt - \sum_{i=1}^{\widetilde{L}_n} d_{in} f(\widetilde{t}_{in}) \right\| &= \left\| \sum_{i=1}^{\widetilde{L}_n} \int_{C_{in}} \left[f(t) - f(\widetilde{t}_{in}) \right] dt \right\| \\ &\le \sum_{i=1}^{\widetilde{L}_n} \int_{C_{in}} \| f(t) - f(\widetilde{t}_{in}) \|\, dt \\ &\le \mathrm{Lip}(f)\delta_{n,L} \sum_{i=1}^{\widetilde{L}_n} |C_{in}| \\ &= \mathrm{Lip}(f)(t_{\mathrm{R}} - t_{\mathrm{L}})\delta_{n,L}.\end{aligned} \tag{B.29}$$

Hence the Voronoi quadrature converges to the corresponding integral as $\delta_{n,L} \to 0$.

It remains to control the normalized ratio. Define

$$\mathbf{A} = \int K_{np}(t)\mathbf{z}_n(t)\,dt, \qquad B = \int K_{np}(t)\,dt, \tag{B.30}$$

and

$$\widehat{\mathbf{A}} = \sum_{i=1}^{\widetilde{L}_n} K_{np}(\widetilde{t}_{in}) d_{in} \widetilde{\mathbf{z}}_{in}, \qquad \widehat{B} = \sum_{i=1}^{\widetilde{L}_n} K_{np}(\widetilde{t}_{in}) d_{in}. \tag{B.31}$$

For positive B and $\widehat{B}$,

$$\frac{\widehat{\mathbf{A}}}{\widehat{B}} - \frac{\mathbf{A}}{B} = \frac{\widehat{\mathbf{A}} - \mathbf{A}}{\widehat{B}} + \mathbf{A}\left(\frac{1}{\widehat{B}} - \frac{1}{B} \right), \tag{B.32}$$

so

$$\left\| \frac{\widehat{\mathbf{A}}}{\widehat{B}} - \frac{\mathbf{A}}{B} \right\| \le \frac{\|\widehat{\mathbf{A}} - \mathbf{A}\|}{\widehat{B}} + \frac{\|\mathbf{A}\|\,|\widehat{B} - B|}{B\widehat{B}}. \tag{B.33}$$

Assume that $B \ge b_0 > 0$, $\widehat{B} \ge b_0/2$, and $\|\mathbf{z}_n(t)\| \le M$ throughout the historical window. Also assume that K_{np} and $K_{np}\mathbf{z}_n$ are Lipschitz. Since $K_{np} \ge 0$,

$$\|\mathbf{A}\| \le \int K_{np}(t)\|\mathbf{z}_n(t)\|\,dt \le MB. \tag{B.34}$$

Applying Eq. (B.29) to the numerator and denominator gives

$$\|\widehat{\mathbf{A}} - \mathbf{A}\| \le (t_{\mathrm{R}} - t_{\mathrm{L}})\,\mathrm{Lip}(K_{np}\mathbf{z}_n)\delta_{n,L}, \tag{B.35}$$

and

$$|\widehat{B} - B| \le (t_{\mathrm{R}} - t_{\mathrm{L}})\,\mathrm{Lip}(K_{np})\delta_{n,L}. \tag{B.36}$$

Combining Eqs. (B.33)–(B.36) yields

$$\left\| \frac{\widehat{\mathbf{A}}}{\widehat{B}} - \frac{\mathbf{A}}{B} \right\| \le \frac{2(t_{\mathrm{R}} - t_{\mathrm{L}})\delta_{n,L}}{b_0} \left[\mathrm{Lip}(K_{np}\mathbf{z}_n) + M\,\mathrm{Lip}(K_{np}) \right]. \tag{B.37}$$

Consequently, under the stated regularity and positive-denominator conditions,

$$\delta_{n,L} \to 0 \quad \Longrightarrow \quad \mathbf{u}_{np}^{(0)} \to \mathcal{P}_{np}(dt; \mathbf{z}_n). \tag{B.38}$$

The implemented token uses the stabilized denominator $\widehat{B}+\varepsilon$. The consistency statement above concerns the unstabilized ratio. Under the same bounded-content and positive-denominator conditions, stabilization changes the discrete token by at most $2M\varepsilon/b_0$; exact convergence to the normalized physical-time operator therefore holds for $\varepsilon = 0$ or when $\varepsilon \to 0$. This is a numerical quadrature consistency result. It does not assert unbiased recovery of an underlying latent continuous trajectory.

C Empirical Verification of Observation Measure Substitution

Purpose. The theoretical analysis characterizes the aggregation bias induced by Observation Measure Substitution through a patch-local covariance term between the Temporal Content Process and observation intensity. This controlled experiment tests whether the covariance-based theoretical bias accurately matches the bias obtained from direct aggregation. It also examines whether the physical-time aggregation used in VQPatch more closely approximates the physical-time target.

Table 4: Controlled verification of Observation Measure Substitution across 1,760 schedule-window cells.

Metric	Result
Number of schedule-window cells	1,760
Pearson(actual bias, theory bias)	0.999973876
Spearman(actual bias, theory bias)	0.999781418
R^2	0.999938904
Bias direction accuracy	0.998295455
Fraction of cells where physical-time aggregation has lower TME	0.996590909
Physical-time aggregation/APN TME ratio under the uniform schedule	0.999998768

Setup. We construct Temporal Content Processes and Observation Processes with different local intensity profiles, and vary both the observation schedule and the patch window. This produces 1,760 cross-schedule-window cells. For each cell, we compare the covariance-based theoretical bias with the actual aggregation bias using Pearson correlation, Spearman correlation, R^2, and bias-direction accuracy. We further compare the Time-Measure Error (TME) of APN aggregation and the physical-time aggregation used in VQPatch, with a uniform observation schedule serving as the control.

Results. Table 4 reports the complete results.

Pearson correlation, Spearman correlation, and R^2 are all close to one, while the bias directions agree in 99.8295% of the evaluated cells. The physical-time aggregation used in VQPatch obtains a lower TME than APN aggregation in 99.6591% of the cells. Under the uniform-schedule control, the physical-time aggregation/APN TME ratio is 0.999998768, showing nearly identical TME for the two aggregation schemes. We next evaluate whether this aggregation-level correction translates into improved forecasting behavior under controlled schedule shifts.

D Controlled Schedule-Shift Evaluation of VQPatch

Purpose. We evaluate whether VQPatch matches APN under the Original Schedule while reducing forecasting error and prediction sensitivity under two controlled observation-schedule interventions.

Setup. We evaluate APN and VQPatch on HumanActivity using seed 2024, with 218 paired samples for each schedule condition. Within each pair, the prediction target, target mask, sample identity, and evaluator are fixed; only the historical observation schedule is modified. *Original Schedule* applies no additional intervention. *Right-side Fixed-count Burst* concentrates a fixed number of historical observations in a local region on the right side of the historical window, close to the forecast query. *Value-dependent Thinning* changes the retention probability according to the observed historical value, creating an Observation Process associated with the Temporal Content Process.

We report mean squared error (MSE) for forecasting accuracy. The Forecast Sensitivity Score (FSS) measures the relative prediction change caused by a schedule intervention:

$$\mathrm{FSS} = \mathrm{median}_j \frac{\|\widehat{y}_{j,\mathrm{stress}} - \widehat{y}_{j,\mathrm{original}}\|}{\|\widehat{y}_{j,\mathrm{original}}\| + \varepsilon}. \tag{10}$$

The Schedule Degradation Ratio (SDR) measures the corresponding sample-level error change relative to the Original Schedule:

$$\mathrm{SDR} = \mathrm{median}_j \frac{\mathrm{MSE}_{j,\mathrm{stress}} - \mathrm{MSE}_{j,\mathrm{original}}}{\mathrm{MSE}_{j,\mathrm{original}} + \varepsilon}. \tag{11}$$

Lower FSS and SDR indicate lower schedule sensitivity and error degradation, respectively. A negative SDR indicates that the median sample-level error change under the intervention is below the Original Schedule reference. By construction, FSS and SDR are zero for the Original Schedule.

Table 5: Controlled schedule-shift results on HumanActivity using seed 2024 and 218 paired samples per condition.

Condition	Metric	APN	VQPatch	Relative Change (VQPatch vs. APN)
Original Schedule	MSE	0.042091	**0.04208789**	-0.0074%
Right-side Fixed-count Burst	MSE	0.045849	**0.04216640**	-8.03%
	FSS	0.014729	**0.00375610**	-74.5%
	SDR	0.013326	-0.00273638	–
Value-dependent Thinning	MSE	0.045704	**0.04441045**	-2.83%
	FSS	0.017386	**0.01581691**	-9.03%
	SDR	0.040840	**0.02576066**	-36.92%

Results. Table 5 reports the complete paired evaluation.

Under the Original Schedule, APN and VQPatch obtain MSE values of 0.042091 and 0.04208789, respectively, a relative change of -0.007%. VQPatch matches APN's forecasting performance under the Original Schedule.

Under the Right-side Fixed-count Burst, VQPatch reduces MSE from 0.045849 to 0.04216640, a reduction of 8.03%, and FSS from 0.014729 to 0.00375610, a reduction of 74.5%. SDR decreases from 0.013326 to -0.00273638.

Under Value-dependent Thinning, VQPatch reduces MSE from 0.045704 to 0.04441045 (2.83%), FSS from 0.017386 to 0.01581691 (9.03%), and SDR from 0.040840 to 0.02576066 (36.92%).

Within the evaluated HumanActivity setting, VQPatch matches APN under the Original Schedule and reduces forecasting error and schedule sensitivity under both evaluated interventions. Together with the aggregation-level verification in Section C, these results connect Observation Measure Substitution to its downstream mitigation by VQPatch under the two controlled schedule interventions.

E Additional Experimental Details

This appendix provides additional details on the datasets, baseline methods, implementation protocol, and additional experimental analyzes used in our evaluation.

E.1 Dataset Details

We evaluate VQPatch on four widely used IMTS forecasting benchmarks: MIMIC, PhysioNet, HumanActivity, and USHCN. These datasets span healthcare, biomechanics, and climate science, and exhibit substantially different temporal scales, sampling densities, and degrees of irregularity. Table 6 summarizes their quantitative statistics.

Table 6: Statistical overview of the four benchmark datasets used in our experiments.

Property	**MIMIC**	**PhysioNet**	**HumanActivity**	**USHCN**
# Variables	96	36	12	5
# Samples	21,250	11,981	1,359	1,114
Avg # Observations	144.6	308.6	362.2	313.5
Max Sequence Length	96	47	131	337

MIMIC. MIMIC (Johnson et al., 2016) is a large-scale critical-care database containing de-identified clinical measurements collected from patients admitted to intensive care units. We use the commonly adopted irregular forecasting benchmark constructed from measurements within the first

48 hours after ICU admission. The dataset is characterized by high dimensionality, asynchronous sampling, and substantial missingness across clinical variables.

PhysioNet. PhysioNet (Silva et al., 2012) is derived from the PhysioNet 2012 Challenge and contains physiological measurements collected during the first 48 hours of ICU stays. Different clinical variables are recorded at non-uniform timestamps, resulting in highly irregular and partially observed multivariate sequences. This dataset therefore provides a representative setting for evaluating forecasting models under sparse clinical observations.

HumanActivity. HumanActivity (Rubanova et al., 2019) contains three-dimensional positional measurements collected from subjects performing different physical activities. The measurements are recorded at irregular timestamps and exhibit rapid local changes at a millisecond-level temporal scale. Compared with the two clinical datasets, HumanActivity contains substantially faster temporal dynamics and therefore tests the model's ability to preserve fine-grained local information.

USHCN. USHCN (Menne et al., 2016) contains historical meteorological measurements collected from weather stations distributed across the United States. The observations cover long temporal spans and are irregularly recorded across multiple climate variables. Compared with the other benchmarks, USHCN provides a relatively low-dimensional but long-range forecasting scenario. Following previous IMTS forecasting studies, all four datasets are split into training, validation, and test sets with a ratio of 80%, 10%, and 10%.

E.2 Baseline Details

We compare VQPatch with seventeen representative baselines. As described in the main paper, they cover both models originally developed for irregular representation, classification, or imputation, and models designed for IMTS forecasting. For methods that were not originally proposed for forecasting, we adapt the prediction head to the forecasting task while preserving their core temporal representation mechanism.

Irregular representation, classification, and imputation models.

PrimeNet (Chowdhury et al., 2023) is a pre-training framework for IMTS. It learns transferable representations from irregular observations and adapts the learned representations to downstream prediction tasks.

SeFT (Horn et al., 2020) represents irregular observations as a set of time-value-variable tuples and employs set-based aggregation to obtain fixed-dimensional representations without explicitly aligning observations onto a common temporal grid.

mTAN (Shukla & Marlin, 2021) employs multi-time attention to map irregular observations to a set of reference time points, enabling the model to handle variable-length and asynchronously sampled sequences.

GRU-D (Che et al., 2016) extends the standard GRU with learnable temporal decay mechanisms. The decay terms explicitly incorporate elapsed time and missingness when updating input values and hidden states.

Raindrop (Zhang et al., 2022) models multivariate observations with graph-based message passing and temporal attention, allowing time-varying dependencies among irregularly observed variables to be captured.

Warpformer (Zhang et al., 2023) introduces a temporal warping mechanism for irregular time-series representation and combines multi-scale temporal alignment with Transformer-based sequence modeling.

IMTS forecasting models.

NeuralFlows (Biloš et al., 2021) models continuous-time temporal dynamics through neural flow transformations, providing an efficient alternative to ODE-based latent-state evolution for irregular sequences.

CRU (Schirmer et al., 2022) employs continuous recurrent units to evolve latent states between irregular observations and update them when new measurements become available.

GNeuralFlow (Mercatali et al., 2024) augments continuous neural-flow modeling with graph neural networks, allowing temporal evolution and cross-variable dependencies to be learned jointly.

t-PatchGNN (Zhang et al., 2024) divides each univariate irregular sequence into transformable temporal patches and combines Transformer-based intra-series modeling with time-adaptive graph neural networks for inter-series dependency learning.

GraFITi (Yalavarthi et al., 2024) formulates irregular multivariate forecasting as a graph-learning problem and performs prediction through message passing over sparse graph representations constructed from irregular observations and forecasting queries.

VIMTS (Hu et al., 2025) is a dedicated IMTS forecasting method designed to model irregular observations while preserving temporal information for future-value prediction.

TFMixer (Qiu et al., 2026) adopts a lightweight feature-mixing architecture for irregular multivariate forecasting, aiming to efficiently capture temporal and cross-variable interactions.

KAFNet (Zhou et al., 2026) revisits structured pre-alignment for IMTS forecasting and incorporates frequency-aware modeling to capture temporal patterns from irregular observations.

HyperIMTS (Li et al., 2025) represents observed values as nodes in a hypergraph and connects them through temporal and variable hyperedges. Its irregularity-aware message passing jointly captures temporal and cross-variable dependencies without requiring dense temporal alignment.

ASTGI (Liu et al., 2026a) directly represents each observation as a point in a learnable spatio-temporal space. It adaptively constructs a causal neighborhood graph for each observation and employs relation-aware propagation to model dynamic dependencies across time and variables.

APN (Liu et al., 2026b) uses time-aware adaptive patch aggregation to transform irregular observations into regularized patch representations, followed by a lightweight query-based aggregation module for forecasting.

E.3 Implementation Protocol

Unless otherwise stated, all experiments follow the training and model-selection settings described in the main paper. For baseline methods, configurations are initialized from those reported in the corresponding papers or official implementations, and key dataset-dependent hyperparameters are selected on the validation set. VQPatch follows the same selection procedure.

For the main comparison, each configuration is independently trained with five random seeds from 2024 to 2028, and the mean and standard deviation are reported. We do not use the "Drop Last" trick (Qiu et al., 2024; 2025). The complete model configurations used in our experiments are provided in the released code.

For the ablation study, all variants share the same training and evaluation protocol as the full model, with only the specified component or combination of components removed. The ablation variants use a fixed configuration and a single random seed, allowing the effect of architectural changes to be examined under controlled conditions; consequently, their absolute values may differ slightly from the tuned five-seed results reported in the main comparison.

For hyperparameter sensitivity analysis, one parameter is varied at a time while all remaining settings are fixed to the corresponding default configuration. These experiments also use a fixed seed and are intended to characterize sensitivity trends rather than establish statistically optimal hyperparameter values.

E.4 Evaluation Metrics

We evaluate forecasting accuracy with MSE and MAE, computed in standard form over the target entries selected by the target mask, so that padded and unobserved positions are excluded from both the numerator and the denominator. Lower values indicate better accuracy. The two metrics are complementary in this setting: MSE is dominated by the small number of extreme excursions that

appear in clinical series, whereas MAE reflects the typical-case error over the bulk of the targets. We therefore report both, and all results are given as mean $\pm$ standard deviation over five runs.

E.5 Forecasting Protocol

Following the standard IMTS forecasting setting, each sample is divided into a historical observation window and a future forecasting window. The model receives a collection of irregular historical observations together with forecasting queries and predicts the corresponding future values.

For HumanActivity, we use 3000 milliseconds of historical observations and a 300-millisecond forecasting window. For PhysioNet, the lookback window is 36 hours and the next three target timestamps are used for prediction. For USHCN, the historical input contains 150 observation steps and the following three target timestamps are predicted. For MIMIC, we follow the same preprocessing and forecasting protocol used in the main benchmark experiments.

E.6 VQPatch Hyperparameter Configuration

The final VQPatch configurations are selected on the validation set. The principal architecture and optimization hyperparameters include the number of patches, model dimension, time-encoding dimension, learning rate, batch size, dropout rate, quadrature power γ, and OPC residual scale η. Dataset-specific settings are used where appropriate. The complete configurations required to reproduce all experiments are provided in the released code.

F Additional Experimental Results

F.1 Complete Ablation Study

Table 2 in the main paper contains the complete combinatorial ablation over COD, VTQ, and OPC, covering every single-component removal, all three pairwise removals, and the variant in which all three components are disabled. We summarize the structure of those results here.

The ordering of single-component effects is stable across both datasets, with VTQ the most influential, COD second, and OPC third. The ordering matches the roles established in Sec. 3: VTQ changes the quantity estimated by patch content, COD protects that quantity from support statistics, and OPC recovers predictive support information as a bounded correction. The variant without all three components is the point-normalized adaptive patch operator, so the gap between it and the full model measures the total benefit of the measure-level redesign.

The pairwise results are the most informative for design. Combinations that remove VTQ are consistently worse than combinations that retain it, and removing VTQ together with OPC is worse than the sum of the two individual effects on both datasets. Support conditioning is therefore useful only after content mass has been placed on a physical-time footing, which is the reason VQPatch applies OPC after aggregation rather than inside it.

F.2 Hyperparameter Sensitivity

The main paper studies four method-related hyperparameters: the quadrature power γ, the OPC residual scale η, the number of patches P, and the time-encoding dimension d_t. For sensitivity analysis, one hyperparameter is varied at a time while all remaining settings are fixed to their corresponding default configurations. Since the absolute hyperparameter settings may differ across datasets, the patch number and time-encoding dimension are reported relative to their dataset-specific defaults.

The results in the main paper show that VQPatch is generally robust to these method-related configurations. Moderate values of the quadrature power achieve competitive or improved performance across datasets, while the OPC residual scale and patch number produce relatively small variations over the evaluated ranges. The time-encoding dimension is also stable on PhysioNet and HumanActivity, although an excessively large value leads to a performance degradation on USHCN. Overall, the results suggest that VQPatch does not rely on narrowly tuned method-specific hyperparameters.

F.3 Additional Hyperparameter Sensitivity

We additionally evaluate four model and training hyperparameters that are omitted from the main paper due to space limitations: model dimension, dropout rate, learning rate, and batch size. As in the main sensitivity study, only one hyperparameter is varied at a time, with all remaining settings fixed to the corresponding default configuration. For model dimension, learning rate, and batch size, values are expressed relative to their dataset-specific defaults where appropriate.

As shown in Fig. 5, VQPatch remains reasonably stable under variations in these model and optimization settings. The model dimension has limited influence on HumanActivity and PhysioNet around the default configuration, whereas an overly large dimension causes a noticeable degradation on USHCN. Dropout has only a minor effect on PhysioNet and HumanActivity, while larger dropout rates are less favorable on USHCN. The learning-rate results show that the default configuration provides a stable operating point across datasets, although moderate deviations can yield small dataset-specific improvements. Similarly, batch-size variations lead to relatively small changes around the default configuration, with somewhat greater sensitivity observed on HumanActivity. These experiments characterize robustness to reasonable configuration changes rather than establish statistically optimal values, and together with Sec. 5.4 they indicate that the gains of VQPatch do not depend on narrow tuning.

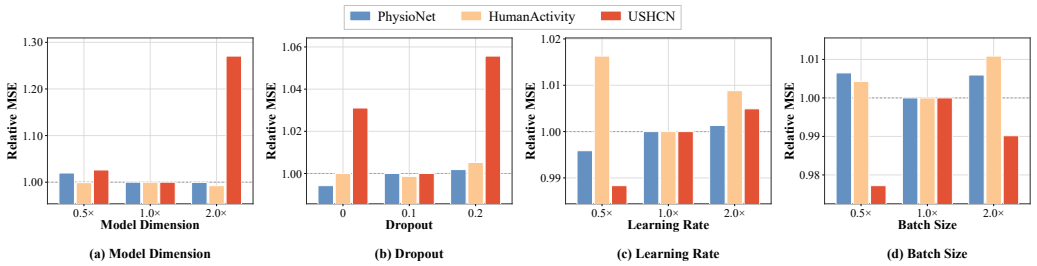


Figure 5: Sensitivity of VQPatch to additional model and training hyperparameters on PhysioNet, HumanActivity, and USHCN. MSE is normalized by the corresponding default configuration, where 1.0 denotes the default performance and lower values are better.

F.4 Additional Efficiency Analysis

To complement the efficiency analysis in Sec. 5.5, we provide a broader comparison across additional forecasting models and datasets. Table 7 reports parameter counts and peak GPU memory during training and inference, while Fig. 6 jointly visualizes MAE, model size, and training memory.

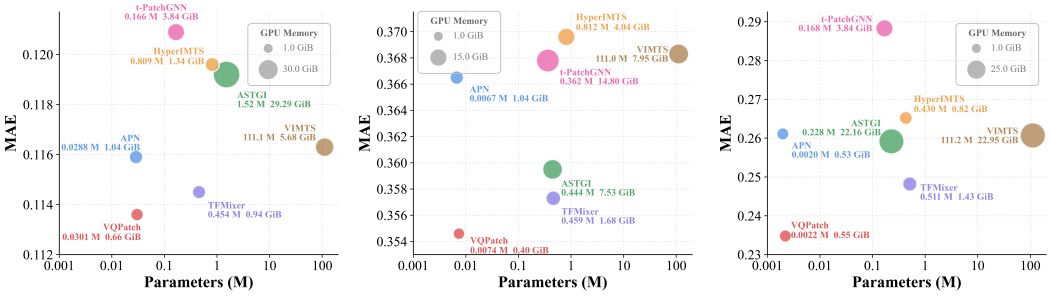


Figure 6: Accuracy–efficiency comparison across three IMTS datasets. The x-axis denotes parameter count, the y-axis reports MAE, and bubble size represents peak training GPU memory.

As shown in Fig. 6, VQPatch consistently occupies a favorable region of the accuracy–efficiency space across HumanActivity, PhysioNet, and USHCN. It achieves the lowest MAE among the compared methods while using only 0.0301M, 0.0074M, and 0.0022M parameters on the three datasets, respectively. Its training memory also remains compact at 0.66 GB, 0.40 GB, and 0.55 GB. In contrast, several recent methods require substantially larger parameter budgets or GPU memory without achieving lower forecasting error. This indicates that the performance gains of VQPatch do not rely on increasing model scale or memory consumption.

Table 7: Model size and GPU memory usage across three datasets. The best value in each column is shown in bold. Lower is better for all metrics.

Model	HumanActivity			USHCN			PhysioNet		
	Params. (M)	Train Mem. (GB)	Infer. Mem. (GB)	Params. (M)	Train Mem. (GB)	Infer. Mem. (GB)	Params. (M)	Train Mem. (GB)	Infer. Mem. (GB)
VQPatch	0.030069	0.6621	0.5078	0.002198	0.5508	0.3809	0.007373	**0.4004**	0.3477
APN	0.028789	1.0381	**0.3789**	**0.001970**	**0.5332**	**0.3418**	0.006701	1.0381	**0.3457**
KAFNet	**0.005327**	**0.5293**	0.4941	0.005215	**0.5332**	0.4727	**0.005711**	0.5313	0.4941
ASTGI	1.516804	29.2939	4.9053	0.227893	22.1572	2.3271	0.443588	7.5283	2.5010
t-PatchGNN	0.165634	3.8408	3.8213	0.167542	3.8428	0.4795	0.362178	14.7959	12.4854
TFMixer	0.454084	0.9365	0.9014	0.510671	1.4287	1.0381	0.459428	1.6787	1.0674
HyperIMTS	0.809090	1.3418	0.4648	0.429698	0.8164	0.5703	0.812162	4.0430	1.1934
VIMTS	111.100798	5.6797	1.6777	111.210429	22.9512	3.7344	111.013676	7.9502	1.7412
GraFITi	0.249409	0.7891	0.9980	0.250945	0.9629	0.5430	0.495617	0.9980	0.3887

Table 8: Computational complexity comparison for irregular multivariate time-series forecasting.

Method	Main operation	Time complexity	Memory complexity
VQPatch	VTQ aggregation + OPC	$\mathcal{O}(BSPD + BS)$	$\mathcal{O}(BSPD)$
APN	Soft patch aggregation	$\mathcal{O}(BSPD)$	$\mathcal{O}(BSPD)$
t-PatchGNN	Patch attention + adaptive GNN	$\mathcal{O}(BNP^2D + BPN^2D)$	$\mathcal{O}(BP^2 + BPN^2)$
VIMTS	TTCN + GCN + MAE	$\mathcal{O}(BSPD + BNP^2D + BPN^2D)$	$\mathcal{O}(BNP^2 + BPN^2)$
ASTGI	Point-neighbor search + propagation	$\mathcal{O}(BS^2d_c + B\ell SkD)$	$\mathcal{O}(BSk)$
HyperIMTS	Hypergraph message passing	$\mathcal{O}(B(T+U)MD + BU^2D + BM^2D)$	$\mathcal{O}(BM^2 + BM(T+U))$
KAFNet	Canonical alignment + kernel aggregation	$\mathcal{O}(BNL_cK)$	$\mathcal{O}(BNL_c)$
TFMixer	NUDFT + patch mixing	$\mathcal{O}(BNSK + BSPD + BNQK)$	$\mathcal{O}(BNK + BWP)$

Notes. B denotes the batch size; N the number of variables; S the total number of observations; P the number of patches; D the hidden dimension; Q the number of queries; and K the number of kernels or frequencies. For HyperIMTS, M, T, and U denote the numbers of observation nodes, unique timestamps, and variable hyperedges, respectively. L_c denotes the aligned sequence length. The complexities are asymptotic estimates based on the dominant operators. For VQPatch, Voronoi cell construction costs $\mathcal{O}(BS)$ after timestamps are sorted; sorting costs $\mathcal{O}\big(B\sum_n L_n \log L_n\big)$ when required.

Table 7 further shows that this compact profile extends to inference. On PhysioNet, VQPatch uses the lowest training memory among all compared methods (0.40 GB), reducing the requirement by 61.4% relative to APN and by more than 94% relative to ASTGI and t-PatchGNN. On HumanActivity and USHCN, its memory usage remains close to the lightest baselines while maintaining substantially lower MAE. Although APN or KAFNet can be slightly smaller in individual resource metrics, VQPatch provides a consistently strong balance among forecasting accuracy, parameter count, and GPU memory across all three datasets.

The measured resource usage is also consistent with the structure of the underlying operators, as summarized in Table 8. VQPatch retains the dominant $\mathcal{O}(BSPD)$ aggregation cost of APN, with only an additional linear $\mathcal{O}(BS)$ term for physical-time processing and observation conditioning. It therefore avoids introducing higher-order interactions over patches or variables. By contrast, several graph- and interaction-based baselines contain quadratic terms in the number of patches, variables, or observations. This complexity profile is consistent with the parameter and GPU-memory results above. The efficiency of VQPatch thus follows from its measure-level redesign rather than from increasing the computational order of the underlying model.