FLEXTEMPRULE: INTERPRETABLE TIME SERIES CLASSIFICATION VIA LEARNING FLEXIBLE TEMPORAL LOGIC RULES

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpretable time series classification is a significant task for safety-critical systems, as it can help experts gain insight of the system behaviors, especially when the model is intrinsically interpretable and provides intuitive yet rigorous rationales that reflect its actual execution. However, existing rule-based approaches either fail to guarantee model–rationale consistency or restrict the evaluation scope of each learned logical rule to a predefined fixed interval. These limitations inevitably hinder the capture of flexible temporal correlation properties that may be satisfied at different regions across time series. In this paper, we propose a novel neuro-symbolic model called FlexTempRule, which supports adaptive learning of both fixed and slidable temporal logic rules to achieve interpretable time series classification with model-consistent rationales. Specifically, the time series is first dynamically divided into patches of varying lengths, expecting that the boundary points do not break temporal continuity. Then, three types of temporal logic rules with different degrees of sliding flexibility, as well as statistical features, are automatically learned and adaptively weighted to form advanced features for classification. Comprehensive experiments on real-world univariate and multivariate datasets in safety-critical domains show that FlexTempRule attains superior accuracy among rule-based baselines and competitive performance against other interpretable methods. More importantly, discriminative temporal logic rules are explicitly provided, along with their flexibility nature, thereby enhancing user trust in the model and deepening understanding of specific scenarios.

1 INTRODUCTION

Interpretable time series classification is an important yet challenging task, especially for safety-critical systems such as industry, medicine, and public security (Theissler et al., 2022; Wang et al., 2025). These systems require not only accurate classification results but also intuitive and rigorous rationales that can faithfully reflect the internal execution of the model, thereby fostering users' trust.

Since post-hoc interpretation fails to restore the actual inference process, intrinsically interpretable approaches have attracted increasing attention, as they can provide explicit explanations as decision rationales. Traditional dictionary-based and shapelet-based methods (Lin et al., 2012; Grabocka et al., 2014) are able output significant patterns, but the pattern types are limited, and the models are difficult to optimize due to their non-differentiable discrete structures.

Recently, neuro-symbolic rule-based approaches for time series have been investigated to learn temporal logic rules via specially designed neural networks. NSTSC (Yan et al., 2022) defines structured templates and neural modules to simulate temporal correlations, but the derived rules cannot be guaranteed to be consistent with actual model executions and intuitive understanding. Through gradient grafting, TemporalRule (Wang et al., 2025) realizes end-to-end learning of expressive temporal logic rules as advanced features for classification, enabling the automatic extraction of model-consistent and compact rationales for domain experts. However, each rule there is evaluated on a predefined fixed patch, which inevitably restricts the flexibility of logical rules in characterizing temporal properties. For example, as illustrated in Figure 1(a), in ECG analysis, a waveform may straddle a fixed patch boundary, calling for a dynamic patch to capture it in full. In power consumption sequences,

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.

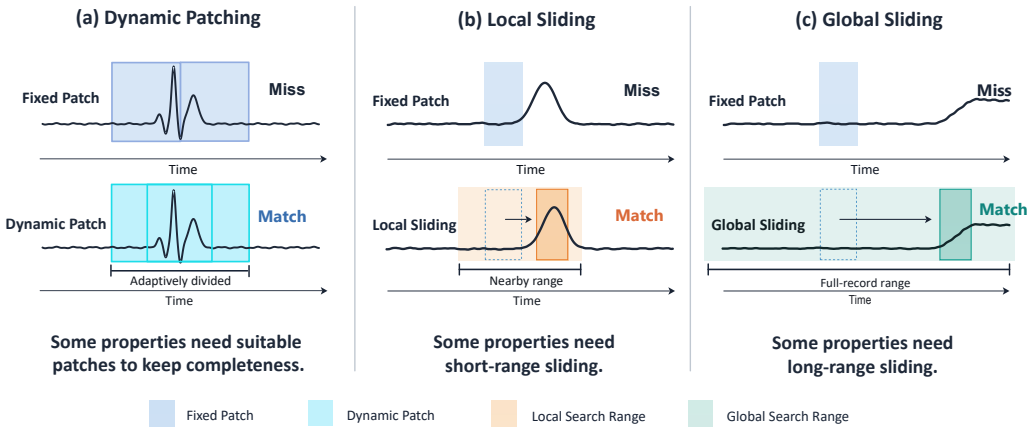

Figure 1: Motivation for dynamic patching and flexible rule evaluation.

the evening peak is likely to shift slightly, requiring the rule to slide locally around its expected position (Figure 1(b)). Moreover, an anomalous upward trend possibly occur anywhere in the time series, so sliding over the full sequence is needed to identify this critical signal (Figure 1(c)).

Therefore, in order to capture precise logical features that coincide with realistic temporal correlations in a specific scenario for accurate and interpretable classification, three progressive challenges need to be addressed: (1) How can we find a flexible and adaptive patch division, such that each significant rule lies within just patch and can thus be effectively learned by a pre-designed neural network? (2) For temporal properties that may appear in different regions, how can the rules in a specific patch slide to a certain degree to accommodate the time uncertainty of these properties? (3) Given different rules with varying degrees of flexibility, how can suitable evaluation scopes be selected adaptively for each time series so that these rules jointly contribute to the final classification?

In this paper, we propose FlexTempRule, a novel neuro-symbolic model that supports adaptive learning of both fixed and slidable temporal logic rules to achieve interpretable time series classification. Specifically, the time series is first dynamically divided into patches of varying lengths, with the expectation that the boundary points do not break temporal continuity. Then, three types of temporal logic rules with different degrees of sliding flexibility are automatically learned and adaptively weighted via gating mechanism to infer the category. The gradient grafting strategy and multiple loss functions are employed to ensure training stability and model–rationale consistency.

In summary, this paper makes the following key contributions:

- To the best of our knowledge, this is the first attempt to distinguish different flexibility degrees of temporal properties as advanced features for interpretable time series classification, strengthening model reliability with more realistic and intuitive rationales.
- A systematic framework is presented to enable the automatic induction of different types of temporal logic rules from time series data, involving dynamic patching, fixed rule learning, locally slidable rule learning, globally slidable rule learning, and an adaptive gating mechanism to choose among the three learning paradigms.
- Extensive experiments on diverse datasets from real-world safety-critical domains show that FlexTempRule significantly achieves superior accuracy among rule-based baselines and competitive performance against other interpretable methods, while providing explicit temporal logic rules with different degree of flexibility, which can enhance user trust and deepen the understanding of temporal correlations in specific scenarios.

2 Related Work

2.1 Intrinsically Interpretable Time Series Classification

Intrinsically interpretable classifiers map explicit properties to categories. Symbolic words and shapelets supply candidate patterns, but need a decision model to turn them into executable conditions (Lin et al., 2007; 2012; Senin & Malinchik, 2013; Ji et al., 2024; Li et al., 2021; Qu et al., 2024). Methods such as r-STSF instead obtain category rules from paths through trees trained on interval statistics (Cabello et al., 2024). TSProto fits a surrogate tree to a deep model (Bobek & Nalepa, 2025), and TimeSliver scores temporal segments (Pandey et al., 2026), but their explanations are separate from the original decision computation. As a hybrid model, InterpGN combines a shapelet-based logical expert with a deep network through a confidence gate (Wen et al., 2025). Although its expert can output rules, the final prediction also depends on the deep branch, thus losing rigorous interpretability.

2.2 Neuro-symbolic Models with explicit Learning of Temporal Logic Rules

Neuro-symbolic rule-based classifiers capture numerical predicates and temporal correlations in a logical rule form. NSTSC learns formulae within a predefined library and uses their satisfaction to route examples through a tree (Yan et al., 2022), but its derived rules are not completely consistent with model execution and human understanding. TemporalRule learns predicate bins, temporal parameters and intervals under fixed templates in a model-consistent way(Wang et al., 2025). However, each rule is evaluated within a predefined patch, so the location where its temporal property holds is fixed, which is not suitable for dynamic and uncertain scenarios. STELLE first generates and selects STL concepts, and then learns their relevance for classification (Ferfoglia et al., 2025), but the formula structures are not learned with the classifier in an end-to-end manner.

3 Preliminaries

3.1 Problem Formulation

Let $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ be a labeled dataset of N time series for classification on a category set $\mathcal{C}$, in which the i-th time series contains observations of V variables at T time points:

$$\mathbf{x}^{(i)} = \left\langle \mathbf{x}_0^{(i)}, \ldots, \mathbf{x}_{T-1}^{(i)} \right\rangle \in \mathbb{R}^{T \times V}, \qquad y^{(i)} \in \mathcal{C} = \{1, \ldots, C\}. \tag{1}$$

The definition covers univariate time series when $V = 1$ and multivariate time series when $V > 1$. Our goal is to learn a mapping from an input sequence $\mathbf{x}$ to category scores $\mathbf{z}_\Theta(\mathbf{x}) \in \mathbb{R}^C$ and predicts the category with the highest score. Main notation is summarized in Table 6.

3.2 Signal Temporal Logic

Signal Temporal Logic (STL) expresses time-bounded properties of real-valued signals (Maler & Nickovic, 2004). Let ψ denote atomic numerical predicates, each specifying a condition on a signal value of the form $\psi := (a_1 \leq x \leq a_2)$. Following TemporalRule (Wang et al., 2025), FlexTempRule uses five types of temporal rule templates:

$$\phi \in \left\{ \Box_I \psi,\ \Diamond_I \psi,\ \psi_1 \,\mathcal{U}_I\, \psi_2,\ \Box_I \Diamond_{[0,h]} \psi,\ \Diamond_I \Box_{[0,h]} \psi \right\}, \tag{2}$$

where $I = [a, b]$ denotes an interval with integer endpoints. Under Boolean semantics, Always($\Box_I \psi$) requires ψ to hold at every point in I, Eventually($\Diamond_I \psi$) requires ψ to hold at least once in I, Until($\psi_1 \mathcal{U}_I \psi_2$) requires ψ_2 to hold at some $t \in I$ and ψ_1 to hold at every point in $[a, t)$. The last two templates nest Always and Eventually with an inner interval $[0, h]$, where h is a non-negative integer. Always–Eventually($\Box_I \Diamond_{[0,h]} \psi$) requires ψ to hold at least once in $[t, t + h]$ for every $t \in I$, representing repetitive properties, while Eventually–Always($\Diamond_I \Box_{[0,h]} \psi$) requires ψ to hold throughout $[t, t + h]$ for at least one $t \in I$, characterizing flexible properties in a given window.

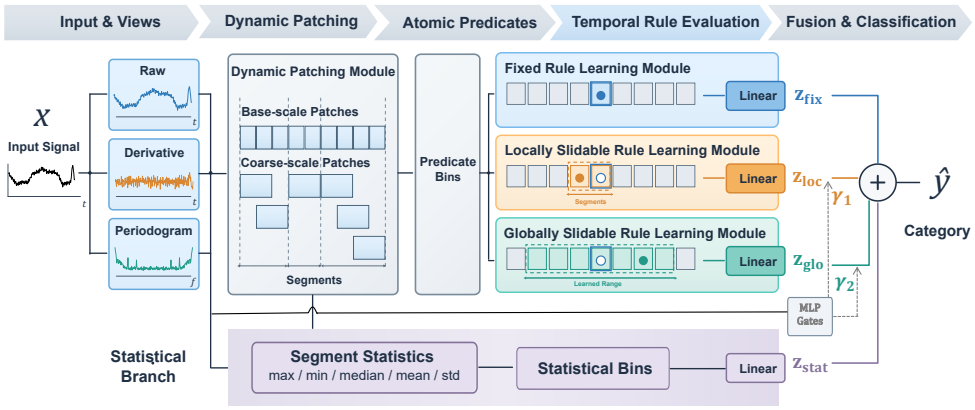

Figure 2: Framework of the proposed model FlexTempRule.

3.3 Multi-view Data Representation

Similar to previous work (Wang et al., 2025; Lines et al., 2018; Keogh & Pazzani, 2001), we represent each input sequence using raw, derivative, and periodogram views. The raw view retains the observed signal values, the derivative view depicts local changes, while the spectral view consists of the magnitudes of the discrete Fourier coefficients and is indexed by frequency bins rather than time points. Each variable–view pair is standardized using statistics from the training set.

The three views are concatenated along the channel dimension to form the augmented representation $\mathbf{x}^{\mathrm{A}} = (\mathbf{x}^{\mathrm{R}}, \mathbf{x}^{\mathrm{P}}, \mathbf{x}^{\mathrm{D}})$, where $\mathbf{x}^{\mathrm{R}}$ is just the raw sequence $\mathbf{x}$. Thus, a univariate input yields three channels, while a multivariate input yields three channels for each observed variable.

4 Proposed Model FlexTempRule

4.1 Framework

In rule-based classification, a temporal property and its appearing window (patch) can both distinguish categories, but sometimes the property may occur at different regions across sequences with same meanings. To address this, as shown in Figure 2, FlexTempRule retains fixed-position rules and introduces locally/globally slidable rules to represent the cases when the same property holds in an unfixed region within varying evaluation scopes. The three rule types share atomic predicates and temporal correlations, allowing the classifier to learn separate category associations with discriminative rules with different degree of flexibility.

At first, through the **dynamic patching module**, the model constructs segments reflecting temporal continuity from the training data, and uses them to select valid patches at two scales: the base scale and the coarse scale with twice the patch length. Then, the **fixed rule learning module** capture temporal logic rules on each valid patch, the **locally slidable rule learning module** transfers each rule to target patches of the same scale within the segment of the source patch, and the **globally slidable rule learning module** further search on a longer range that can cross segment boundaries for each base-scale rule. In addition, auxiliary statistical features describing the level and variability of values within each segment are extracted to complement logical rules. The four branches produce separate category scores, and two input-dependent gates adjust their contributions to the final classification.

4.2 DYNAMIC PATCHING MODULE

TemporalRule evaluates temporal properties within a fixed patch configuration (Wang et al., 2025). Including more patches at different positions and scales can cover a wider range of temporal properties, but assigning rules to every candidate patch also increases model complexity and weakens significant features. To balance coverage and complexity, we actively select patches with the aim of preserving the continuity of temporal properties. Since the relevant structure varies across scenarios (datasets), we first derive segment boundaries from the training data to guide this selection.

Specifically, we adapt the bottom-up refinement strategy of InteDisUX (Tran et al., 2025) to iteratively merge adjacent regions in each augmented channel d of each training instance. The initial regions are just the base-scale patches of length n_0. We restrict the combined length not exceeding a threshold $h_{\max}$, and assess the reasonability of each merge using two metrics without using category labels, discrimination gain (DG) and temporal-structure score (TSS). DG measures the increase in the mean symmetric Hausdorff distance between adjacent regions after a candidate merge (Huttenlocher et al., 1993), which favors retaining boundaries between regions with different raw values. TSS describes stable levels and local deviations within the merged region, designed to encourage structures tending to form the three types of temporal rules (Always, Eventually, and Until).

We combine them with equal weights to score each feasible merge m on a channel of time series $\mathbf{x}$:

$$\mathrm{Score}(m) = \mathrm{DG}(\mathrm{x}, m) + \mathrm{TSS}(\mathrm{x}, m). \tag{3}$$

We repeatedly merge the highest-scoring feasible pair of regions until the preset per-channel segment number M_{pre} is reached or no further merge satisfies the length constraint. The final boundaries are aggregated by training-set voting, which divide each sequence into M contiguous **segments**.

Next, we generate candidate **patches** for the evaluation of temporal logic rules, at both the base and coarse scales with stride equal to the base-scale patch length n_0, so the coarse-scale patches are overlapped. Let $\mathcal{P}_{n,p}$ denote the length-n candidate starting at time point $p \cdot n_0$, with p counted from zero. We only retain those patches not crossing segment boundaries, and collect their indices in $\mathcal{K}_n$. For each valid patch, $g(\mathcal{P}_{n,p}) = s$ records the segment containing the patch.

Besides guiding patch selection, the division of segments also delimits local sliding and provide the scope for statistical feature extraction as described in below. Details of the merge criteria and boundary aggregation are provided in Appendix B.

4.3 FIXED RULE LEARNING MODULE

As the basis of the learning framework, this module establishes the numerical properties and temporal correlations of each rule while retaining information about where they occur. To construct atomic predicates, we learn K bin centers for each of the $D = 3V$ augmented channels. For a value $x^{\mathrm{A}}_{t,d}$ and center $c_{d,k}$, the continuous assignment and its hard indicator are defined as

$$\pi_{t,d,k} = \frac{\exp[-\eta(x^{\mathrm{A}}_{t,d} - c_{d,k})^2]}{\sum_{k'=1}^{K} \exp[-\eta(x^{\mathrm{A}}_{t,d} - c_{d,k'})^2]}, \qquad o_{t,d,k} = \mathbb{I}\left[k = \arg\max_{k'} \pi_{t,d,k'}\right]. \tag{4}$$

where $\eta > 0$ controls the sharpness of the assignment, $o_{t,d,k}$ used in forward rule evaluation indicates whether the value is assigned to the bin k with the nearest center, while $\pi_{t,d,k}$ provides gradients for predicate learning and contributes to the clustering objective.

These predicates are then processed by five temporal templates similar to TemporalRule (Wang et al., 2025): Always, Eventually, Until, Always–Eventually, and Eventually–Always. We permit the interaction of different views through the Until operator, enabling the capture of richer temporal correlations than TemporalRule. However, taking into account the complexity, for multivariate data, if either Until predicates comes from the spectral view, both predicates must originate from the same original variable. The detailed design of the five temporal operators is presented in Appendix C. For every scale $n \in \{n_0, 2n_0\}$ and valid patch index $p \in \mathcal{K}_n$, rule r has learnable parameters $\theta_{p,r}$ including bin centers and time bounds. Letting Sat be Boolean rule evaluation on the augmented input within a specified patch, the truth values of fixed rules and the category scores are defined as:

$$q_{p,r}(\mathbf{x}^{\mathrm{A}}) = \mathrm{Sat}(\mathbf{x}^{\mathrm{A}}, \theta_{p,r}; \mathcal{P}_{n,p}), \qquad \mathbf{z}_{\mathrm{fix}}(\mathbf{x}^{\mathrm{A}}) = \mathbf{W}_{\mathrm{fix}}\mathbf{q}(\mathbf{x}^{\mathrm{A}}) + \mathbf{b}_{\mathrm{fix}}, \tag{5}$$

where $\mathbf{q}(\mathbf{x}^{\mathrm{A}})$ concatenates the truth values of rules from all valid source patches at both scales.

4.4 Locally Slidable Rule Learning Module

When a temporal property shifts along the time series, a rule evaluated only on its source patch may no longer detect it. To this end, for each source patch p, we search valid patches of the same scale within the segment $g(\mathcal{P}_{n,p})$. The target patch set and the truth values of locally slidable rules are:

$$\mathcal{J}_n(p) = \{j \in \mathcal{K}_n : g(\mathcal{P}_{n,j}) = g(\mathcal{P}_{n,p})\}, \qquad \ell_{p,r}(\mathbf{x}^{\mathrm{A}}) = \max_{j \in \mathcal{J}_n(p)} \mathrm{Sat}(\mathbf{x}^{\mathrm{A}}, \theta_{p,r}; \mathcal{P}_{n,j}). \quad (6)$$

Here, the same $\theta_{p,r}$ is shared throughout all the patches in the target set to emsure rule consistency. $\ell_{p,r}(\mathbf{x}) = 1$ if and only if at least one target patch satisfies the rule. These pooled truth values remain indexed by source rule and can be mapped to category scores $\mathbf{z}_{\mathrm{loc}}(\mathbf{x}^{\mathrm{A}})$ through a separate linear classifier similar to $\mathbf{z}_{\mathrm{fix}}$ in Equation 5.

4.5 Globally Slidable Rule Learning Module

For more flexible properties which may occur in different phases, the local range constraint limits their effects for classification. To allow positional variation across segments while retaining a rule-specific scope, this module further learns an admissible interval at the patch level for each base-scale rule. For patch p and rule r, we introduce two unconstrained parameters $u_{p,r}$ and $v_{p,r}$ with Sigmoid function, which represent expanding ratios in two directions and determine continuous search endpoints defined over patch indices:

$$a_{p,r} = p - p \cdot \sigma(u_{p,r}), \qquad b_{p,r} = p + (K_0 - 1 - p) \cdot \sigma(v_{p,r}). \quad (7)$$

This guarantees $0 \le a_{p,r} \le p \le b_{p,r} \le K_0 - 1$, where $K_0 = T/n_0$, so the search range always contains the source patch, and is shared across instances. Then, through rounding the continuous endpoints to get integer ones $A_{p,r}$ and $B_{p,r}$, the global branch evaluates the same source rule on each target patch in this range and takes the maximum Boolean truth values:

$$e_{p,r}(\mathbf{x}^{\mathrm{A}}) = \max_{A_{p,r} \le j \le B_{p,r}} \mathrm{Sat}(\mathbf{x}^{\mathrm{A}}, \theta_{p,r}; \mathcal{P}_{n_0,j}), \quad (8)$$

Thus, $e_{p,r}(\mathbf{x}) = 1$ indicates that the rule holds on at least one patch in the learned range, and the corresponding category scores $\mathbf{z}_{\mathrm{glo}}(\mathbf{x}^{\mathrm{A}})$ are computed also in the same way as $\mathbf{z}_{\mathrm{fix}}$ in Equation 5.

4.6 Adaptive Gating and Fusion

Besides temporal logic rules, statistical features also provide additional information about the level and variability of values over an entire region. We therefore compute the maximum, minimum, median, mean and standard deviation for each augmented channel within each segment. These summaries are converted into hard predicates through learnable bins, and concatenated as input to another linear classifier to obtain $\mathbf{z}_{\mathrm{stat}}(\mathbf{x}^{\mathrm{A}})$.

For the fusion of the four types of category scores, we employ two input-dependent scalar gates to modulate the local and global branches, while adding the fixed and statistical logits directly to the final score. Each gate is an independent two-layer perceptron from the flattened representation:

$$\gamma_h(\mathbf{x}^{\mathrm{A}}) = \sigma\big(\mathbf{w}_{h,2}^{\top} \mathrm{ReLU}\big(\mathbf{W}_{h,1}(\mathbf{x}^{\mathrm{A}}) + \mathbf{b}_{h,1}\big) + b_{h,2}\big), \qquad h \in \{1, 2\}, \quad (9)$$

where $h = 1$ and $h = 2$ index the local and global gates, respectively, and the Sigmoid function constrains each weight to $(0, 1)$.

In this way, the category scores $\mathbf{z}_{\mathrm{fix}}, \mathbf{z}_{\mathrm{loc}}, \mathbf{z}_{\mathrm{glo}}, \mathbf{z}_{\mathrm{stat}} \in \mathbb{R}^C$ from four branches are combined as

$$\mathbf{z}(\mathbf{x}^{\mathrm{A}}) = \mathbf{z}_{\mathrm{fix}}(\mathbf{x}^{\mathrm{A}}) + \gamma_1(\mathbf{x}^{\mathrm{A}})\, \mathbf{z}_{\mathrm{loc}}(\mathbf{x}^{\mathrm{A}}) + \gamma_2(\mathbf{x}^{\mathrm{A}})\, \mathbf{z}_{\mathrm{glo}}(\mathbf{x}^{\mathrm{A}}) + \mathbf{z}_{\mathrm{stat}}(\mathbf{x}^{\mathrm{A}}). \quad (10)$$

Here, $\mathbf{z}_{\Theta}(\mathbf{x}) = \mathbf{z}(\mathbf{x}^{\mathrm{A}})$, with Θ denoting the learned model parameters. This fusion allows the contribution of slidable rules to adapt to the input time series.

Note that compared to TemporalRule (Wang et al., 2025), our model omits the logical layer to avoid the combinatorial explosion problem caused by the growing number of temporal logic rules, which is compensated for by more expressive slidable rules.

Table 1: Datasets used in the experiments. T is sequence length and V is the number of variables.

Type	Dataset	#Train	#Test	T	V	#Category
Univariate	Epilepsy2	80	11,420	178	1	2
	NerveDamage	163	41	1,500	1	3
	ECG200	100	100	96	1	2
	ECG5000	500	4,500	140	1	5
	EOGVerticalSignal (EOG)	362	362	1,250	1	12
	SharePriceIncrease (SPI)	965	966	60	1	2
	GunPoint	50	150	150	1	2
	GunPointOVY	136	315	150	1	2
Multivariate	ArticularyWordRecognition (AWR)	275	300	144	9	25
	BasicMotions	40	40	100	6	4
	ERing	30	270	65	4	6
	Libras	180	180	45	2	15
	PenDigits	7,494	3,498	8	2	10
	RacketSports	151	152	30	6	4

4.7 Training and Inference

Predicate assignment, STL rule evaluation and global sliding produce discrete values in the forward pass, requiring gradients to learn their parameters. Following TemporalRule (Wang et al., 2025), we connect a hard value h to a continuous surrogate $\widetilde{h}$ through gradient grafting:

$$\mathrm{GradientGrafting}(h, \widetilde{h}) = \mathrm{stopgrad}(h - \widetilde{h}) + \widetilde{h}. \tag{11}$$

The expression enables the classification loss to update the parameters that produce discrete and model-consistent truth values of temporal logic rules. Moreover, continuous predicate assignments, soft surrogate computations for temporal operators and global range masks provide gradients for learning corresponding parameters.

The training objective combines the classification loss with multiple regularizations:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{clu}} \left(\mathcal{L}_{\mathrm{clu}}^{\mathrm{temp}} + \mathcal{L}_{\mathrm{clu}}^{\mathrm{stat}} \right) + \lambda_{\mathrm{spa}} \|\Theta_{\mathrm{lin}}\|_1 + \lambda_{\mathrm{short}} \mathcal{L}_{\mathrm{short}}^{(1)} + \lambda_{\mathrm{short}} \mathcal{L}_{\mathrm{short}}^{(2)} + \lambda_{\mathrm{long}} \mathcal{L}_{\mathrm{long}}^{(2)}, \tag{12}$$

where $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss, while $\mathcal{L}_{\mathrm{clu}}^{\mathrm{temp}}$ and $\mathcal{L}_{\mathrm{clu}}^{\mathrm{stat}}$ respectively adapt the temporal and statistical predicate centers to the observed values. Θ_{lin} contains the linear weights of the four-branch classifiers and gate networks, and the ℓ_1 penalty encourages sparsity in these weights. $\mathcal{L}_{\mathrm{short}}^{(1)}$ and $\mathcal{L}_{\mathrm{short}}^{(2)}$ penalize the internal STL and global sliding intervals when they degenerate to single points, respectively. Inversely, $\mathcal{L}_{\mathrm{long}}^{(2)}$ penalizes excessively long range of global search, balancing coverage and locality of temporal properties. Details are provided in Appendix D.

At inference, the discrete version is adopted to guarantee the strict consistency between the model execution and learned temporal logic rules.

5 Experiments

We evaluate the classification performance of FlexTempRule, examine the role of each branch providing category scores, and test whether the exported rules actually determine classification results.

5.1 Datasets

We use the eight univariate datasets in TemporalRule (Wang et al., 2025) and six UEA multivariate datasets (Bagnall et al., 2018), retaining their supplied train/test splits. Table 1 reports the statistics of these datasets, three of which are given abbreviated names. Details appear in Appendix E.1.

5.2 Baselines

The univariate comparison includes the symbolic bag-of-pattern methods BOP and SAX–VSM (Lin et al., 2012; Senin & Malinchik, 2013), the shapelet methods LS and ST (Grabocka et al., 2014;

Table 2: Classification accuracy (%) on univariate datasets. The best scores except the hybrid model InterpGN are marked in bold.

Method	Epilepsy2	NerveDamage	ECG200	ECG5000	EOG	SPI	GunPoint	GunPointOVY	Average
BOP	88.98	**100.00**	78.57	90.98	14.36	68.53	98.67	93.02	79.14
SAX–VSM	90.46	97.56	83.54	91.45	21.82	68.53	98.67	87.30	79.92
LS	76.38	46.34	87.14	94.02	17.96	68.43	**100.00**	99.68	73.74
ST	94.75	97.56	84.02	94.34	44.20	62.94	**100.00**	94.92	84.09
RRL	75.15	46.34	78.00	92.18	25.14	58.80	80.67	**100.00**	69.54
NSTSC	92.57	**100.00**	87.00	93.40	37.29	65.84	96.00	**100.00**	84.01
TemporalRule	94.80	**100.00**	89.00	94.34	45.86	69.15	**100.00**	**100.00**	86.64
TimeSliver	95.15	95.12	90.33	93.05	39.96	68.98	98.00	89.52	83.76
InterpGN-SBM	93.93	95.12	81.67	92.39	43.74	68.67	94.22	94.50	83.03
FlexTempRule	**95.70**	**100.00**	**94.33**	**94.37**	**46.87**	**69.32**	99.56	**100.00**	**87.52**
InterpGN	95.23	100.00	93.33	94.47	53.87	69.12	100.00	100.00	88.25

Table 3: Classification accuracy (%) on multivariate datasets. The best scores except the hybrid model InterpGN are marked in bold.

Method	AWR	BasicMotions	ERing	Libras	PenDigits	RacketSports	Average
TimeSliver	**99.33**	**100.00**	84.10	83.33	**98.10**	**92.10**	92.83
InterpGN-SBM	99.30	**100.00**	**96.70**	86.00	97.10	89.70	94.80
FlexTempRule	96.00	**100.00**	95.93	**90.74**	97.70	91.01	**95.23**
InterpGN	99.10	100.00	93.30	96.90	99.00	90.10	96.40

Hills et al., 2014), as well as the rule-based methods RRL, NSTSC, and TemporalRule (Wang et al., 2021; Yan et al., 2022; Wang et al., 2025). We also include TimeSliver (Pandey et al., 2026) and InterpGN (Wen et al., 2025) as recent interpretable classifiers, and report InterpGN-SBM, the logical expert of InterpGN, separately. Results for TemporalRule and its baselines are taken from that paper (Wang et al., 2025), while results for TimeSliver, InterpGN, and InterpGN-SBM are obtained via our runs with three random seeds. Details appear in Appendix E.1.

5.3 Experimental Settings

FlexTempRule is implemented in PyTorch and trained with Adam at an initial learning rate of 0.02, multiplied by 0.75 every 50 epochs, for up to 500 epochs. We set $(\lambda_{\text{clu}}, \lambda_{\text{spa}}, \lambda_{\text{short}}, \lambda_{\text{long}}) = (0.05, 10^{-4}, 0.5, 0.01)$ and use a bin-assignment sharpness of $\eta = 100$. For dynamic patching, the target number of segments per channel depends on the sequence length T. We set $M_{\text{pre}} = \lfloor T/16 \rfloor$ and the base-scale patch length to $n_0 = 8$ for $T < 1000$, while $M_{\text{pre}} = \lfloor T/128 \rfloor$ and $n_0 = 64$ for $T > 1000$. For each dataset, we follow the official train–test split and report the mean test accuracy over three runs with different random seeds. We apply early stopping with a patience of 200 epochs. Additional experimental settings are given in Appendix E.2.

5.4 Overall Results

FlexTempRule has the highest or tied-highest accuracy on seven of eight univariate datasets among fully interpretable methods, except the hybrid model InterpGN (2). This indicates that flexible rule evaluation can enhance the discriminative power of temporal logic rules in real-world scenarios. InterpGN attains higher results with a deep branch, which sacrifices interpretability in some cases, and its logical expert SBM alone underperforms FlexTempRule on all eight datasets.

For multivariate datasets, FlexTempRule averages 95.23% across the six datasets, outperforming TimeSliver and InterpGN-SBM but below the hybrid InterpGN (Table 3). Its advantage over TimeSliver and InterpGN-SBM is particularly pronounced on Libras, which contains sign-language hand trajectories. Similarly, on the finger-posture recognition task ERing, it outperforms TimeSliver and InterpGN while being slightly below InterpGN-SBM. Thus, the advantage and stability of our model are most evident in scenarios with rich and flexible temporal properties.

Three-run standard deviations of these results are reported in Appendix F.1.

Table 4: Classification accuracy (%) of ablated models, with best scores in bold.

Variant	Epilepsy2	NerveDamage	ECG200	ECG5000	SPI	GunPoint	GunPointOVY	Average
Fixed only	94.44	**100.00**	88.33	93.56	68.56	96.67	**100.00**	91.65
Statistical only	**96.13**	**100.00**	89.33	93.93	68.67	97.11	**100.00**	92.17
−Local	95.48	**100.00**	91.00	94.03	69.08	97.11	**100.00**	92.39
−Global	95.22	**100.00**	90.33	94.14	69.19	96.67	**100.00**	92.22
−Statistical	94.77	**100.00**	89.00	93.87	68.91	96.44	**100.00**	91.86
Full	95.70	**100.00**	**94.33**	**94.37**	**69.32**	**99.56**	**100.00**	**93.33**

Table 5: Classification consistency (%) after retaining the top 20% of contributive features.

Method	Epilepsy2	NerveDamage	ECG200	ECG5000	EOG	SPI	GunPoint	GunPointOVY
Instance-level Consistency								
TemporalRule	100.00	100.00	99.00	100.00	80.48	100.00	100.00	100.00
FlexTempRule	100.00	99.19	100.00	100.00	100.00	100.00	100.00	100.00
Dataset-level Consistency								
TemporalRule	95.14	69.11	75.00	53.33	52.58	98.72	84.89	95.56
FlexTempRule	99.81	86.18	98.33	99.73	97.88	99.79	99.78	100.00

5.5 Ablation Study

We examine the roles of the four branches through examining variants with incomplete branches on seven univariate datasets. As shown in Table 4, removing the local, global, or statistical branch will lower accuracy relative to the full model on all five datasets below ceiling. This confirms that these branches yield complementary evidence: fixed rules preserve source-patch positions, local/global rules accommodate pattern shifts over different ranges, and segment statistics describe value distributions. Exceptionally, pure statistical features perform better on Epilepsy2, which may be attributed to the clear difference between categories in the within-window variability of EEG values, and the parameters of temporal rules may be harder to estimate reliably from only 80 training instances. Three-seed deviations appear in Appendix F.2.

5.6 Interpretability Analysis

We test whether exported high-contribution rules can reconstruct category scores and preserve full-model classification. Examples of learned logical rules are presented in Appendix F.5.

Here, we consider both instance-level ranks and dataset-level ranks. The former adopts absolute rule contributions to the full model's predicted category for each test instance, while the latter uses training-set mean absolute contributions and remain fixed for testing. We report the classification consistency just retaining top rule features in Table 5, with details reported in Appendix F.3.

With instance-level selection, the top 20% of FlexTempRule features preserve 99.19–100% of full-model predictions across the eight datasets. Training-derived dataset-level subsets retain 86.18–100% prediction consistency, exceeding TemporalRule on all eight datasets. This validates the model–rationale consistency for temporal logic rules provided by our model, thereby ensuring their fidelity. In addition, although the full feature bank is larger, the logical features in FlexTempRule are short and have an explicit evaluation scope, so they are more intuitive and valuable for understanding specific scenarios.

6 Conclusion

This paper explores a significant optimization of rule-based interpretable time series classification: the learned temporal logic rules become flexible, no longer confined to fixed patches and able to slide along the sequence to varying degrees. This mechanism fully leverages temporal correlation properties when determining the category of a time series, especially in dynamic and open-ended scenarios. Comprehensive experiments demonstrate its classification effectiveness, module complementarity, and rule interpretability. A promising direction for future work is to pursue a smaller rule bank that preserves the link between rule scope, execution, and category scores.

AI USE STATEMENT

Generative AI was used to draft the initial manuscript of some technical parts from an outline prepared by the authors. The authors subsequently revised that draft. AI also assisted the authors in preparing supplementary explanations and comparing the relevant descriptions with existing implementation and experiment records, and helped them create schematic drafts of figures. The final figures were completed manually on the basis of those drafts. The authors take responsibility for the final content, conclusions, and artifacts.

ETHICS STATEMENT

All data used in this study come from public benchmarks. No new experiments involving human participants were conducted, and no non-public data were used.

REPRODUCIBILITY STATEMENT

The datasets and splits are listed in Table 1, with training settings in Appendix E. Appendix C specifies the implementation of the five temporal operators, Appendix D details the training objective, while Appendices F & G provide supplementary results and complexity analysis. We also upload source codes in the supplementary materials.

REFERENCES

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Szymon Bobek and Grzegorz J. Nalepa. TSProto: Fusing deep feature extraction with interpretable glass-box surrogate model for explainable time-series classification. *Information Fusion*, 124: 103357, 2025.

Nestor Cabello, Elham Naghizade, Jianzhong Qi, and Lars Kulik. Fast, accurate and explainable time series classification through randomization. *Data Mining and Knowledge Discovery*, 38(2): 748–811, 2024.

Irene Ferfoglia, Simone Silvetti, Gaia Saveri, Laura Nenzi, and Luca Bortolussi. Guided by stars: Interpretable concept learning over time series via temporal logic semantics. *arXiv preprint arXiv:2511.04244*, 2025.

Josif Grabocka, Nicolas Schilling, Martin Wistuba, and Lars Schmidt-Thieme. Learning time-series shapelets. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 392–401, 2014.

Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28(4):851–881, 2014.

Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

Cun Ji, Yanxuan Wei, and Xiangwei Zheng. Shapelet selection for time series classification. *Applied Soft Computing*, 167:112431, 2024.

Eamonn J. Keogh and Michael J. Pazzani. Derivative dynamic time warping. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 1–11, 2001.

Guozhong Li, Byron Choi, Jianliang Xu, Sourav S. Bhowmick, Kwok-Pan Chun, and Grace Lai-Hung Wong. ShapeNet: A shapelet-neural network approach for multivariate time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8375–8383, 2021.

Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

Jessica Lin, Rohan Khade, and Yuan Li. Rotation-invariant similarity in time series using bag-of-patterns representation. *Journal of Intelligent Information Systems*, 39(2):287–315, 2012.

Jason Lines, Sarah Taylor, and Anthony Bagnall. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data*, 12(5):52:1–52:35, 2018.

Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pp. 152–166. Springer, 2004.

Akash Pandey, Payal Mohapatra, Wei Chen, Qi Zhu, and Sinan Keten. TimeSliver: Symbolic-linear decomposition for explainable time series classification. In *International Conference on Learning Representations*, 2026.

Eric Qu, Yansen Wang, Xufang Luo, Wenqiang He, Kan Ren, and Dongsheng Li. CNN kernels can be the best shapelets. In *International Conference on Learning Representations*, 2024.

Pavel Senin and Sergey Malinchik. SAX-VSM: Interpretable time series classification using SAX and vector space model. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 1175–1180, 2013.

Andreas Theissler, Francesco Spinnato, Udo Schlegel, and Riccardo Guidotti. Explainable AI for time series classification: A review, taxonomy and research directions. *IEEE Access*, 10:100700–100724, 2022.

Viet-Hung Tran, Zichi Zhang, Tuan Dung Pham, Ngoc Phu Doan, Anh-Tuan Hoang, Peixin Li, Hans Vandierendonck, Ira Assent, and Son T. Mai. InteDisUX: Intepretation-guided discriminative user-centric explanation for time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20921–20928, 2025.

Yang Wang, Jiaqi Zhu, Miaomiao Li, Jiang Liu, Yilin Li, Yi Yang, Jiafan Li, and Hongan Wang. Learning reliable and intuitive temporal logic rules for interpretable time series classification. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, volume 2, pp. 3067–3078. ACM, 2025.

Zhuo Wang, Wei Zhang, Ning Liu, and Jianyong Wang. Scalable rule-based representation learning for interpretable classification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 30479–30491, 2021.

Yunshi Wen, Tengfei Ma, Ronny Luss, Debarun Bhattacharjya, Achille Fokoue, and Anak Agung Julius. Shedding light on time series classification using interpretability gated networks. In *International Conference on Learning Representations*, 2025.

Ruixuan Yan, Tengfei Ma, Achille Fokoue, Maria Chang, and Agung Julius. Neuro-symbolic models for interpretable time series classification using temporal logic description. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 618–627. IEEE, 2022.

A MAIN NOTATIONS AND DEFINITIONS

The main notations used in this paper are listed in Table 6.

Table 6: Main notations and Descriptions.

Notation	Description
$\mathcal{D}$	Labeled time-series dataset
N	Number of time series
T	Length of each time series
V	Number of variables
$\mathcal{C}$	Set of categories
C	Number of categories
$\mathbf{x}^{\mathrm{A}}$	Augmented time-series representation
D	Number of augmented channels ($D = 3V$)
M	Number of segments
$\mathcal{P}_{n,p}$	Patch of length n with source index p
n_0, $2n_0$	Base- and coarse-scale patch lengths
$\mathcal{K}_n$	Set of valid source-patch indices
$g(\mathcal{P}_{n,p})$	Index of the segment containing patch $\mathcal{P}_{n,p}$
ψ	Atomic predicate
ϕ	Temporal-logic formula
I	Discrete temporal interval
K	Number of value bins
$c_{d,k}$	center of bin k for augmented channel d
$\pi_{t,d,k}$	Soft assignment probability to bin k
$o_{t,d,k}$	Hard truth value of an atomic predicate
$\theta_{p,r}$	Parameters of source rule r at patch p
$\mathcal{J}_n(p)$	Admissible target patch indices for local sliding
$q_{p,r}$	Fixed response of source rule (p, r)
$\ell_{p,r}$	Locally slidable response of source rule (p, r)
$e_{p,r}$	Globally slidable response of source rule (p, r)
$[a_{p,r}, b_{p,r}]$	Continuous global search range
$[A_{p,r}, B_{p,r}]$	Hard global search range
$\mathbf{z}_{\mathrm{fix}}$	Category logits of the fixed position branch
$\mathbf{z}_{\mathrm{loc}}$	Category logits of the local branch
$\mathbf{z}_{\mathrm{glo}}$	Category logits of the global branch
$\mathbf{z}_{\mathrm{stat}}$	Category logits of the statistical branch
γ_1, γ_2	Gates for local and global rule evidence

B MERGING METRICS AND DETAILS FOR DYNAMIC PATCHING

In the dynamic patching module, the merge scores are computed separately for each normalized augmented channel of each training instance. Let $\mathcal{R} = \langle R_1, \ldots, R_J \rangle$ be the current ordered partition for one channel, where each R_j is a contiguous region. For discrimination gain (DG), the input to the Hausdorff distance is the set $\mathcal{V}_j = \{x^{\mathrm{A}}_{t,d} : t \in R_j\}$ of real scalar values in region R_j of channel d. The implementation reshapes these values into one-dimensional points, and neither time indices nor the order of values enter this distance. Thus, regions of different lengths can be compared without temporal alignment.

Discrimination Gain (DG). For two nonempty value sets P and Q, we use the symmetric Hausdorff distance (Huttenlocher et al., 1993)

$$h(P, Q) = \max_{a \in P} \min_{b \in Q} |a - b|, \qquad H(P, Q) = \max\{h(P, Q), h(Q, P)\}. \tag{13}$$

The implementation computes both directed distances and takes their maximum. Following the adjacent-region comparison in InteDisUX (Tran et al., 2025), we define

$$\overline{H}(\mathcal{R}) = \begin{cases} \dfrac{1}{J-1}\displaystyle\sum_{j=1}^{J-1} H(\mathcal{V}_j, \mathcal{V}_{j+1}), & J > 1, \\ 0, & J = 1. \end{cases} \tag{14}$$

For a candidate merge of R_i and R_{i+1}, let $\mathcal{R}^{(i)}$ replace these two regions with their concatenation while retaining all others. The discrimination gain for this merge is

$$\mathrm{DG}_i(\mathbf{x}_d^{\mathrm{A}}) = \overline{H}(\mathcal{R}^{(i)}) - \overline{H}(\mathcal{R}). \tag{15}$$

Both means cover all adjacent pairs in their respective partitions, including unchanged pairs. If the merge leaves one region, the post-merge mean is zero. A larger DG favors boundaries that separate more distinct sets of raw values on average. DG uses no class labels or classifier outputs.

Temporal-Structure Score (TSS). Let $\mathbf{v} = (x_k, \ldots, x_{k+L})$ be the ordered values in the candidate merged region, with $\mathbf{v}^{\mathrm{L}} = (x_k, \ldots, x_{\lfloor (k+L)/2 \rfloor})$ and $\mathbf{v}^{\mathrm{R}}$ containing the remaining values. The split is at the midpoint of the merged region, regardless of the original boundary. With the overbar denoting the mean and Var the population variance, the temporal-structure score is defined as

$$\mathrm{TSS}(\mathbf{v}) = 0.3\,\frac{1}{\mathrm{Var}(\mathbf{v}) + 10^{-10}} + 0.3\,\max_{u \in \mathbf{v}} |u - \overline{\mathbf{v}}| + 0.4\,\frac{\max_{u \in \mathbf{v}^{\mathrm{R}}} |u - \overline{\mathbf{v}^{\mathrm{R}}}|}{\mathrm{Var}(\mathbf{v}^{\mathrm{L}}) + 10^{-10}}. \tag{16}$$

The three terms favor low overall variance, large deviations within the merged region, and a stable first half with deviations in the second. They just correspond to the tendency of forming temporal logic rules based on the Always, Eventually, and Until templates, respectively. If $L = 1$, the last term is set to zero. Unlike the Hausdorff distance, TSS depends on the ordering of values. It is a heuristic motivated by persistent, ordered, and event-like patterns, not a direct measure of STL satisfaction or the faithfulness gain used by InteDisUX.

A pair is eligible only when its combined length does not exceed a threshold $h_{\max}$. Among eligible pairs, the algorithm merges the one maximizing the summation of the two metrics, $\mathrm{DG}_i(\mathbf{x}_d^{\mathrm{A}}) + \mathrm{TSS}(\mathbf{v})$, and then repeats this process until the target region count is reached or no eligible pair remains.

To aggregate these individual boundaries generated from each channel, for univariate data, we take the union of boundaries selected for the three views. For multivariate data, we sum up training-set boundary votes across all variable–view channels, and then select one shared boundary set subject to the segment-count and length constraints.

C Implementation of temporal logic operators

For a patch, let $\psi(t), \psi_1(t), \psi_2(t) \in \{0, 1\}$ be the hard truth values of the atomic predicates in Equation 2, obtained from the bin indicators in Equation 4. The five rule templates are evaluated by Boolean products. For an interval $I = [a, b]$ within the patch,

$$B_{\Box_I \psi} = \prod_{t=a}^{b} \psi(t), \qquad B_{\Diamond_I \psi} = 1 - \prod_{t=a}^{b} [1 - \psi(t)], \tag{17}$$

$$B_{\psi_1 \mathcal{U}_I \psi_2} = 1 - \prod_{t=a}^{b} \left[1 - \psi_2(t) \prod_{u=a}^{t-1} \psi_1(u) \right], \tag{18}$$

where an empty product is one. Let $I_h = \{t \in I : 0 \le t,\ t + h < n\}$ contain the outer positions with a complete inner window. The nested templates are

$$B_{\Box_I \Diamond_{[0,h]} \psi} = \prod_{t \in I_h} \left(1 - \prod_{u=t}^{t+h} [1 - \psi(u)] \right), \tag{19}$$

$$B_{\Diamond_I \Box_{[0,h]} \psi} = 1 - \prod_{t \in I_h} \left(1 - \prod_{u=t}^{t+h} \psi(u) \right), \tag{20}$$

for patch length n and a nonnegative integer h. For a rule ϕ instantiated by source parameters $\theta_{p,r}$, its value B_ϕ is $\mathrm{Sat}(\mathbf{x}^{\mathrm{A}}, \theta_{p,r}; \mathcal{P}_{n,j})$ when evaluated on target patch $\mathcal{P}_{n,j}$. An empty I_h gives truth values one and zero, respectively. The implementation clips learned temporal endpoints to the patch and applies hard masks in the forward pass. Continuous masks and gradient grafting provide gradients during training, and exported rule truth values use the hard computations. For Until, ordered pairs of distinct predicates are enumerated. In multivariate inputs, an Until pair involving a spectral channel is restricted to channels from the same original variable.

Regarding the soft version of the neural networks for these five temporal operators, please refer to TemporalRule Wang et al. (2025).

D Loss Functions in Training objective

For a mini-batch $\mathcal{D}_b$, with $\mathbf{x}^{\mathrm{A}}$ computed from each input $\mathbf{x}$, the classification term in Equation 12 is

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{|\mathcal{D}_b|} \sum_{(\mathbf{x},y)\in\mathcal{D}_b} \log \frac{\exp z_y(\mathbf{x}^{\mathrm{A}})}{\sum_{c=1}^{C} \exp z_c(\mathbf{x}^{\mathrm{A}})}. \tag{21}$$

The temporal clustering term uses the soft assignments in Equation 4:

$$\mathcal{L}_{\mathrm{clu}}^{\mathrm{temp}} = \frac{1}{|\mathcal{D}_b|} \sum_{i,t,d,k} \pi_{t,d,k}^{(i)} \big(x_{t,d}^{\mathrm{A},(i)} - c_{d,k}\big)^2. \tag{22}$$

Let $u_j^{(i)}$ be standardized segment statistic j, $c_{j,k}^{\mathrm{stat}}$ its bin center, and $\alpha_{j,k}^{(i)}$ its soft assignment. The statistical clustering term is

$$\mathcal{L}_{\mathrm{clu}}^{\mathrm{stat}} = \frac{1}{|\mathcal{D}_b|} \sum_{i,j,k} \alpha_{j,k}^{(i)} \big(u_j^{(i)} - c_{j,k}^{\mathrm{stat}}\big)^2. \tag{23}$$

If $\mathcal{I}_1$ indexes the effective first-layer STL intervals, d_j is the span of interval j, and ρ_1 is its minimum target, then

$$\mathcal{L}_{\mathrm{short}}^{(1)} = \frac{1}{|\mathcal{I}_1|} \sum_{j\in\mathcal{I}_1} [\max(0, \rho_1 - d_j)]^2. \tag{24}$$

For continuous global endpoints $a_{p,r}, b_{p,r}$, let K_0 be the number of base-scale candidate patches and R_0 the number of rules per source patch. Then

$$\mathcal{L}_{\mathrm{short}}^{(2)} = \frac{1}{K_0 R_0} \sum_{p=0}^{K_0-1} \sum_{r=1}^{R_0} [\max(0, \rho_2 - b_{p,r} + a_{p,r})]^2, \tag{25}$$

$$\mathcal{L}_{\mathrm{long}}^{(2)} = \frac{1}{K_0 R_0} \sum_{p=0}^{K_0-1} \sum_{r=1}^{R_0} \frac{b_{p,r} - a_{p,r}}{\max(K_0 - 1, 1)}. \tag{26}$$

The two short-span terms discourage degenerate intervals, and the long-span term limits the global search range.

E Additional Experimental Settings

E.1 Dataset and baseline details

Datasets. We use the eight univariate datasets evaluated in TemporalRule (Wang et al., 2025):

- **Epilepsy2** distinguishes seizure from non-seizure segments of single-channel EEG recordings.
- **NerveDamage** uses tibialis anterior EMG segments to distinguish three subjects with different neuromuscular conditions.
- **ECG200** separates normal and myocardial-infarction heartbeats.

- **ECG5000** assigns extracted heartbeats from a long ECG recording to five annotated waveform classes.
- **EOGVerticalSignal** identifies 12 eye-written stroke types from the vertical electrooculogram.
- **SharePriceIncrease** uses 60 daily price-change observations to predict whether the price rises by more than 5% after an earnings announcement.
- **GunPoint** distinguishes drawing a replica gun from pointing with a finger.
- **GunPointOVY** separates earlier and later recordings of the actors across these actions.

The six multivariate tasks are from the UEA archive (Bagnall et al., 2018):

- **ArticularyWordRecognition** classifies spoken words using articulator movement traces.
- **BasicMotions** uses watch accelerometer and gyroscope signals to identify four activities.
- **ERing** identifies finger postures from electric-field measurements.
- **Libras** classifies two-dimensional hand trajectories for Brazilian sign-language movements.
- **PenDigits** classifies handwritten digits from pen coordinates.
- **RacketSports** distinguishes badminton and squash strokes using watch motion sensors.

Baselines.

- **BOP** counts SAX words from sliding windows in a bag-of-pattern representation (Lin et al., 2012), and **SAX–VSM** weighs symbolic words against class-level document profiles (Senin & Malinchik, 2013).
- **LS** learns discriminative shapelets by optimizing a classification objective (Grabocka et al., 2014), and **ST** selects shapelets, then uses their distances to each series as classifier inputs (Hills et al., 2014).
- **RRL** learns conjunctions and disjunctions over input features (Wang et al., 2021), **NSTSC** combines weighted temporal-logic formulae with a tree classifier (Yan et al., 2022), and **TemporalRule** learns hard temporal rules evaluated within fixed patches (Wang et al., 2025).
- **TimeSliver** combines raw and symbolic representations to obtain signed temporal attributions (Pandey et al., 2026), and **InterpGN** gates a shapelet-based logical expert with a deep classifier (Wen et al., 2025), with **InterpGN-SBM** denoting the variant just retaining the logical expert.

E.2 Training and evaluation settings

As TimeSliver does not specify a fixed hyperparameter configuration, we select its hyperparameters on a validation split for each dataset. InterpGN is reproduced using the fixed hyperparameters specified in its original report. For the reproduced univariate TimeSliver and InterpGN results in Table 2, we retain the three runs with the full-model test accuracy for each dataset. InterpGN-SBM uses the same checkpoints selected for InterpGN. The attribution analyses of FlexTempRule and reproduced TemporalRule use the same selection procedure for each dataset, and all attribution metrics use these checkpoints without further selection.

During training, dropout with ratio 0.5 is applied to branch features before the four linear classifiers. For the hyperparameters in gradient grafting, the local sliding temperature is set to 0.75, and the global-range mask temperature is annealed from 1.0 to 0.1, with a pooling temperature of 0.1.

F Additional Experimental Results

This section reports run-to-run variation for the main and ablation classification results, followed by contribution calculations, reconstruction checks, and supplementary analyses of the selected and learned features.

Table 7: Three-run standard deviations (%) on univariate datasets.

Method	Epilepsy2	NerveDamage	ECG200	ECG5000	EOG	SPI	GunPoint	GunPointOVY
TimeSliver	0.11	0.00	3.21	0.14	0.42	0.06	0.67	0.95
InterpGN-SBM	1.25	0.00	2.08	0.69	1.52	0.06	2.34	0.66
InterpGN	0.13	0.00	0.58	0.03	1.27	0.33	0.00	0.00
FlexTempRule	0.09	0.00	1.15	0.06	0.32	0.06	0.39	0.00

Table 8: Three-run standard deviations (%) of FlexTempRule on multivariate datasets.

Method	AWR	BasicMotions	ERing	Libras	PenDigits	RacketSports
FlexTempRule	1.89	0.00	2.12	0.94	0.20	0.31

F.1 Standard deviations of the main results

Tables 7 and 8 report the standard deviations corresponding to the three-run means in Tables 2 and 3, respectively. Each value is the sample standard deviation of the three reported test accuracies, in percentage points: the squared deviations from their mean are summed, divided by two, and then square-rooted. These values describe variation across the reported runs, not uncertainty across test instances. Published baseline entries have no corresponding run-level deviations in this comparison. FlexTempRule has smaller deviations than the reproduced TimeSliver on six univariate datasets and equal deviations on two, which demonstrates the stability of our model, but it has no consistent deviation advantage over InterpGN.

F.2 Standard deviations of the ablation results

Table 9 reports sample standard deviations for the ablated variants in Table 4, using the same calculation as above.

Most ablations have standard deviations below one percentage point. On ECG200, Fixed only and −Statistical have larger deviations (2.31 and 2.00) than the full model (1.15). Statistical only has a smaller deviation (0.58), but its mean accuracy is also lower (89.33% versus 94.33%; Table 4), showing that lower variability alone does not imply better classification performance.

F.3 Feature contributions and category-score reconstruction

A checkpoint is a saved set of trained model parameters, together with the pre-processing configuration used to evaluate it. We load each checkpoint, fix the segment boundaries and normalization statistics, disable dropout, and evaluate the hard rule responses without retraining. Reconstruction refers to recovering the model's category scores from its feature contributions, but it does not reconstruct the checkpoint or the input time series.

For category c, a feature's signed contribution is its truth value multiplied by the corresponding classification weight and, for a sliding branch, the input-dependent gate. Using j to index the concatenated features within each branch, the Fixed, Local, and Global terms are, respectively,

$$\big[\mathbf{W}_{\mathrm{fix}}\big]_{c,j} q_j(\mathbf{x}^{\mathrm{A}}), \qquad \gamma_1(\mathbf{x}^{\mathrm{A}})\big[\mathbf{W}_{\mathrm{loc}}\big]_{c,j} \ell_j(\mathbf{x}^{\mathrm{A}}), \qquad \gamma_2(\mathbf{x}^{\mathrm{A}})\big[\mathbf{W}_{\mathrm{glo}}\big]_{c,j} e_j(\mathbf{x}^{\mathrm{A}}). \tag{27}$$

Here, the weight matrices are those of the corresponding affine classification heads, as in Equation 5. A Statistical contribution is computed in the same way from its hard predicate truth value and class weight, without a gate. Positive and negative terms increase and decrease the category score, respectively. Summing all feature contributions and adding the Fixed and Statistical biases, the Local bias multiplied by γ_1, and the Global bias multiplied by γ_2 recovers Equation 10. Biases are accounted for separately and are not ranked as features.

For instance-level ranking, we take the absolute contributions to the full model's predicted category for that test instance. For dataset-level ranking, we compute each feature's absolute contribution to the model's own predicted category on every training instance, average these values over the training split, and use this fixed ordering for all test instances. The two rankings concern the same pool of features across all four branches. For the top-20% test, the retained count is rounded up to the nearest integer. The truth values of unselected feature responses are set to zero before the original

Table 9: Three-run standard deviations (%) of branch ablated models.

Variant	Epilepsy2	NerveDamage	ECG200	ECG5000	SPI	GunPoint	GunPointOVY
Fixed only	0.22	0.00	2.31	0.06	0.59	0.67	0.00
Statistical only	0.10	0.00	0.58	0.12	0.42	0.77	0.00
−Local	0.11	0.00	1.00	0.14	0.26	0.38	0.00
−Global	0.15	0.00	1.53	0.09	0.47	0.67	0.00
−Statistical	0.54	0.00	2.00	0.12	0.33	0.38	0.00

Table 10: Branch composition (% of selected feature indices) of the training-derived global top-20% contributive features, averaged over three checkpoints.

Branch	Epilepsy2	NerveDamage	ECG200	ECG5000	EOG	SPI	GunPoint	GunPointOVY
Fixed	58.98	70.38	50.82	66.55	66.46	84.72	59.97	54.54
Local	6.56	14.37	11.50	0.27	0.00	0.00	4.96	20.19
Global	7.59	0.00	14.75	0.00	0.00	0.00	12.89	5.01
Statistical	26.87	15.25	22.92	33.18	33.54	15.28	22.18	20.26

classification heads, while the original biases and gates computed from the unmodified input are retained. The selected indices are used for all categories, and the resulting prediction is compared with the full-model prediction. Table 5 reports the percentage of matching test predictions, averaged over three checkpoints for each dataset and model.

For each checkpoint, we compare the reconstructed score with the model's forward output for every category, using training and test instances for FlexTempRule and test instances for TemporalRule. Across 24 frozen checkpoints on the eight univariate datasets for each model, the maximum absolute category-score reconstruction error is 5.34×10^{-5} for FlexTempRule and 3.58×10^{-7} for reproduced TemporalRule models. For TemporalRule, the reconstruction uses its final composite-rule responses, their classification weights, and the classification bias. These checks establish numerical agreement between the contribution sum and the original scores at the evaluated checkpoints. The retention analysis separately measures how well a subset of those features preserves the model's predictions.

F.4 Branch composition of globally selected features

For each checkpoint, we examine the training-derived dataset-level ranking defined in Appendix F.3. Table 10 reports the composition of the top 20% feature indices. Each branch's percentage is the number of selected indices from that branch divided by the total number of selected indices, multiplied by 100, then averaged over the same three checkpoints as Table 5. Fixed, Local, Global, and Statistical denote the four model branches. These percentages count selected features, not their share of the category score, so absolute contributions determine the ranking but are not summed to obtain the percentages.

We can see that Fixed branch accounts for the largest share of selected indices on every dataset, followed by Statistical. Local and Global together take up about a quarter of the selected set on ECG200 and GunPointOVY, but neither appears in the selected set on SPI or EOG. This analysis identifies which branches supply the features retained by the dataset-level ranking and complements the consistency results in Table 5. The proportions also depend on the number of candidate features in each branch, so they do not by themselves establish a branch's necessity or a classification advantage.

F.5 Examples of learned rule features

Table 11 exhibits four features from two correctly classified ECG200 heartbeats. Each of the first three rows defines its own $\phi := \Diamond_I \psi$ using the STL syntax in Equation 2. The fixed and local predicates use the raw view, and the global predicate uses the first-difference view. Each learned temporal interval is relative to the evaluated patch, and each learned numerical predicate is shown at its listed match on the released UCR scale. Since standardization is fitted separately at each position, its original-scale bounds change at other time points. Indices start at zero and patch spans are half-open.

Table 11: Learned temporal logic rules from two ECG200 heartbeats. Numerical predicates are shown at their matched positions, and dashes mark inapplicable patch evaluations.

Case	Branch	Rule and evidence	Patch evaluation
1	Fixed	$\phi := \Diamond_{[1,3]}\psi$, where $\psi := (-1.025 \le \mathbf{x}^{\mathrm{R}} < -0.537)$.	Source $[16, 24)$: true. Target: –.
1	Local	$\phi := \Diamond_{[0,1]}\psi$, where $\psi := (-2.003 \le \mathbf{x}^{\mathrm{R}} < -1.531)$.	Source $[16, 24)$: false. Target $[24, 32)$: true. segment $[16, 32)$
2	Global	$\phi := \Diamond_{[0,1]}\psi$, where $\psi := (0.464 \le \mathbf{x}^{\mathrm{D}})$.	Source $[72, 80)$: false. Target $[40, 48)$: true. Cross-segment

In case 1, the fixed rule marks a negative deflection at a specified position, whereas the local rule finds a deeper deflection one patch later within the same segment despite failing at its source. In case 2, the global rule finds a rise four base patches away from its source, and an auxiliary statistical feature records the level reached in that region (omitted here). Therefore, Local and global sliding retain evidence when a waveform change shifts within or across regions of a heartbeat.

G Complexity analysis

Let R_n be the number of rule templates per valid source patch of length $n \in \{n_0, 2n_0\}$, $\mathcal{K}_n$ the valid source indices, and $\mathcal{J}_n(p)$ the local targets for source p. Let $R_0 = R_{n_0}$ and K_0 count the base-scale candidate patches, as in the global search range in Equation 7. Counting rule–target evaluations, with the fixed truth value reused for the local branch, gives

$$E_{\mathrm{hard}} = \sum_{n \in \{n_0, 2n_0\}} R_n \left[|\mathcal{K}_n| + \sum_{p \in \mathcal{K}_n} (|\mathcal{J}_n(p)| - 1) \right] + \sum_{p=0}^{K_0 - 1} \sum_{r=1}^{R_0} (B_{p,r} - A_{p,r} + 1). \tag{28}$$

The first term calculates fixed and additional local evaluations, and the second counts target patches in learned global ranges. If evaluating template r on an n-point patch costs $c_r(n)$, rule inference is obtained by weighting each term by its template cost. A direct scan of Always and Eventually costs $O(n)$, while direct Until and nested-template scans cost at most $O(n^2)$, with lower cost possible when intermediate reductions are reused. Constructing the spectral view takes $O(VT \log T)$, and predicate assignment takes $O(TDK)$. The five segment statistics take $O(TD)$ apart from median computation, conservatively $O(TD \log T)$ by sorting. For fixed template families, the dominant rule inference cost is $O(E_{\mathrm{hard}} n_{\max}^2)$ as a conservative upper bound, where $n_{\max} = 2n_0$.

During training, the continuous global mask evaluates the full $K_0 \times K_0$ source–target grid, giving $O(K_0^2 R_0 n_0^2)$ conservative rule evaluation. Chunking reduces peak intermediate memory without changing that pair count. With $N_{\mathrm{pred}} = DK$ atomic predicates, the univariate five-template bank has $4N_{\mathrm{pred}} + N_{\mathrm{pred}}(N_{\mathrm{pred}} - 1)$ candidates per source patch, and the multivariate Until restrictions can only reduce this count. Interval parameters of rules, global endpoints, and affine classifier weights require $O((1 + C) \sum_{n \in \{n_0, 2n_0\}} |\mathcal{K}_n| R_n + K_0 R_0)$ storage. The statistical head adds $O(CMDK_{\mathrm{stat}})$ weights for M segments and K_{stat} statistical bins. For hidden gate width H, the two gates add $O(HTD + H)$ parameters. A dense source–target tensor for training requires $O(K_0^2 R_0)$ temporary elements before chunking. These are asymptotic counts, not measured runtime or GPU-memory comparisons.