LISTS: LOCAL INSTANCE SELECTION FOR TIME SERIES CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep time series classification models commonly aggregate temporal representations into a single global representation, which may obscure class-discriminative information contained in local regions. To address this limitation, we propose LISTS, a **L**ocal **I**nstance **S**election framework for **T**ime **S**eries classification. LISTS sequentially selects a small number of local temporal representations that can complement the global context and explicitly retains them alongside the global representation for classification. We formulate this selection process as a sequential decision-making problem and optimize the selector using reinforcement learning with the classification objective as its learning signal. We further provide a theoretical analysis that motivates the proposed global-local formulation. Experiments on all 30 UEA benchmark datasets demonstrate strong classification performance, with LISTS achieving 74.4% average accuracy and the best average rank among ten methods, while a controlled synthetic study verifies that LISTS identifies class-informative local regions.

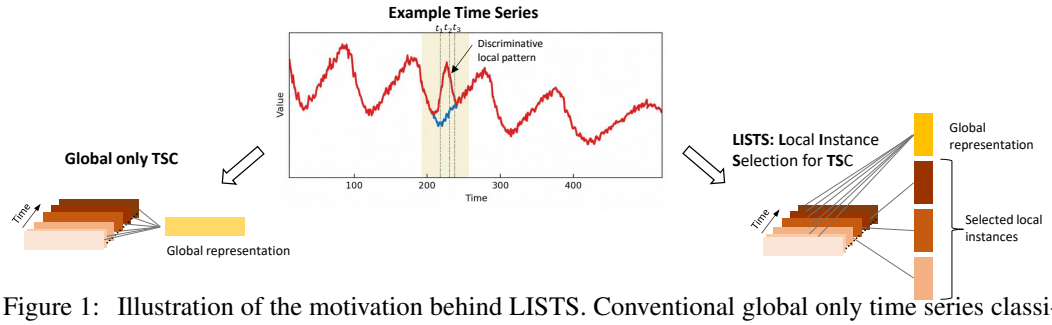

Figure 1: Illustration of the motivation behind LISTS. Conventional global only time series classification aggregates the entire sequence into a global representation, which may dilute discriminative local patterns. LISTS explicitly selects informative local instances and combines them with the global representation for classification.

1 INTRODUCTION

Time series classification (TSC) is a fundamental task in temporal data analysis, with applications spanning healthcare (Fan et al., 2026), industrial monitoring (Farahani et al., 2025), human activity recognition (Xie et al., 2025), and many other domains (Jeong et al., 2026). Modern deep TSC models commonly encode an input sequence into a sequence of temporal representations and subsequently aggregate them into a fixed-dimensional representation for classification (Foumani et al., 2024a; Jeong et al., 2026). We refer to the temporal representation at each position as a local instance. A widely used aggregation mechanism is global average pooling (GAP), which summarizes local instances across the entire sequence into a global representation. This representation provides useful overall context, but compressing a sequence of local instances into a single vector may not fully preserve information in individual local instances (Lee et al., 2021; Chen et al., 2024). This loss can be particularly consequential when class-discriminative evidence is concentrated in a local region.

The importance of local instances has been recognized in the multiple-instance learning (MIL) literature, where instance-based (Wang et al., 2018), attention-based (Ilse et al., 2018), and additive (Javed et al., 2022) aggregation mechanisms have been developed to emphasize informative in-

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.

stances. More recently, these ideas have been extended to time series classification, demonstrating the benefit of exploiting informative temporal instances (Early et al., 2024). These approaches primarily reconsider how to aggregate local representations into a sequence-level prediction. On the other hand, a complementary question arises: rather than replacing the global representation with an alternative aggregation of local instances, can a small number of local instances provide additional class evidence alongside the global context?

In this work, we consider a global-local representation in which the global context captures sequence-level information while selected local instances retain complementary temporal evidence. To address such a sparse selection problem, we propose **LISTS**, a sample-adaptive **L**ocal **I**nstance **S**election method for **T**ime **S**eries classification. LISTS sequentially selects k temporal representations from the encoder output while conditioning each selection on both the global context and the previously selected local evidence. We formulate local instance selection as a sequential decision-making problem and learn the selector using reinforcement learning. Specifically, the selector defines a policy over candidate temporal positions, and we optimize it with REINFORCE using the classification likelihood as the learning signal. We then concatenate the selected local representations with the global context for classification, preserving their information rather than compressing them into a shared representation. We further provide a theoretical analysis of the proposed global-local formulation showing that local evidence is useful for classification precisely when it contains class information beyond the global context, and that further compressing the global and local evidence can discard label-relevant information. We also show that the classification objective provides a tractable learning signal for selecting such complementary local evidence.

We evaluate LISTS from two complementary perspectives. First, experiments on the UEA time series classification benchmark (Bagnall et al., 2018) evaluate its performance against competing methods and examine its key design choices through ablation studies. Across the benchmark, LISTS achieves the highest average accuracy and the best average rank among the compared methods. Second, we introduce a controlled synthetic *Sleep* dataset in which the class-discriminative local region is known for every sample. This setting allows us to directly assess whether LISTS selects instances within the truly discriminative regions.

Our contributions are threefold:

- We introduce **LISTS**, a framework that sequentially selects a small number of local temporal instances based on the global context and selection history, and retains them alongside the global representation for classification.
- We provide an information-theoretic analysis that characterizes when local instances provide complementary class information beyond the global representation and motivates the proposed selection and preservation strategy.
- We evaluate LISTS on the UEA time series classification benchmark and a controlled synthetic *Sleep* dataset with known discriminative regions, allowing us to assess both classification performance and the localization of class-informative instances.

2 Related Work

Time Series Classification and Instance-Aware Aggregation. Time series classification (TSC) has been extensively studied using both non-deep-learning and deep learning approaches (Middlehurst et al., 2024; Wang et al., 2026). Strong non-deep methods include ROCKET (Dempster et al., 2020) and DrCIF (Middlehurst et al., 2021), while deep architectures such as FCN, ResNet (Wang et al., 2017), and InceptionTime (Ismail Fawaz et al., 2020) have also achieved competitive performance. Many deep TSC models encode an input sequence into temporal representations and then aggregate them into a fixed-dimensional sequence-level representation for classification, commonly using Global Average Pooling (GAP). While GAP provides a simple and effective summary of the entire sequence, compressing all temporal representations into a single global representation may attenuate class-discriminative information concentrated in a small number of local regions (Lee et al., 2021; Chen et al., 2024). This limitation motivates approaches that explicitly account for the contributions of individual temporal instances.

The importance of informative instances has been extensively studied in MIL. Instance-based (Wang et al., 2018), attention-based (Ilse et al., 2018), and additive (Javed et al., 2022) MIL methods as-

sign different importance to instances when constructing bag-level predictions. Researchers have recently adapted these ideas to TSC. MILLET (Early et al., 2024) introduces MIL aggregation mechanisms into deep TSC models to obtain sequence-level predictions while providing instance-level explanations. TimeMIL (Chen et al., 2024) further formulates multivariate TSC as a weakly supervised MIL problem and develops time-aware pooling to capture informative temporal patterns and their dependencies. These methods use instance-level evidence to construct the sequence-level representation or prediction itself. Our focus is different: given a global representation that has already been formed, we ask which local instances contain class information that remains complementary to this global context. Thus, the distinction is not simply whether local instances are informative, but whether they provide additional class evidence beyond what the global context already captures.

Global-Local Modeling for TSC. A related line of work jointly captures local patterns and global dependencies through dedicated representation-learning architectures. DA-Net (Chen et al., 2022) combines window-based attention for local patterns with sparse self-attention for global dependencies, while DSDCLNet (Liu et al., 2024) employs a dual-stream architecture to model multiscale local features and global dependencies. MF-Net (Du et al., 2023) additionally incorporates cross-variable information, and MPTSNet (Mu et al., 2025) models local patterns and global dependencies across multiple periodic scales. These methods highlight the importance of jointly exploiting local and global structure, focusing on how to encode and integrate this information into a unified representation. In contrast, LISTS studies selective retention of local evidence that complements an already-formed global representation.

Discrete Temporal Selection. Discrete temporal selection remains relatively underexplored in general TSC. RLPAM (Gao et al., 2022) employs reinforcement learning to discover discriminative temporal patterns for TSC, while RELEVANT (Pantiskas et al., 2023) dynamically selects input channels over time to retain task-relevant information. More recently, INSHAPE (Lee et al., 2026) identifies sample-specific discriminative temporal segments through discrete instance-level shapelet selection. These methods explore sparse selection at the level of temporal patterns, input channels, or raw temporal segments. In contrast, LISTS performs discrete selection directly over learned local instances produced by a TSC encoder.

3 METHODOLOGY

This section introduces the proposed LISTS framework for TSC. First, we formulate the TSC problem and define the temporal representations and local instances used throughout the method in Sec. 3.1. Second, we quantify the complementary class information provided by local instances beyond the global context in Sec. 3.2. Third, we show why local evidence should be preserved without compression and introduce its concatenation with the global context in Sec. 3.3. Last, we present the sequential reinforcement-learning selector and its training objective in Sec. 3.4. Fig. 2 illustrates the overall architecture of LISTS.

3.1 PROBLEM FORMULATION

Given a time series dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, each input $x_i \in \mathbb{R}^{T\times C}$ contains T time points over C channels, and $y_i \in \{1, \dots, K\}$ denotes its class label. The goal of TSC is to learn a mapping $f : \mathbb{R}^{T\times C} \to \{1, \dots, K\}$ from feature space to label space.

Existing deep learning TSC architectures mainly consist of two modules, an encoder ϕ_E and a classifier ϕ_C. For each input time series x_i, ϕ_E produces a D-dimensional temporal representation $r_i = [r_i^1, \dots, r_i^T] \in \mathbb{R}^{T\times D}$, where $r_i^t \in \mathbb{R}^D$ denotes a local instance at temporal position t.

3.2 WHEN ARE LOCAL INSTANCES USEFUL?

A common way to consolidate these instances is Global Average Pooling (GAP). This produces a global context g_i, which is then passed to ϕ_C for classification.

$$g_i = \frac{1}{T}\sum_{t=1}^{T} r_i^t, \qquad \phi_C(g_i) = \hat{y}_i. \tag{1}$$

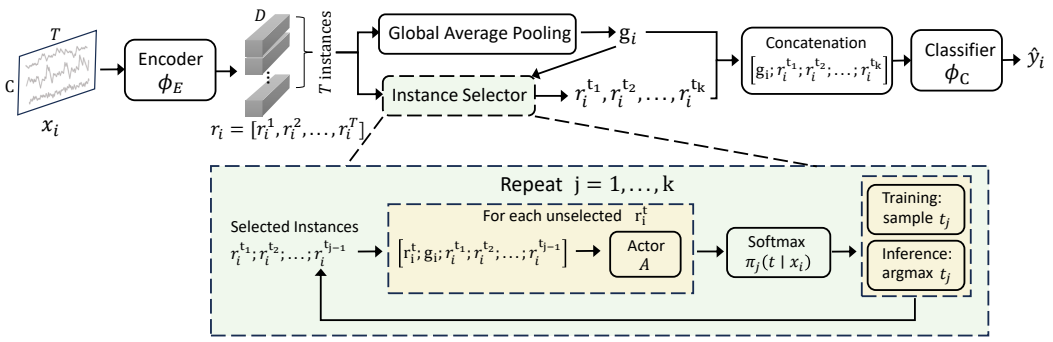

Figure 2: Overview of the proposed framework LISTS. The encoder produces temporal representations, from which global average pooling extracts the global context. The selector sequentially selects local instances conditioned on the global context and previously selected instances. The selected local evidence is concatenated with the global context for classification.

Since g_i summarizes all local instances into a single global context, we ask whether some local instances can still provide additional class information beyond it.

For the theoretical analysis, we drop the sample index and write G for the global context and Y for the class label. We use L to denote an arbitrary subset of local instances, referred to as local evidence. To quantify the classification value of an observation Z, we use the Bayes log-risk $\mathcal{R}^*(Z) = \inf_q \mathbb{E}[-\log q(Y \mid Z)]$, where the infimum is taken over measurable Markov kernels. We use $I(Y; L \mid G)$ to denote the conditional mutual information between Y and L given G. The full assumptions and proofs are provided in Appendix A.

Proposition 1 (Complementary information)**.** *Under the assumptions of Appendix A,*

$$\mathcal{R}^*(G) - \mathcal{R}^*(G, L) = I(Y; L \mid G), \tag{2}$$

and $\mathcal{R}^*(G, L) < \mathcal{R}^*(G)$ holds exactly when Y and L are dependent given G.

Therefore, the class information in the local evidence L that is complementary to G can be written as

$$I(Y; L \mid G) = \mathbb{E}_{G,L}\Big[D_{\mathrm{KL}}\big(P(Y \mid G, L) \,\|\, P(Y \mid G)\big)\Big], \tag{3}$$

where $D_{\mathrm{KL}}(P\|Q)$ denotes the KL divergence of P from Q. This shows that a subset of local instances can contain class information beyond the global context and motivates retaining such complementary local evidence alongside the global context.

3.3 Preserving Local Evidence

Local evidence L and the global context G can either be compressed into a representation $\psi(G, L)$ or preserved separately through concatenation.

Proposition 2 (Preservation)**.** *Under log loss,*

$$\mathcal{R}^*\big(\psi(G, L)\big) \;\geq\; \mathcal{R}^*\big([G; L]\big) \tag{4}$$

for any measurable ψ, with equality exactly when $\psi(G, L)$ is sufficient for Y. The Bayes-risk inequality holds for every lower-bounded loss and every action space.

Proposition 2 shows that compressing (G, L) into $\psi(G, L)$ can discard label-relevant information. The information-preserving alternative is therefore to retain $[G; L]$ without compressing either component. However, the dimension of the concatenated representation grows with the number of retained local instances. Retaining k local instances produces a representation of dimension $(k+1)D$, increasing classifier parameters as k grows. This increases model complexity and can make parameter estimation and optimization harder, particularly with limited training data. We therefore retain only a small subset of the local instances that best complement the global context.

3.4 LOCAL INSTANCE SELECTION

We use a discrete reinforcement-learning policy π to sequentially select local instances conditioned on the global context and previously selected instances, yielding the local evidence L_π (Williams, 1992). At selection step j, let $\mathcal{H}_{i,j-1} = (r_i^{t_1}, \ldots, r_i^{t_{j-1}})$ denote the history of previously selected local instances. The actor A assigns each unselected temporal representation a selection logit,

$$a_{i,j}^t = A\big(\mathrm{sg}(r_i^t); \mathrm{sg}(g_i); \mathrm{sg}(\mathcal{H}_{i,j-1})\big), \tag{5}$$

$$\pi_j(t \mid x_i) = \frac{\exp(a_{i,j}^t)}{\sum_{s \notin \{t_1,\ldots,t_{j-1}\}} \exp(a_{i,j}^s)}, \tag{6}$$

where $\pi_j(t \mid x_i)$ is the probability of selecting position t at step j, and sg leaves its input unchanged in the forward computation while blocking its gradient.

During training, the next position is sampled from the conditional policy $t_j \sim \pi_j(\cdot \mid x_i)$, whereas inference uses deterministic decoding $t_j = \arg\max_t \pi_j(t \mid x_i)$.

For sample i, the selected positions are retained in selection order as $\mathcal{S}_i = (t_1, \ldots, t_k)$, yielding the selected representations $(r_i^{t_1}, \ldots, r_i^{t_k})$. We concatenate their representations with the global context $z_i = \big[g_i; r_i^{t_1}; \ldots; r_i^{t_k}\big]$ and use a single linear layer to produce the class logits,

$$\mathrm{logits}_i = W_g\, g_i + \sum_{j=1}^{k} W_j\, r_i^{t_j} + b. \tag{7}$$

Section 3.2 motivates selecting local evidence with large complementary information $I(Y; L_\pi \mid G)$. However, directly optimizing this quantity is not feasible because it depends on the unknown population distribution. We therefore use the classification log-likelihood as the reward signal for the selector. The classifier defines the class probability

$$q_{\phi_C}(y \mid g_i, r_i^{t_1}, \ldots, r_i^{t_k}) = \frac{\exp\big(\mathrm{logits}_i(y)\big)}{\sum_{m=1}^{K} \exp\big(\mathrm{logits}_i(m)\big)}. \tag{8}$$

Its population objective is defined as $\mathcal{J}(\pi, \phi_C) = \mathbb{E}\left[\log q_{\phi_C}(Y \mid G, L_\pi)\right]$. We further define the classifier mismatch as $\Delta_{\phi_C}(\pi) = \mathbb{E}\left[D_{\mathrm{KL}}\left(P_{Y|G,L_\pi} \| q_{\phi_C}\right)\right]$.

Proposition 3 (Variational bound)**.** *Under the assumptions of Appendix A and integrability of* $\log q_{\phi_C}(Y \mid G, L_\pi)$*,*

$$\mathcal{J}(\pi, \phi_C) + H(Y \mid G) = I(Y; L_\pi \mid G) - \Delta_{\phi_C}(\pi) \le I(Y; L_\pi \mid G), \tag{9}$$

with equality when q_{ϕ_C} *is the Bayes predictor given* (G, L_π)*.*

Proposition 3 shows that, for a fixed encoder, maximizing $\mathcal{J}(\pi, \phi_C)$ with respect to the selector is equivalent to maximizing $I(Y; L_\pi \mid G) - \Delta_{\phi_C}(\pi)$, since $H(Y \mid G)$ is independent of π. For a fixed selector, maximizing $\mathcal{J}(\pi, \phi_C)$ with respect to the classifier is equivalent to minimizing the classifier mismatch $\Delta_{\phi_C}(\pi)$.

Therefore, the classification log-likelihood provides a tractable objective that encourages the selection of class-informative local evidence and trains the model to better predict the corresponding class distribution. Accordingly, we use the same classification signal throughout training. The encoder and classifier are trained with the classification loss

$$\mathcal{L}_{\mathrm{cls}} = -\frac{1}{B} \sum_{i=1}^{B} \log q_{\phi_C}\left(y_i \mid g_i, r_i^{t_1}, \ldots, r_i^{t_k}\right), \tag{10}$$

where B is the batch size.

For the selector, the corresponding sample reward is $R_i = \log q_{\phi_C}(y_i \mid g_i, r_i^{t_1}, \ldots, r_i^{t_k})$. The mean reward of a batch is $\bar{R} = \frac{1}{B}\sum_{i=1}^{B} R_i$, which is used as a baseline reward for all samples in the batch. The probability assigned by the policy π to the sampled sequence is $\pi(\mathcal{S}_i \mid x_i) = \prod_{j=1}^{k} \pi_j(t_j \mid x_i)$. Using this reward and the sequence probability, we train the selector with the REINFORCE objective (Williams, 1992)

$$\mathcal{L}_{\mathrm{RL}} = -\frac{1}{B} \sum_{i=1}^{B} \big(R_i - \bar{R}\big) \log \pi(\mathcal{S}_i \mid x_i). \tag{11}$$

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate LISTS on all 30 datasets from the UEA multivariate time series classification archive. The archive was released in 2018 and covers a wide range of applications, numbers of channels, and sequence lengths. We use the official training and test splits for all datasets. Detailed dataset statistics are provided in Appendix D.1.

Compared Methods. We compare LISTS with a broad set of baselines, including the non-deep learning methods DTW-1NN-D (Bagnall et al., 2018) and MiniROCKET (Dempster et al., 2021), representative deep TSC methods MLSTM-FCN (Karim et al., 2019), TapNet (Zhang et al., 2020), InceptionTime (Ismail Fawaz et al., 2020), and ConvTran (Foumani et al., 2024b), the related methods TimeMIL (Chen et al., 2024) and MPTSNet (Mu et al., 2025), and the recent method TimeSliver (Pandey et al., 2026). For the established baselines DTW-1NN-D, MLSTM-FCN, and TapNet, we use published results reported in TapNet (Zhang et al., 2020), while the InceptionTime results are taken from ConvTran (Foumani et al., 2024b). The remaining methods are reproduced using their released implementations and default settings. For ConvTran, TimeMIL, MPTSNet, and TimeSliver, we preserve their original criterion of selecting checkpoints based on accuracy by constructing a 20% split from the training data as a validation set and selecting the checkpoint with the highest validation accuracy. For each random seed, the same validation split is used across these methods. All reproduced methods are evaluated over three independent runs with random seeds 0, 1, and 42.

Implementation Details. We use InceptionTime as the backbone with an output dimension of 128 and select $k = 3$ local instances for each input. We denote this default configuration as LISTS-3 and use it throughout the main experiments, ablation study, and case study. We set $\lambda_{\mathrm{RL}} = 1.0$ and optimize the model using Adam with a learning rate of 10^{-3}. Models are trained for 600 epochs. For two extremely small datasets, AtrialFibrillation and StandWalkJump, we train for 100 epochs. We select the checkpoint with the lowest training loss. The official test split is used only for final evaluation. We perform three independent runs with random seeds 0, 1, and 42. Additional architecture and optimization details are provided in Appendix B.

Metrics. We report classification accuracy on each dataset for both established baselines and reproduced methods. For reproduced methods, we report the mean accuracy across three runs. We further report Average Accuracy, Average Rank, and the number of dataset-wise best results over the full benchmark, together with pairwise wins, draws, and losses between LISTS and each baseline. Standard deviations across runs are reported in Appendix C.

4.2 MAIN RESULTS

We obtain a 74.4% average accuracy across all 30 UEA datasets, achieving the best performance on 11 datasets and an average rank of 3.57 among 10 methods. In particular, LISTS outperforms the strongest competing method, MiniROCKET, by 2.6 percentage points in average accuracy. The gains are particularly pronounced on several challenging datasets. Compared with the best competing method on each dataset, LISTS improves accuracy by 8.0, 6.7, 4.5, and 4.0 percentage points on Handwriting, StandWalkJump, HandMovementDirection, and LSST, respectively.

4.3 ABLATION STUDY

Table 2 summarizes the ablation results on the 30 UEA datasets. The full LISTS-3 model achieves an average accuracy of 74.4%. We first examine the architectural design and observe that both global and local information are important for classification. Using only the local instance branch reduces the accuracy to 69.1%, while relying only on the global branch achieves 71.6%, supporting that the two sources provide complementary information. We also observe consistent degradation when replacing concatenation with summation or removing positional encoding, with accuracy dropping by 2.1 and 1.9 percentage points, respectively.

We next study the selector design. Removing either global or history conditioning reduces performance by 0.9 and 1.4 percentage points, suggesting that instance selection benefits from both the overall global context and previously selected evidence. We further compare the proposed selector

Table 1: Comparison of classification accuracy on the 30 UEA datasets. The best result on each dataset is shown in **bold**, and the second-best is underlined.

Dataset	DTW-1NN-D	MiniROCKET KDD'21	MLSTM-FCN Neur. Netw.'19	TapNet AAAI'20	InceptionTime DMKD'20	ConvTran DMKD'24	TimeMIL ICML'24	MPTSNet AAAI'25	TimeSliver ICLR'26	**LISTS-3**
ArticularyWordRecognition	0.987	**0.993**	0.973	0.987	0.987	0.969	0.936	0.958	0.978	0.967
AtrialFibrillation	0.200	0.089	0.267	0.333	0.200	0.267	0.244	**0.400**	0.333	0.378
BasicMotions	0.975	**1.000**	0.950	**1.000**	**1.000**	**1.000**	0.983	0.875	**1.000**	**1.000**
CharacterTrajectories	0.990	0.943	0.985	**0.997**	0.988	0.987	0.984	0.968	0.968	0.993
Cricket	**1.000**	0.986	0.917	0.958	0.986	0.977	0.935	0.884	0.935	0.958
DuckDuckGeese	0.600	0.687	0.675	0.575	0.360	0.580	0.587	0.513	0.420	**0.700**
EigenWorms	0.618	**0.944**	0.504	0.489	0.527	0.794	0.723	0.466	0.733	0.919
Epilepsy	0.964	**1.000**	0.761	0.971	0.993	0.964	0.952	0.877	0.961	0.966
ERing	0.133	**0.978**	0.133	0.133	0.930	0.802	0.770	0.867	0.441	0.889
EthanolConcentration	0.323	**0.449**	0.373	0.323	0.349	0.265	0.283	0.376	0.340	0.251
FaceDetection	0.529	0.578	0.545	0.556	0.589	0.653	0.545	**0.671**	0.624	0.650
FingerMovements	0.530	0.553	**0.580**	0.530	0.560	0.497	0.527	0.497	0.507	0.543
HandMovementDirection	0.231	0.333	0.365	0.378	0.378	0.360	0.266	0.455	0.270	**0.500**
Handwriting	0.607	0.524	0.286	0.357	0.301	0.212	0.194	0.245	0.211	**0.687**
Heartbeat	0.717	0.740	0.663	0.751	0.625	**0.759**	0.676	0.727	0.686	0.727
InsectWingbeat	0.115	0.355	0.167	0.208	0.696	0.709	**0.710**	0.641	0.570	0.683
JapaneseVowels	0.949	0.959	0.976	0.965	0.970	0.985	0.950	0.925	0.852	**0.986**
Libras	0.872	0.909	0.856	0.850	0.872	0.835	0.719	0.741	0.437	**0.913**
LSST	0.551	0.641	0.373	0.568	0.446	0.627	0.639	0.569	0.621	**0.681**
MotorImagery	0.500	0.493	0.510	**0.590**	0.530	0.463	0.477	0.523	0.517	0.543
NATOPS	0.883	0.909	0.889	0.939	0.917	0.941	0.924	0.820	0.869	**0.969**
PEMS-SF	0.711	0.821	0.699	0.751	**0.890**	0.709	0.204	0.838	0.803	0.813
PenDigits	0.977	0.964	0.978	0.980	0.980	0.982	0.943	0.974	0.976	**0.986**
PhonemeSpectra	0.151	**0.313**	0.110	0.175	0.159	0.267	0.287	0.102	0.159	0.184
RacketSports	0.803	0.855	0.803	**0.868**	0.822	0.829	0.833	0.820	**0.868**	0.789
SelfRegulationSCP1	0.775	**0.903**	0.874	0.652	0.863	0.895	0.775	0.871	0.782	0.757
SelfRegulationSCP2	0.539	0.511	0.472	**0.550**	0.472	0.546	0.522	0.507	0.513	0.513
SpokenArabicDigits	0.963	0.805	0.990	0.983	0.987	0.993	0.990	0.985	0.987	**0.995**
StandWalkJump	0.200	0.378	0.067	0.400	0.400	0.356	0.378	0.333	0.444	**0.511**
UWaveGestureLibrary	0.903	**0.917**	0.891	0.894	0.909	0.861	0.832	0.797	0.580	0.860
Average Accuracy (↑)	0.643	0.718	0.621	0.657	0.689	0.703	0.660	0.674	0.646	**0.744**
# Best (↑)	1	9	1	5	2	2	1	2	2	**11**
Average Rank (↓)	6.450	4.217	6.683	4.850	4.733	4.800	6.683	6.683	6.333	**3.567**
Ours Wins	23	16	24	19	18	18	24	23	24	–
Ours Draws	0	1	0	2	1	1	0	1	2	–
Ours Losses	7	13	6	9	11	11	6	6	4	–

Table 2: Ablation results on the 30 UEA datasets. Parentheses indicate decreases (pp) from LISTS-3 (0.744).

Architecture		***Selector Design***	
Local Only	0.691 (-5.3)	w/o Global Cond.	0.735 (-0.9)
Global Only	0.716 (-2.8)	w/o History Cond.	0.730 (-1.4)
Sum Fusion	0.723 (-2.1)	Random-3	0.720 (-2.4)
w/o PE	0.725 (-1.9)	Attention-3	0.731 (-1.3)

Table 3: Sensitivity to the number of selected local instances k.

k	**Average Accuracy**	**Latency (ms/sample)**	**Relative Latency**
1	0.727	9.24	1.00×
3	**0.744**	11.83	1.28×
5	0.742	13.84	1.50×

with two alternative selection strategies. We denote random selection of three local instances as Random-3, while Attention-3 replaces the reinforcement-learning selector with an attention-based selector while keeping the remaining design unchanged. Random-3 achieves an average accuracy of 72.0%, while Attention-3 improves it to 73.1%. However, Attention-3 still remains 1.3 percentage points below LISTS-3, indicating that the reinforcement-learning formulation provides additional benefit beyond learned selection. Further details of Attention-3 are provided in Appendix B.3.

4.4 Sensitivity to the Number of Selected Instances

Table 3 studies the sensitivity of LISTS to the number of selected local instances k. Increasing k from 1 to 3 improves the average accuracy by 1.7 percentage points, suggesting that multiple selected local instances provide additional useful information for classification. Further increasing k to 5 does not yield additional improvement. Inference latency also increases with k, with $k = 3$ requiring 1.28× the latency of $k = 1$ and $k = 5$ increasing this to 1.50×. These results suggest that $k = 3$ provides a favorable balance between classification performance and inference efficiency, and we therefore use LISTS-3 as the default configuration.

5 CASE STUDY

5.1 SYNTHETIC SLEEP DATASET

We construct a synthetic dataset named *Sleep* with explicitly defined class-discriminative local regions. Each time series represents an eight-hour synthetic sleep slow-wave activity (SWA) pattern, containing $T = 480$ observations sampled at one-minute intervals and consisting of five cycles. A decreasing amplitude trend is designed to reflect the overnight decline in SWA. Inspired by the synthetic signatures proposed in (Goswami et al., 2023), we construct nine classes, including one normal class and eight signature classes. For each signature class, a class-specific signature is randomly injected into each sample with a duration between 17 and 137 time points. The injected region is treated as the ground-truth discriminative region for that sample.

This design allows us to evaluate the ability of LISTS to capture class-informative local evidence by measuring whether the selected positions fall within the ground-truth region. The training and test splits each contain 50 samples per class, resulting in 450 time series per split. Further details of the dataset construction are provided in Appendix D.2.

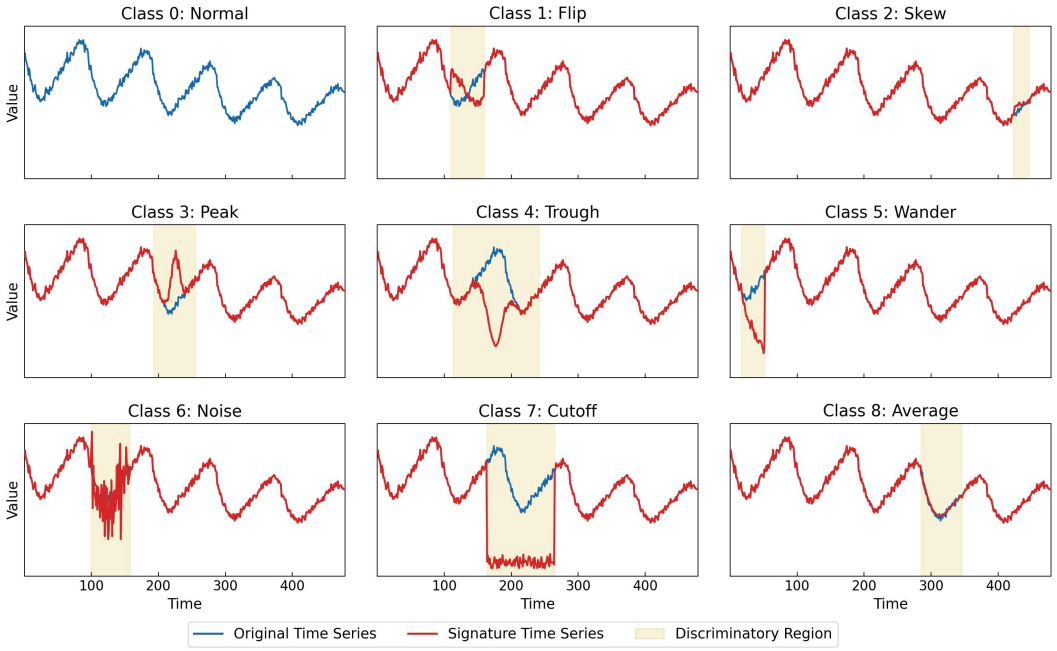

Figure 3: Examples of the nine classes in the synthetic *Sleep* dataset. Shaded areas indicate the injected ground-truth discriminative regions.

5.2 *Sleep* RESULTS

We first evaluate classification performance on the synthetic *Sleep* dataset. Global Only achieves an accuracy of 87.3%, while Random-3 improves this to 88.7%. Attention-3 further increases the accuracy to 91.1%, and LISTS-3 achieves the best performance at 93.1%. These results show that learned local instance selection improves classification over both the global representation alone and random selection.

We further evaluate whether the selected local instances correspond to the ground-truth discriminative regions. Since the normal class contains no injected signature, we evaluate on the 400 test samples from the eight signature classes. We use two selection metrics. Any-Hit@3 measures whether at least one of the three selected positions falls within the ground-truth region, while All-Hit@3 measures whether all three selected positions fall within the region. We compare against Random-3 and Attention-3.

Table 4: Any-Hit@3 and All-Hit@3 localization performance on the synthetic *Sleep* dataset.

Metric	Method	Flip	Skew	Peak	Trough	Wander	Noise	Cutoff	Average	Overall
Any-Hit@3	Random-3	30%	52%	42%	30%	40%	40%	38%	44%	39.50%
	Attention-3	84%	82%	**98%**	**100%**	**96%**	**100%**	62%	**98%**	90.00%
	LISTS-3	**98%**	**92%**	**98%**	98%	76%	**100%**	**100%**	**98%**	**95.00%**
All-Hit@3	Random-3	0%	0%	0%	0%	0%	2%	0%	0%	0.25%
	Attention-3	40%	60%	50%	90%	42%	0%	8%	14%	38.00%
	LISTS-3	**88%**	**88%**	**98%**	**98%**	**66%**	**100%**	**54%**	**96%**	**86%**

Table 4 reports the localization results. LISTS-3 substantially outperforms Random-3 under both metrics, increasing the overall Any-Hit@3 from 39.50% to 95.00% and All-Hit@3 from 0.25% to 86.00%. Compared with Attention-3, LISTS-3 further improves the overall Any-Hit@3 from 90.00% to 95.00%, with a much larger gain in All-Hit@3 from 38.00% to 86.00%. LISTS-3 also achieves consistently strong localization across the eight signature classes, with particularly high All-Hit@3 scores on Noise, Peak, Trough, and Average. These results demonstrate that the instances selected by LISTS-3 closely align with the ground-truth discriminative regions, providing direct evidence that the selector captures class-relevant local patterns.

6 Conclusion

The LISTS framework presented in this work introduces sample-adaptive local instance selection for time series classification. Our theoretical analysis characterizes when local evidence provides additional class information beyond the global representation and motivates preserving global and local information without further compression. LISTS realizes this formulation through a sequential reinforcement-learning selector and concatenates the selected local evidence with the global representation for classification. Its value is demonstrated across all 30 UEA time series classification datasets and the synthetic *Sleep* dataset proposed in this work. We believe this work provides a foundation for further investigation of selective global-local representations for TSC and facilitates several directions for future work:

Adaptive selection budgets: In this work, LISTS selects a fixed number k of local instances for every input. However, the amount of useful local evidence may vary across samples and datasets. Future work could allow the selector to determine the number of instances dynamically, for example by introducing a stopping action or an input-dependent selection budget. This could reduce unnecessary selections for some samples.

Application to other models: We demonstrate LISTS using InceptionTime as the temporal encoder. The proposed global-local formulation is not inherently restricted to this backbone, and future work could investigate its use with other TSC architectures that produce temporal representations, including other convolutional, recurrent models, and Transformer-based models. This would help determine how the benefits of local instance selection depend on the underlying representation architecture.

Alternative selection strategies: LISTS formulates local instance selection as a sequential reinforcement-learning problem and trains the selector using REINFORCE. Beyond the attention-based alternative examined in our ablation study, future work could explore other discrete or differentiable selection mechanisms and more advanced reinforcement-learning objectives. These approaches may improve optimization efficiency and overall predictive performance in local evidence selection.

AI Use Statement

We used ChatGPT to polish the paper and retrieve some relevant literature for the Related Work section. We used Claude to assist with mathematical proofs and code debugging. All AI-assisted content was reviewed and verified by the authors. The authors made all final research decisions and take full responsibility for the correctness of the submitted work.

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.

David Barber and Felix Agakov. Information maximization in noisy channels: A variational approach. In *Advances in Neural Information Processing Systems*, volume 16, 2003.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24(2):265–272, 1953. doi: 10.1214/aoms/1177729032.

Rongjun Chen, Xuanhui Yan, Shiping Wang, and Guobao Xiao. DA-Net: Dual-attention network for multivariate time series classification. *Information Sciences*, 610:472–487, 2022. doi: 10.1016/j.ins.2022.07.178.

Xiwen Chen, Peijie Qiu, Wenhui Zhu, Huayu Li, Hao Wang, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. TimeMIL: Advancing multivariate time series classification via a time-aware multiple instance learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7190–7206. PMLR, 2024.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 2nd edition, 2006. doi: 10.1002/047174882X.

Angus Dempster, François Petitjean, and Geoffrey I. Webb. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020. doi: 10.1007/s10618-020-00701-z.

Angus Dempster, Daniel F. Schmidt, and Geoffrey I. Webb. MiniRocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 248–257, 2021. doi: 10.1145/3447548.3467231.

Mingsen Du, Yanxuan Wei, Xiangwei Zheng, and Cun Ji. Multi-feature based network for multivariate time series classification. *Information Sciences*, 639:119009, 2023. doi: 10.1016/j.ins.2023.119009.

Joseph Early, Gavin Cheung, Kurt Cutajar, Hanting Xie, Jas Kandola, and Niall Twomey. Inherently interpretable time series classification via multiple instance learning. In *International Conference on Learning Representations*, 2024.

Wei Fan, Jingru Fei, Jindong Han, Jie Lian, Hangting Ye, Xiaozhuang Song, Xin Lv, Kun Yi, and Min Li. MedViA: Empowering medical time series classification with vision augmentation and multimodal fusion. *Information Fusion*, 127:103659, 2026. doi: 10.1016/j.inffus.2025.103659.

Mojtaba A. Farahani, M. R. McCormick, Ramy Harik, and Thorsten Wuest. Time-series classification in smart manufacturing systems: An experimental evaluation of state-of-the-art machine learning algorithms. *Robotics and Computer-Integrated Manufacturing*, 91:102839, 2025. doi: 10.1016/j.rcim.2024.102839.

Navid Mohammadi Foumani, Lynn Miller, Chang Wei Tan, Geoffrey I. Webb, Germain Forestier, and Mahsa Salehi. Deep learning for time series classification and extrinsic regression: A current survey. *ACM Computing Surveys*, 56(9):1–45, 2024a. doi: 10.1145/3649448.

Navid Mohammadi Foumani, Chang Wei Tan, Geoffrey I. Webb, and Mahsa Salehi. Improving position encoding of transformers for multivariate time series classification. *Data Mining and Knowledge Discovery*, 38:22–48, 2024b. doi: 10.1007/s10618-023-00948-2.

Ge Gao, Qitong Gao, Xi Yang, Miroslav Pajic, and Min Chi. A reinforcement learning-informed pattern mining framework for multivariate time series classification. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2994–3000, 2022. doi: 10.24963/ijcai.2022/415.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Mononito Goswami, Cristian Ignacio Challu, Laurent Callot, Lenon Minorics, and Andrey Kan. Unsupervised model selection for time-series anomaly detection. In *International Conference on Learning Representations*, 2023.

Peter D. Grünwald and A. Philip Dawid. Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory. *The Annals of Statistics*, 32(4):1367–1433, 2004. doi: 10.1214/009053604000000553.

Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2127–2136. PMLR, 2018.

Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller, and François Petitjean. InceptionTime: Finding AlexNet for time series classification. *Data Mining and Knowledge Discovery*, 34(6):1936–1962, 2020. doi: 10.1007/s10618-020-00710-y.

Syed Ashar Javed, Dinkar Juyal, Harshith Padigela, Amaro Taylor-Weiner, Limin Yu, and Aaditya Prakash. Additive MIL: Intrinsically interpretable multiple instance learning for pathology. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20689–20702, 2022.

Jahoon Jeong, Youngje Oh, Donghwan Kim, and Hyunsoo Yoon. From benchmarks to interpretability: A holistic survey of deep learning for time series classification. *Knowledge-Based Systems*, 336:115251, 2026. doi: 10.1016/j.knosys.2025.115251.

Olav Kallenberg. *Foundations of Modern Probability*. Springer, New York, 2nd edition, 2002. doi: 10.1007/978-1-4757-4015-8.

Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Samuel Harford. Multivariate LSTM-FCNs for time series classification. *Neural Networks*, 116:237–245, 2019. doi: 10.1016/j.neunet.2019.04.014.

Dongha Lee, Seonghyeon Lee, and Hwanjo Yu. Learnable dynamic temporal pooling for time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8288–8296, 2021. doi: 10.1609/aaai.v35i9.17008.

Seongjun Lee, Seokhyun Lee, and Changhee Lee. INSHAPE: Instance-level shapelets for interpretable time-series classification. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, 2026.

Min Liu, Hui Sheng, Ningyi Zhang, Panpan Zhao, Yugen Yi, Yirui Jiang, and Jiangyan Dai. DSDCLNet: Dual-stream encoder and dual-level contrastive learning network for supervised multivariate time series classification. *Knowledge-Based Systems*, 292:111638, 2024. doi: 10.1016/j.knosys.2024.111638.

Matthew Middlehurst, James Large, Michael Flynn, Jason Lines, Aaron Bostrom, and Anthony Bagnall. HIVE-COTE 2.0: a new meta ensemble for time series classification. *Machine Learning*, 110:3211–3243, 2021. doi: 10.1007/s10994-021-06057-9.

Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, 38:1958–2031, 2024. doi: 10.1007/s10618-024-01022-1.

Yang Mu, Muhammad Shahzad, and Xiao Xiang Zhu. MPTSNet: Integrating multiscale periodic local patterns and global dependencies for multivariate time series classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19572–19580, 2025. doi: 10.1609/aaai.v39i18.34155.

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.

Leonardos Pantiskas, Kees Verstoep, Mark Hoogendoorn, and Henri Bal. Reinforcement learning-guided channel selection across time for multivariate time series classification. In *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1406–1413. IEEE, 2023. doi: 10.1109/SSCI52147.2023.10371794.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5171–5180. PMLR, 2019.

Peng Wang, Xuan Wu, Yafei Song, Deyang Tian, and Xiaodan Wang. A comprehensive review of time series classification: Traditional, deep learning, and few-shot learning methods. *Computer Science Review*, 61:100953, 2026. doi: 10.1016/j.cosrev.2026.100953.

Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018. doi: 10.1016/j.patcog.2017.08.026.

Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585. IEEE, 2017. doi: 10.1109/IJCNN.2017.7966039.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696.

Haoyu Xie, Haoxuan Li, Chunyuan Zheng, Haonan Yuan, Guorui Liao, Jun Liao, and Li Liu. Decomposing and fusing intra- and inter-sensor spatio-temporal signal for multi-sensor wearable human activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 14441–14449, 2025.

Xuchao Zhang, Yifeng Gao, Jessica Lin, and Chang-Tien Lu. TapNet: Multivariate time series classification with attentional prototypical network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6845–6852, 2020. doi: 10.1609/aaai.v34i04.6165.

A Assumptions and Proofs

A.1 Assumptions

Throughout, $(\Omega, \mathcal{F}, \mathbb{P})$ carries the input X, the label Y, and the representations $R^{1:T} = (R^1, \dots, R^T)$ produced by the encoder, with $G = \frac{1}{T}\sum_t R^t$. We use L to denote an arbitrary subset of local instances from $R^{1:T}$, referred to as local evidence. For a policy π, $L_\pi = (L_1, \dots, L_k)$ denotes the random local evidence formed by the sequence of local instances selected by π. We use the following conditions.

- (A1) The label alphabet $\mathcal{Y}$ is finite with $|\mathcal{Y}| = K$. Every conditional entropy of Y is then bounded by $\log K$.
- (A2) $\mathcal{X}$, $\mathcal{G}$ and $\mathcal{R}$ are standard Borel spaces, so regular conditional distributions exist and are almost surely unique (Kallenberg, 2002).
- (A3) The encoder is fixed, so the joint law of $(Y, G, R^{1:T})$ does not depend on the selector or the classifier.
- (A4) Any randomisation used by the selector is generated by $U \sim \mathrm{Unif}[0, 1]$ independent of (X, Y), so Y and the selection are conditionally independent given $(G, R^{1:T})$.

Assumption (A3) fixes the object of study. Every statement below describes the selector at a given encoder state, and end-to-end training moves that state under the classification objective. Assumption (A4) makes each statement a population claim about a fixed policy evaluated on fresh data; the selector parameters are fitted with labels, so the training sample does not satisfy this conditional independence.

A.2 Preliminary Lemmas

Lemma 1 (Strict properness of the log score)**.** *For $p, q \in \Delta(\mathcal{Y})$, $\sum_y p(y) \log \frac{1}{q(y)} = H(p) + D_{\mathrm{KL}}(p\|q) \geq H(p)$, with equality if and only if $q = p$.*

Proof. Expand the left side and apply nonnegativity of the Kullback-Leibler divergence together with its equality condition (Cover & Thomas, 2006; Gneiting & Raftery, 2007). □

Lemma 2 (Bayes log-risk equals conditional entropy)**.** *Let Z take values in a standard Borel space and let $\eta(z) = P_{Y|Z}(\cdot \mid z)$. Under (A1) and (A2),*

$$\mathcal{R}^*(Z) = \inf_q \mathbb{E}\big[-\log q(Y \mid Z)\big] = H(Y \mid Z),$$

and the infimum is attained exactly at $q(\cdot \mid z) = \eta(z)$ for P_Z-almost every z.

Proof. The integrand is nonnegative, so Tonelli's theorem permits exchanging the order of integration and gives

$$\mathbb{E}\big[-\log q(Y \mid Z)\big] = \mathbb{E}\big[H(\eta(Z))\big] + \mathbb{E}\big[D_{\mathrm{KL}}\big(\eta(Z)\|q(\cdot \mid Z)\big)\big].$$

The first term equals $H(Y \mid Z)$, is finite by (A1), and does not depend on q. The second is nonnegative and vanishes exactly under the stated condition by Lemma 1. □

Lemma 3 (Conditional mutual information identity)**.** *Under (A1) and (A2), $I(Y; L \mid G) = H(Y \mid G) - H(Y \mid G, L)$, and this quantity lies in $[0, \log K]$.*

Proof. Disintegration gives $P_{Y,L|G=g}(dy, dl) = P_{Y|G=g,L=l}(y) P_{L|G=g}(dl)$. Since $\mathbb{P}\big(P_{Y|G}(Y \mid G) = 0\big) = \mathbb{E}\big[\sum_y \eta_y(G) \mathbf{1}\{\eta_y(G) = 0\}\big] = 0$, the Radon-Nikodym derivative of $P_{Y,L|G}$ with respect to $P_{Y|G} \otimes P_{L|G}$ is almost surely well defined and equals $P_{Y|G,L}(Y)/P_{Y|G}(Y)$. Taking logarithms and expectations splits the result into $-H(Y \mid G, L)$ and $-H(Y \mid G)$, both finite by (A1), so the separation is legitimate. Nonnegativity follows from that of the divergence. □

Lemma 4 (Posterior form of complementary information)**.** *Under (A1) and (A2),*

$$I(Y; L \mid G) = \mathbb{E}_{G,L}\Big[D_{\mathrm{KL}}\big(P(Y \mid G, L) \,\|\, P(Y \mid G)\big)\Big].$$

Proof. Integrate the derivative of Lemma 3 first over Y given (G, L) and then over L given G. The inner integral is the stated divergence and all terms are finite by (A1). □

A.3 Proof of Proposition 1

Proof. Under the logarithmic score, the Bayes risk equals the conditional entropy,

$$\mathcal{R}^*(Z) = H(Y \mid Z) \tag{12}$$

(Gneiting & Raftery, 2007; Grünwald & Dawid, 2004). Therefore,

$$\mathcal{R}^*(G) - \mathcal{R}^*(G, L) = H(Y \mid G) - H(Y \mid G, L) \tag{13}$$

$$= I(Y; L \mid G), \tag{14}$$

where the second equality follows from the entropy identity for conditional mutual information (Cover & Thomas, 2006). It can be written as

$$I(Y; L \mid G) = \mathbb{E}_G \left[D_{\mathrm{KL}} \left(P(Y, L \mid G) \,\|\, P(Y \mid G) P(L \mid G) \right) \right]. \tag{15}$$

Since the KL divergence is non-negative and equals zero if and only if its two arguments are equal almost surely, this condition is

$$P(Y, L \mid G) = P(Y \mid G) P(L \mid G) \quad \text{a.s.}, \tag{16}$$

which is precisely the condition

$$Y \perp L \mid G. \tag{17}$$

Hence,

$$I(Y; L \mid G) = 0 \quad \Longleftrightarrow \quad Y \perp L \mid G, \tag{18}$$

$$I(Y; L \mid G) > 0 \quad \Longleftrightarrow \quad Y \not\perp L \mid G. \tag{19}$$

Combining this with $\mathcal{R}^*(G) - \mathcal{R}^*(G, L) = I(Y; L \mid G)$ gives

$$\mathcal{R}^*(G, L) < \mathcal{R}^*(G) \quad \Longleftrightarrow \quad Y \not\perp L \mid G. \tag{20}$$

□

A.4 Proof of Proposition 2

Proof. The variables form the Markov chain

$$Y \to (G, L) \to \psi(G, L)$$

Since $\psi(G, L)$ is a measurable function of (G, L),

$$I\big(Y; \psi(G, L) \mid G, L\big) = 0.$$

By the chain rule for mutual information,

$$I\big(Y; (G, L)\big) = I\big(Y; \psi(G, L)\big) + I\big(Y; (G, L) \mid \psi(G, L)\big).$$

Since conditional mutual information is non-negative,

$$I\big(Y; \psi(G, L)\big) \le I\big(Y; (G, L)\big),$$

which is the data-processing inequality. Moreover, since $\psi(G, L)$ is determined by (G, L), the chain rule gives

$$I(Y; (G, L)) = I(Y; \psi(G, L)) + I\big(Y; (G, L) \mid \psi(G, L)\big).$$

Hence, equality in the data-processing inequality holds exactly when

$$I\big(Y; (G, L) \mid \psi(G, L)\big) = 0,$$

which is equivalent to

$$Y \perp (G, L) \mid \psi(G, L).$$

Thus, equality holds exactly when $\psi(G, L)$ is sufficient for Y relative to (G, L) (Cover & Thomas, 2006). Lemma 2 converts both mutual informations into Bayes log-risks. The map

$$(g, \ell) \mapsto [g; \ell]$$

is a bijection, so

$$\sigma([G; L]) = \sigma(G, L)$$

and

$$\mathcal{R}^*([G; L]) = \mathcal{R}^*(G, L).$$

For the loss-free statement, let

$$\ell : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$$

be bounded below. For any observation Z, let $\mathcal{D}_Z$ denote the set of decision rules that depend only on Z. Every $d \in \mathcal{D}_{\psi(G,L)}$ induces $d \circ \psi \in \mathcal{D}_{(G,L)}$ with the same risk, so $\mathcal{D}_{\psi(G,L)}$ embeds into $\mathcal{D}_{(G,L)}$ and the infimum over the latter is no larger. This deterministic information reduction is a special case of Blackwell's comparison of experiments (Blackwell, 1953). □

A.5 Proof of Proposition 3

Proof. By definition,

$$\mathcal{J}(\pi, \phi_C) = \mathbb{E}\big[\log q_{\phi_C}(Y \mid G, L_\pi)\big].$$

Adding and subtracting $\log P_{Y|G}(Y)$ inside the expectation gives

$$\mathcal{J}(\pi, \phi_C) = \mathbb{E}\left[\log \frac{q_{\phi_C}(Y \mid G, L_\pi)}{P_{Y|G}(Y)}\right] + \mathbb{E}\left[\log P_{Y|G}(Y)\right] \tag{21}$$

$$= \mathbb{E}\left[\log \frac{q_{\phi_C}(Y \mid G, L_\pi)}{P_{Y|G}(Y)}\right] - H(Y \mid G). \tag{22}$$

Therefore,

$$\mathcal{J}(\pi, \phi_C) + H(Y \mid G) = \mathbb{E}\left[\log \frac{q_{\phi_C}(Y \mid G, L_\pi)}{P_{Y|G}(Y)}\right]. \tag{23}$$

By Lemma 3, the complementary information of the selected evidence can be written as

$$I(Y; L_\pi \mid G) = \mathbb{E}\left[\log \frac{P_{Y|G,L_\pi}(Y)}{P_{Y|G}(Y)}\right]. \tag{24}$$

Subtracting Eqn. 23 from Eqn. 24 cancels the common denominator $P_{Y|G}(Y)$ and yields

$$I(Y; L_\pi \mid G) - \big[\mathcal{J}(\pi, \phi_C) + H(Y \mid G)\big] \tag{25}$$

$$= \mathbb{E}\left[\log \frac{P_{Y|G,L_\pi}(Y)}{q_{\phi_C}(Y \mid G, L_\pi)}\right] \tag{26}$$

$$= \mathbb{E}\left[D_{\mathrm{KL}}\left(P_{Y|G,L_\pi} \| q_{\phi_C}\right)\right] = \Delta_{\phi_C}(\pi). \tag{27}$$

Rearranging gives

$$\mathcal{J}(\pi, \phi_C) + H(Y \mid G) = I(Y; L_\pi \mid G) - \Delta_{\phi_C}(\pi).$$

Since the KL divergence is non-negative,

$$\Delta_{\phi_C}(\pi) \geq 0,$$

and hence

$$\mathcal{J}(\pi, \phi_C) + H(Y \mid G) \leq I(Y; L_\pi \mid G).$$

Equality holds exactly when $\Delta_{\phi_C}(\pi) = 0$, which occurs when

$$q_{\phi_C}(Y \mid G, L_\pi) = P_{Y|G,L_\pi}(Y) \quad \text{a.s.}$$

Thus, the bound is tight when q_{ϕ_C} is the Bayes predictor given (G, L_π). Assumption (A1) and the integrability hypothesis ensure that the quantities above are finite and that these rearrangements are well defined. □

The selector update increases $I(Y; L_\pi \mid G) - \Delta_{\phi_C}(\pi)$ rather than the first term alone, so a step raising the reward need not raise the complementary information. This is the position of every variational estimator of mutual information (Barber & Agakov, 2003; Poole et al., 2019), and the two objectives coincide as the classifier approaches the Bayes predictor.

B Additional Implementation Details

B.1 Model Architecture

We use InceptionTime as the temporal encoder for LISTS. The encoder consists of two consecutive Inception blocks, each containing three Inception modules. Each module contains four parallel branches with convolutional kernel sizes of 10, 20, and 40, together with a max-pooling branch using a kernel size of 3 and stride 1 followed by a 1×1 convolution. Each branch produces 32 output channels, and their outputs are concatenated to form a 128-dimensional temporal representation at

each time step. A 1×1 convolution with 32 output channels is used as the bottleneck layer. Replicate padding is applied to preserve the temporal resolution. Each Inception module is followed by batch normalization and ReLU activation, and each Inception block includes a residual connection implemented using a 1×1 convolution and batch normalization. For sequences shorter than 21 time steps, the input is right-padded to a minimum length of 21 before being processed by the backbone.

Given an input time series, the backbone produces temporal representations $\mathbf{r}_i = [\mathbf{r}_i^1, \ldots, \mathbf{r}_i^T]$, where $\mathbf{r}_i^t \in \mathbb{R}^{128}$. The global representation $\mathbf{g}_i \in \mathbb{R}^{128}$ is obtained by average pooling over all temporal representations. A fixed sinusoidal positional encoding with the same dimensionality is added to the temporal representations, followed by dropout with probability 0.1. The positional encoding supports sequences of up to 20,000 time steps.

LISTS sequentially selects k local instances from the temporal representations. At each selection step, the actor evaluates candidate temporal instances conditioned on the global representation and the previously selected local instances. The actor input consists of the 128-dimensional candidate representation, the 128-dimensional global representation, and $k-1$ history slots, each with dimension 128. If fewer than $k-1$ local instances have been selected, the remaining history slots are filled with zero vectors. The actor is implemented as a two-layer multilayer perceptron,

$$128(k+1) \to 32 \to 1,$$

where the first linear layer is followed by a GELU activation. Already selected positions are masked to prevent repeated selection. During training, positions are sampled from the categorical policy, whereas inference uses greedy argmax selection. The temporal representations supplied to the actor are detached from the computation graph, so the policy loss does not update the backbone through the actor.

The final representation concatenates the global representation and all selected local representations,

$$\mathbf{z}_i = [\mathbf{g}_i; \mathbf{r}_i^{t_1}; \ldots; \mathbf{r}_i^{t_k}] \in \mathbb{R}^{128(k+1)}.$$

A single linear classifier maps this representation directly to the class logits,

$$\mathbb{R}^{128(k+1)} \to \mathbb{R}^{N_{\text{class}}}.$$

B.2 Training and Preprocessing

We use the official UEA TRAIN and TEST splits without constructing an additional validation split for LISTS. Missing values are replaced with zeros. If channels within a sample have unequal lengths, they are right-padded with zeros, and samples of unequal lengths within each split are further right-padded to the maximum length of that split. Each sample is independently normalized as

$$\mathbf{x} \leftarrow \frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})},$$

where the mean and standard deviation are computed jointly over all channels and time steps of that sample. For constant-valued samples, the standard deviation is set to one. No statistics are shared between the training and test sets, and no data augmentation is applied.

During training, the batch size is set adaptively as $\min(\max(\lfloor N_{\text{train}}/10 \rfloor, 2), 16)$, while the test batch size is fixed to 16. Training batches are shuffled at each epoch. For optimization, the classification network and the actor use separate Adam optimizers with a learning rate of 10^{-3} and no weight decay. The classification optimizer updates the InceptionTime backbone and fusion classifier, whereas the actor optimizer updates only the actor parameters.

The classification objective is the mean cross-entropy loss. To optimize the selection policy, the reward for each sample is defined as the negative classification cross-entropy. The mean reward within each mini-batch is used as the baseline, and the resulting advantage is detached from the computation graph.

For checkpoint selection, the training loss is recomputed over the complete training split at the end of each epoch in evaluation mode using deterministic greedy instance selection.

Experiments are conducted on NVIDIA RTX A5000 GPUs with 24 GB of memory.

B.3 ATTENTION-BASED SELECTOR

To distinguish the contribution of reinforcement learning from that of learned local instance selection itself, we construct an attention-based variant of LISTS, denoted as Attention-3. The variant preserves the same temporal representations, positional encoding, global conditioning, history conditioning, sequential selection procedure, and final concatenation used by LISTS-3. The RL policy and REINFORCE objective are replaced by a differentiable attention-based selector.

At selection step j, the attention selector assigns a logit to each unselected temporal representation using the same contextual information as the RL actor,

$$e_{i,j}^{t} = A_{\mathrm{attn}}\left(r_i^t; g_i; \mathcal{H}_{i,j-1}\right), \tag{28}$$

where $\mathcal{H}_{i,j-1}$ contains the representations selected in the preceding steps. Positions that have already been selected are masked before normalization.

During training, we use the straight-through Gumbel-Softmax estimator to obtain a discrete selection while retaining gradients through the attention distribution. Specifically,

$$p_{i,j}^{t} = \frac{\exp\left((e_{i,j}^{t} + \gamma_t)/\tau\right)}{\sum_{s \notin \mathcal{S}_{i,j-1}} \exp\left((e_{i,j}^{s} + \gamma_s)/\tau\right)}, \qquad \gamma_t \sim \mathrm{Gumbel}(0,1), \tag{29}$$

where τ is the Gumbel-Softmax temperature. We use $\tau = 1$ throughout the experiment. Let $h_{i,j}$ denote the one-hot vector corresponding to the largest component of $p_{i,j}$. The straight-through selection vector is

$$\tilde{p}_{i,j} = h_{i,j} + p_{i,j} - \mathrm{sg}(p_{i,j}), \tag{30}$$

which uses the hard selection in the forward pass and the soft attention distribution for gradient propagation.

The selected representation at step j is

$$\tilde{r}_i^{\,j} = \sum_{t=1}^{T} \tilde{p}_{i,j}^{t} r_i^t. \tag{31}$$

After each step, the selected position is masked and the procedure is repeated until three local instances have been obtained. The resulting representations are concatenated with the global representation and passed to the same classifier used by LISTS-3.

Attention-3 is trained solely with the classification loss $\mathcal{L}_{\mathrm{cls}}$ and does not use a reward signal or the REINFORCE objective. At inference time, the Gumbel noise is removed and the highest-scoring unselected position is selected deterministically at each step. This construction isolates the selection mechanism while keeping the remaining LISTS architecture unchanged.

C ADDITIONAL RESULTS

Table 5 reports the standard deviations of classification accuracy across three independent runs for all reproduced methods. The results show generally modest variation across random seeds. LISTS-3 achieves a mean standard deviation of 0.0161 across the 30 datasets, while larger fluctuations are mainly observed on several small datasets.

D DATASET DETAILS

D.1 UEA DATASETS

We evaluate LISTS on all 30 datasets from the UEA time series classification archive. Table 6 summarizes the official training and test split sizes, number of classes, number of channels, and time series length of each dataset. For the four datasets containing unequal-length time series, the minimum and maximum sequence lengths are reported.

Table 5: Standard deviations of classification accuracy across three independent runs on the 30 UEA multivariate time series classification datasets.

Dataset	MiniROCKET KDD'21	ConvTran DMKD'24	TimeMIL ICML'24	MPTSNet AAAI'25	TimeSliver ICLR'26	**LISTS-3 (Ours)**
ArticularyWordRecognition	0.0000	0.0068	0.0057	0.0096	0.0083	0.0027
AtrialFibrillation	0.0314	0.0943	0.0629	0.0544	0.0544	0.0629
BasicMotions	0.0000	0.0000	0.0118	0.0816	0.0000	0.0000
CharacterTrajectories	0.0190	0.0013	0.0029	0.0096	0.0046	0.0017
Cricket	0.0000	0.0065	0.0398	0.0236	0.0236	0.0000
DuckDuckGeese	0.0094	0.0432	0.0680	0.0499	0.0712	0.0163
EigenWorms	0.0036	0.0286	0.0559	0.0187	0.0249	0.0252
Epilepsy	0.0000	0.0000	0.0090	0.0296	0.0068	0.0034
ERing	0.0000	0.0126	0.0240	0.0277	0.0709	0.0288
EthanolConcentration	0.0062	0.0249	0.0450	0.0522	0.0109	0.0112
FaceDetection	0.0030	0.0102	0.0452	0.0027	0.0068	0.0033
FingerMovements	0.0047	0.0340	0.0094	0.0368	0.0170	0.0249
HandMovementDirection	0.0127	0.0608	0.0255	0.0521	0.0552	0.0671
Handwriting	0.0019	0.0141	0.0080	0.0312	0.0151	0.0164
Heartbeat	0.0023	0.0061	0.0409	0.0105	0.0046	0.0211
InsectWingbeat	0.0429	0.0038	0.0019	0.0006	0.0110	0.0044
JapaneseVowels	0.0034	0.0056	0.0111	0.0270	0.0677	0.0038
Libras	0.0052	0.0094	0.0340	0.0343	0.0228	0.0069
LSST	0.0035	0.0042	0.0143	0.0038	0.0022	0.0082
MotorImagery	0.0047	0.0386	0.0249	0.0262	0.0330	0.0464
NATOPS	0.0069	0.0052	0.0069	0.0105	0.0069	0.0026
PEMS-SF	0.0082	0.0238	0.0547	0.0216	0.0263	0.0072
PenDigits	0.0004	0.0037	0.0212	0.0063	0.0033	0.0032
PhonemeSpectra	0.0020	0.0042	0.0084	0.0037	0.0076	0.0042
RacketSports	0.0000	0.0142	0.0082	0.0082	0.0054	0.0142
SelfRegulationSCP1	0.0016	0.0281	0.0174	0.0196	0.0084	0.0202
SelfRegulationSCP2	0.0045	0.0302	0.0276	0.0139	0.0396	0.0052
SpokenArabicDigits	0.0484	0.0017	0.0028	0.0045	0.0014	0.0033
StandWalkJump	0.0314	0.0314	0.0831	0.0000	0.0629	0.0629
UWaveGestureLibrary	0.0015	0.0273	0.0128	0.0159	0.0596	0.0039
Mean Std.	0.0086	0.0192	0.0261	0.0229	0.0244	0.0161

D.2 SYNTHETIC *Sleep* DATASET DETAILS

The synthetic *Sleep* dataset is adapted from the WebTraffic generation process introduced by MILLET (Early et al., 2024). Each time series represents an eight-hour synthetic sleep-inspired cyclic pattern sampled at one-minute intervals, resulting in $T = 480$ time points. Each series contains five equal-length cycles of 96 time points. The training and test splits are generated separately under a fixed random seed, and each split contains 50 samples per class.

Background generation. We first generate a five-cycle warped sinusoidal background. For a normalized time coordinate u, we define

$$\mathrm{WarpedSin}(a, b, p, s, u) = \frac{a}{2} \sin\left(2\pi(u-p) - \frac{\sin(2\pi(u-p))}{s}\right) + b, \tag{32}$$

where a, b, p, and s control the amplitude, intercept, phase, and skew, respectively. For each time series, these parameters are sampled independently as

$$a \sim \mathcal{U}(2,4), \qquad b \sim \mathcal{U}(2.5,5), \qquad p \sim \mathcal{U}(-0.05,0.05), \qquad s \sim \mathcal{U}(1,3). \tag{33}$$

A fixed phase offset of 0.55 is added when generating the cyclic background. Gaussian noise is then added to each time point with standard deviation

$$\sigma \sim \mathcal{U}(0.1, 0.3). \tag{34}$$

To introduce a gradual decrease over the synthetic sleep period, the five cycles are scaled using

$$c = [0.60, 0.55, 0.50, 0.45, 0.40]. \tag{35}$$

Table 6: Statistics of the 30 datasets in the UEA multivariate time series classification archive.

Dataset	Train	Test	Classes	Channels	Length
ArticularyWordRecognition	275	300	25	9	144
AtrialFibrillation	15	15	3	2	640
BasicMotions	40	40	4	6	100
CharacterTrajectories	1422	1436	20	3	60–182
Cricket	108	72	12	6	1197
DuckDuckGeese	50	50	5	1345	270
ERing	30	270	6	4	65
EigenWorms	128	131	5	6	17984
Epilepsy	137	138	4	3	206
EthanolConcentration	261	263	4	3	1751
FaceDetection	5890	3524	2	144	62
FingerMovements	316	100	2	28	50
HandMovementDirection	160	74	4	10	400
Handwriting	150	850	26	3	152
Heartbeat	204	205	2	61	405
InsectWingbeat	25000	25000	10	200	2–22
JapaneseVowels	270	370	9	12	7–29
LSST	2459	2466	14	6	36
Libras	180	180	15	2	45
MotorImagery	278	100	2	64	3000
NATOPS	180	180	6	24	51
PEMS-SF	267	173	7	963	144
PenDigits	7494	3498	10	2	8
PhonemeSpectra	3315	3353	39	11	217
RacketSports	151	152	4	6	30
SelfRegulationSCP1	268	293	2	6	896
SelfRegulationSCP2	200	180	2	7	1152
SpokenArabicDigits	6599	2199	10	13	4–93
StandWalkJump	12	15	3	4	2500
UWaveGestureLibrary	120	320	8	3	315

For each sample, an independent jitter term $u_j \sim \mathcal{U}(0.95, 1.05)$ is sampled for cycle j. The final scale applied to that cycle is

$$\alpha_j = c_j u_j. \tag{36}$$

This produces five equal-length cycles with a gradually decreasing magnitude while maintaining small sample-specific variation.

Signature injection. The dataset contains nine classes consisting of one normal class and eight signature classes. For every signature-class sample, one contiguous interval is selected uniformly at random. Its length is sampled as

$$l \sim \mathcal{U}_{\mathbb{Z}}(17, 137), \tag{37}$$

and its starting position is sampled such that the entire interval lies within the time series. The selected interval is recorded as the ground-truth discriminative region. The normal class contains no injected signature. After signature injection, negative values are clipped to zero.

The eight signature classes are generated as follows.

Flip. The values within the selected interval are reversed in temporal order.

Skew. The selected interval is temporally warped by interpolation. A skew amount is sampled from $\mathcal{U}(0.25, 0.45)$ and is added to or subtracted from 0.5 with equal probability. This produces a relative skew position

$$w \in [0.05, 0.25] \cup [0.75, 0.95]. \tag{38}$$

The interval is then interpolated so that its original midpoint is shifted to approximately position wl within the interval.

Peak. A smooth peak is generated using the probability density function of a standard normal distribution evaluated from -5 to 5. A scale value $q \sim \mathcal{U}(1.5, 2.5)$ is sampled, and the selected

interval is multiplied by

$$1 + q \ \mathrm{PDF}_{\mathcal{N}(0,1)}\,. \tag{39}$$

Trough. The same procedure as Peak is used, but the scale is negative. Specifically, $q \sim \mathcal{U}(-2.5, -1.5)$, producing a smooth local decrease within the selected interval.

Wander. A linear trend is added to the selected interval. Its magnitude is sampled from $\mathcal{U}(2, 3)$ and is assigned a positive or negative direction with equal probability. The trend changes linearly from zero at the beginning of the interval to the sampled magnitude at the end.

Noise. Gaussian noise is added to every point in the selected interval. Its standard deviation is sampled from

$$\sigma_{\mathrm{noise}} \sim \mathcal{U}(0.5, 1.0). \tag{40}$$

Cutoff. A cutoff level is sampled from

$$c \sim \mathcal{U}(0, 0.2). \tag{41}$$

The selected interval is replaced by values sampled from $\mathcal{N}(c, 0.1)$.

Average. The selected interval is replaced by a smoothed version of the time series obtained using a moving average. The moving-average window size is sampled uniformly from the integers between 10 and 19.