UTICA: MULTI-OBJECTIVE SELF-DISTLLATION FOUNDATION MODEL PRETRAINING FOR TIME SERIES CLASSIFICATION

Yessin Moakher
Ecole Polytechnique, France
yessin.moakher@polytechnique.edu

Youssef Attia El Hili
Huawei Noah's Ark Lab

Vasilii Feofanov
42.com

ABSTRACT

Self-supervised foundation models have achieved remarkable success across domains, including time series. However, the potential of non-contrastive methods, a paradigm that has driven significant advances in computer vision, remains underexplored for time series. In this work, we adapt DINOv2-style self-distillation to pretrain a time series foundation model, building on the Mantis tokenizer and transformer encoder architecture as our backbone. Through a student–teacher framework, our method Utica learns representations that capture both temporal invariance via augmented crops and fine-grained local structure via patch masking. Our approach achieves state-of-the-art classification performance on both UCR and UEA benchmarks. These results suggest that non-contrastive methods are a promising and complementary pretraining strategy for time series foundation models. [1]

Track: Research

1 INTRODUCTION

Foundation models have fundamentally reshaped machine learning. Building on their success in language and vision, researchers are now developing Time Series Foundation Models (TSFMs) to learn universal representations from temporal data. Most TSFMs focus on forecasting, employing autoregressive (Ansari et al., 2024), supervised (Auer et al., 2025), or masked reconstruction objectives (Goswami et al., 2024). While effective for prediction, such objectives prioritize local temporal consistency over global semantic structure, which is vital for classification tasks such as fault detection (Zhou et al., 2025), cardiovascular diagnostics (Li et al., 2025), and EEG decoding (Gnassounou et al., 2025).

One prevailing approach to bridging this gap is contrastive learning. Drawing inspiration from computer vision (Chen et al., 2020; Radford et al., 2021), TSFMs such as Mantis (Feofanov et al., 2025) have achieved remarkable performance using a contrastive objective: pulling together positive pairs (different augmentations of the same example) while pushing apart negative ones (different examples within a batch). However, this approach rests on the risky assumption that different samples within a batch are semantically distinct. In time series, where samples may share similar dynamics, frequency content, or temporal structure, this assumption often fails. This introduces false negatives, potentially harming representation quality and discouraging the model from capturing globally shared patterns.

Other approaches rely on self-distillation. Pieper et al. (2023) propose a CNN-based student–teacher framework trained on masked views. Subsequently, NuTime (Lin et al., 2024) introduced a Transformer architec-

[1] The code is available at: `https://www.github.com/fegounna/Utica`.

ture trained with a BYOL-style (Grill et al., 2020) self-distillation loss on pairs of randomly cropped global views.

While both methods avoid explicit negatives, they rely on a single view-generation strategy: masking only (Pieper et al., 2023) or paired global crops only (Lin et al., 2024). In contrast, inspired by the success of DINOv2 (Oquab et al., 2024) in computer vision, we propose **Utica**, which pretrains a Transformer-based TSFM with a combination of (i) a self-distillation loss applied to heterogeneous global and local multi-crop augmentations, and (ii) a masking objective applied to the original sequence. We argue that combining masking with multi-crop augmentations is a natural fit for temporal data, facilitating the learning of global representations invariant to scale, partial observability and temporal offsets. We experimentally show the superiority of Utica on two time series benchmarks: UCR (Dau et al., 2019) and UEA (Bagnall et al., 2018).

2 Methodology

Pretraining Dataset. Following Xie et al. (2025) who showed that time series foundation models can be efficiently pretrained entirely on synthetic data, we generate synthetic sequences using a causal generative model defined by a directed acyclic graph (DAG). For each root node v, we sample a time series $x_v \in \mathbb{R}^T$ from a Gaussian Process $x_v(t) \sim \mathcal{GP}(\mu_v(t), k_v(t, t'))$, where k_v is a randomly composed covariance kernel and μ_v is a non-stationary mean function. For each non-root node v with parents $\mathrm{pa}(v)$, the signal is generated as $x_v(t) = f_v\left(\sum_{u \in \mathrm{pa}(v)} w_{uv} x_u(t) + b_v\right)$, with $w_{uv}, b_v \sim \mathcal{N}(0,1)$ and f_v a randomly sampled nonlinearity. A subset of nodes is selected to generate time series examples used for pretraining .

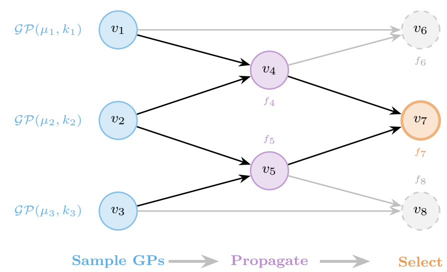

Figure 1: Example of a time series sample generation via causal DAG.

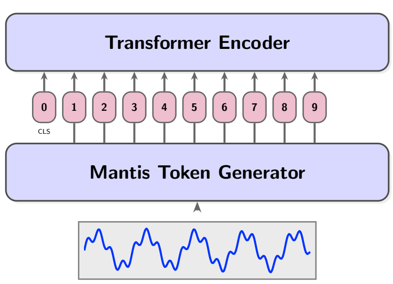

Figure 2: Architecture.

Architecture. As a backbone, we use a classical Transformer (Dosovitskiy et al., 2021) with a modality-specific token generator from Mantis (Feofanov et al., 2025). Each univariate input series is represented using three complementary transformations: the instance-normalized series, its first-order differential (to capture stationarity), and patch-level encodings of the mean and standard deviation of raw segments (Lin et al., 2024). These embeddings are concatenated, projected to the model dimension $D = 256$, and processed through 6 Transformer encoder layers, with a learnable class [CLS] token and sinusoidal positional encodings to preserve temporal information. The output that corresponds to the [CLS] token are treated as final embeddings produced by the model.

Pretraining Loss. The training setup of Utica consists of a Student network f_{θ_s} and a Teacher network f_{θ_t} with identical architectures. The Teacher weights are updated via an exponential moving average of the Student weights, following the schedule $\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$, where $\lambda \in [0, 1]$ increases linearly during pretraining. The total loss $\mathcal{L}$ is a sum of three distinct objectives: $\mathcal{L} = \mathcal{L}_{\text{DINO}} + \mathcal{L}_{\text{iBOT}} + 0.1\mathcal{L}_{\text{KoLeo}}$, which we explain below.

- **DINO Loss.** To encourage invariance to temporal scale and local noise, we employ a multi-crop strategy combined with random jittering. Given a time series input $x \in \mathbb{R}^T$, we generate the following augmentations: (a) two global random crops that cover from 40% to 100% of the signal resized to

$T = 512$, (b) eight local random crops that represent smaller segments with a 10% to 40% coverage of the signal and that are further resized to $T_{local} = 256$. Gaussian jitter noise is applied randomly to some crops. The DINO objective minimizes the cross-entropy between the Student's and Teacher's [CLS] token probability distributions obtained via a shared-architecture projection head applied to each network's output. The Student is exposed to all augmented views, while the Teacher processes only global views. Probabilities are derived using softmax normalization. To prevent collapse, the Teacher's output is regularized using the Sinkhorn-Knopp algorithm for centering and a temperature parameter for sharpening.

- **iBOT Loss.** To learn dense local features, we employ the iBOT objective. We apply patch-level masking to the global views fed into the Student network, with variable ratio drawn from a uniform distribution $\mathcal{U}(0.1, 0.7)$ and a sample probability of 0.5. For a masked view $\hat{x}$, the Student predicts the token distribution of the masked patches, while the Teacher views the unmasked original signal. The loss is calculated as the cross-entropy between the Teacher and Student projected patch distributions at masked indices.
- **KoLeo Regularizer (Sablayrolles et al., 2019).** To encourage a uniform distribution of features in the batch and prevent collapse, we apply the Kozachenko-Leonenko (KoLeo) differential entropy estimator to the Student's global [CLS] tokens before the projection head.

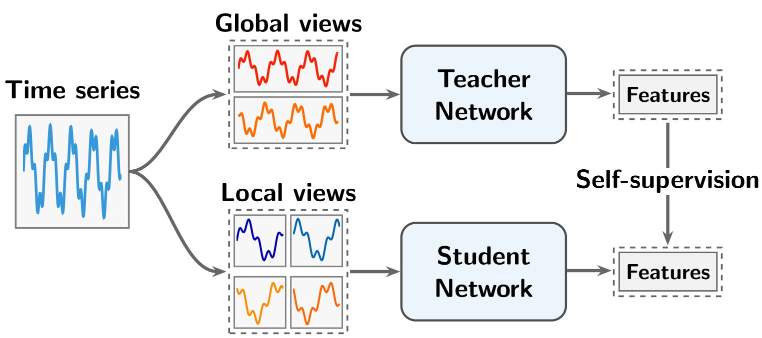

Figure 3: **U**nlabeled **Ti**me-series **C**rop **A**ugmented framework. The self-supervised objective aims to match the features produced by the teacher with those produced by the student. Full details about pretraining can be found in Appendix B.

3 Experimental Results

In our experiments, we follow the Mantis evaluation protocol (Feofanov et al., 2025), considering two regimes: linear probing (frozen representations) and fine-tuning (end-to-end).

Baselines. We compare UTICA, for which we use the Teacher network as the final model, against the following methods: Mantis (Feofanov et al., 2025), a 8M-parameter model trained with a contrastive loss; Moment (Goswami et al., 2024), a 385M-parameter T5-based masked auto-encoder; NuTime (Lin et al., 2024), a 2M-parameter Transformer encoder trained with self-distillation and GPT4TS (Zhou et al., 2023) a 80M-parameter partially fine-tuned GPT2.

Datasets. We evaluate on the UCR archive (Dau et al., 2019), consisting of 128 univariate time series datasets, and the UEA archive (Bagnall et al., 2018), from which we use 21 multivariate datasets.

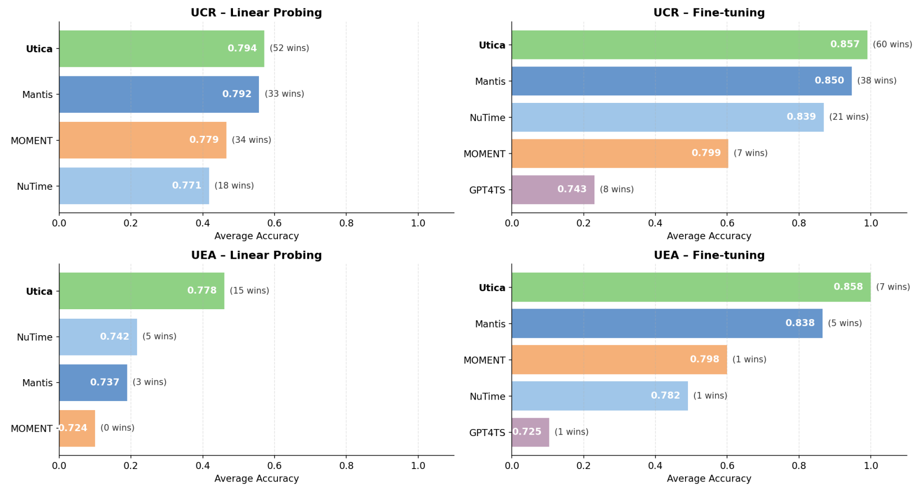

Figure 4: Average accuracy comparison across linear probing and fine-tuning on both UCR and UEA benchmarks. Number of wins shown in parentheses.

Results. Figure 4 displays the obtained results on UCR and UEA when models are used in frozen and fine-tuned way (see the complete results in Appendix D). Utica consistently outperforms all baselines across benchmarks and evaluation regimes. On UCR linear probing, Utica achieves 0.794 average accuracy with $52/128$ wins, versus Mantis (0.792, 33 wins) and MOMENT (0.779, 34 wins). Under fine-tuning: Utica reaches 0.857 with 60 wins versus Mantis (0.850, 38 wins). On UEA, Utica achieves the best average rank in both settings (1.60 linear probing, 1.50 fine-tuning).

Table 1: Ablation on the loss type used for pretraining.

Loss Type	data2vec	iBOT+KoLeo	DINO+KoLeo	UTICA
Accuracy	0.7802	0.735	0.747	**0.794**

Ablation Study. In this section, we analyze our pretraining loss function more thoroughly. First, we study the contribution of the masking and scale-invariant loss components by pretraining the foundation model with iBOT+KoLeo and DINO+KoLeo, respectively. In Table 1, we display the linear probing performance on UCR dataset. The experimental results reveal that local masked prediction (iBOT) and global multi-crop alignment (DINO) provide complementary supervision signals: individually, they have significantly lower performances (0.735 and 0.747, respectively) compared to their combination (0.794). In addition, we compare the proposed loss with data2vec (Baevski et al., 2022), another self-distillation approach adapted for time series data (Pieper et al., 2023). Under the same linear probing protocol, we can see that UTICA outperforms data2vec by 1.38% on the UCR benchmark.

4 Conclusion

Our results demonstrate that an adaptation of DINOv2-style self-distillation, previously successful in computer vision, transfers effectively to time series foundation models. The consistent results on both UCR and UEA benchmarks suggest that self-distillation pretraining is a promising direction for time series classification foundation models. Future work includes exploring alternative backbone architectures and scaling the model parameters.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. Tirex: Zero-shot forecasting across long and short horizons with enhanced in-context learning, 2025. URL https://arxiv.org/abs/2505.23719.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language, 2022. URL https://arxiv.org/abs/2202.03555.

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, 2018. URL https://arxiv.org/abs/1811.00075.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive, 2019. URL https://arxiv.org/abs/1810.07758.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Vasilii Feofanov, Songkang Wen, Marius Alonso, Romain Ilbert, Hongbo Guo, Malik Tiomoko, Lujia Pan, Jianfeng Zhang, and Ievgen Redko. Mantis: Lightweight calibrated foundation model for user-friendly time series classification. *arXiv preprint arXiv:2502.15637*, 2025.

Théo Gnassounou, Yessin Moakher, Shifeng Xie, Vasilii Feofanov, and Ievgen Redko. Leveraging generic time series foundation models for EEG classification. *arXiv preprint arXiv:2510.27522*, 2025.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models, 2024. URL https://arxiv.org/abs/2402.03885.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

Hao Li, Bowen Deng, Chang Xu, ZhiYuan Feng, Viktor Schlegel, Yu-Hao Huang, Yizheng Sun, Jingyuan Sun, Kailai Yang, Yiyao Yu, and Jiang Bian. MIRA: Medical time series foundation model for real-world health data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Auy2DmlJBO.

Chenguo Lin, Xumeng Wen, Wei Cao, Congrui Huang, Jiang Bian, Stephen Lin, and Zhirong Wu. NuTime: Numerically multi-scaled embedding for large- scale time-series pretraining. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=TwiSBZ0p9u.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Felix Pieper, Konstantin Ditschuneit, Martin Genzel, Alexandra Lindt, and Johannes Otterbach. Self-distilled representation learning for time series, 2023. URL https://arxiv.org/abs/2311.11335.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search, 2019. URL https://arxiv.org/abs/1806.03198.

Shifeng Xie, Vasilii Feofanov, Marius Alonso, Ambroise Odonnat, Jianfeng Zhang, and Ievgen Redko. Cauker: Classification time series foundation models can be pretrained on synthetic data only. In *ICML Workshop on Foundation Models for Structured Data (FMSD)*, 2025.

Songqi Zhou, Ruixue Liu, Yixing Wang, Jia Lu, and Benben Jiang. Batterybert for realistic battery fault detection using point-masked signal modeling, 2025. URL https://arxiv.org/abs/2506.15712.

Tian Zhou, PeiSong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all:power general time series analysis by pretrained lm, 2023. URL https://arxiv.org/abs/2302.11939.

Appendix

A Synthetic Data Generation

We generate synthetic time series using a causal DAG. Root node signals are sampled from Gaussian Processes:

$$x_v(t) \sim \mathcal{GP}(\mu_v(t), k_v(t, t')),$$

where $\mu_v(t)$ is a non-stationary mean function and k_v is a randomly composed covariance kernel.

Non-root nodes combine their parent signals via a weighted sum followed by a nonlinear transformation:

$$x_v(t) = f_v\left(\sum_{u \in \mathrm{pa}(v)} w_{uv} x_u(t) + b_v\right),$$

with weights w_{uv} and biases b_v sampled from $\mathcal{N}(0, 1)$ and f_v a randomly sampled nonlinearity.

A subset of node signals is observed to form the input time series used for pretraining. Figure 5 shows an example of the resulting synthetic sequences.

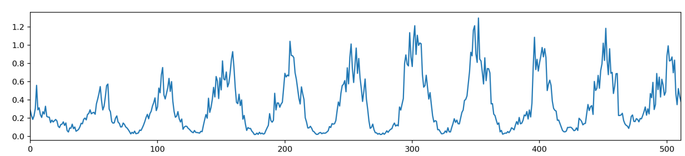

Figure 5: Example of synthetic time series from the DAG-based generator.

B Self-Supervised Pretraining Details

This appendix provides a comprehensive description of the self-supervised pretraining procedure used to train the Mantis-8M time-series foundation model.

B.1 Model Architecture

Mantis-8M is a Transformer encoder operating on univariate time series of length multiple of 32. The architecture consists of two stages:

Token Generator Unit. Each input series $\mathbf{x} \in \mathbb{R}^T$ is split into $P = 32$ non-overlapping patches of window size $w = T/P = 16$. Two convolutional branches extract features from (i) the z-score-normalised series and (ii) its first-order finite difference $\Delta\mathbf{x}$, both padded to preserve length. Each branch uses a 1-D convolution and output channels equal to the hidden dimension $d = 256$, followed by layer normalisation. The branch outputs are averaged within each patch to yield P patch embeddings of dimension d.

In parallel, per-patch mean and standard deviation are encoded by *Multi-Scaled Scalar Encoders* with nine log-spaced scales $\{10^{-4}, \ldots, 10^4\}$, each producing embeddings of dimension $d_{\text{scalar}} = 32$ (tolerance $\epsilon = 1.1$). The concatenated convolutional and scalar embeddings are projected to $d = 256$ by a linear encoder.

Transformer Encoder Unit. The P patch tokens are prepended with a learnable `[CLS]` token and summed with sinusoidal positional encodings. The sequence is processed by a Transformer with the following hyper-parameters:

Table 2: Transformer hyper-parameters of the Transformer Encoder Unit.

Parameter	**Value**
Depth (number of layers)	6
Attention heads	8
Key/query/value dimension per head	128
MLP hidden dimension	512
Hidden dimension d	256
Dropout	0.1

A final layer normalisation is applied, and the output is split into the `[CLS]` token embedding $\mathbf{z}_{\texttt{cls}} \in \mathbb{R}^d$ and the patch token embeddings $\mathbf{Z}_{\text{patch}} \in \mathbb{R}^{P \times d}$.

A learnable `[MASK]` token $\mathbf{m} \in \mathbb{R}^d$ (initialised to zero) is used to replace masked patch tokens before they enter the Transformer (see Section B.5).

B.2 Student–Teacher Framework

We adopt the exponential moving average (EMA) student–teacher paradigm. The student and teacher share the same architecture (backbone + projection heads) but differ in how they are updated:

- **Student:** updated via gradient descent.
- **Teacher:** updated via EMA of the student parameters, $\boldsymbol{\theta}_t \leftarrow m\,\boldsymbol{\theta}_t + (1-m)\,\boldsymbol{\theta}_s$, where m follows a cosine schedule from $m_0 = 0.992$ to $m_f = 1.0$ over the course of training. The teacher is kept in evaluation mode with no gradient computation.

B.3 Projection Heads

Both the DINO (CLS-token) and iBOT (patch-token) objectives use separate head projectors with identical architectures:

Table 3: Projection head hyper-parameters (shared by DINO and iBOT heads).

Parameter	**Value**
Input dimension	256
Hidden dimension	2 048
Bottleneck dimension	256
Output dimension (prototypes) K	65 536
Number of MLP layers	3
Activation	GELU
Weight normalisation on last layer	ℓ_2-normalisation before linear

The MLP maps the input to the bottleneck dimension, applies ℓ_2 normalisation, and a final linear layer (without bias) projects to $K = 65{,}536$ prototypes. Weights are initialised with truncated normals ($\sigma = 0.02$).

B.4 Data Augmentation

Each univariate time series $\mathbf{x} \in \mathbb{R}^T$ undergoes multi-crop augmentation:

Global crops. Two global crop are generated. A random contiguous sub-sequence with a fraction $r \sim \mathcal{U}(0.4, 1.0)$ of the original length, then linearly risized to $T_g = 512$ via interpolation. Gaussian jitter with $\sigma_{\text{jitter}} = 0.2 \times \text{std}(\mathbf{x})$ is applied to one of the global crops.

Local crops. Eight local crops are generated. Each takes a contiguous sub-sequence with a fraction $r \sim \mathcal{U}(0.1, 0.4)$ and is resized to $T_\ell = 256$ via interpolation. Jitter is applied with probability $p = 0.5$.

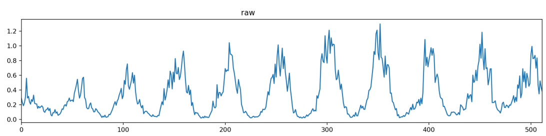

Figure 6: Original time series.

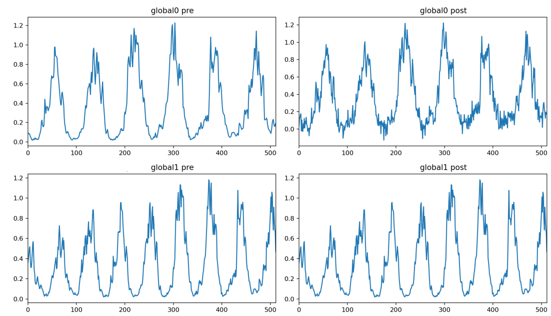

Figure 7: Two global views before and after eventual jittering.

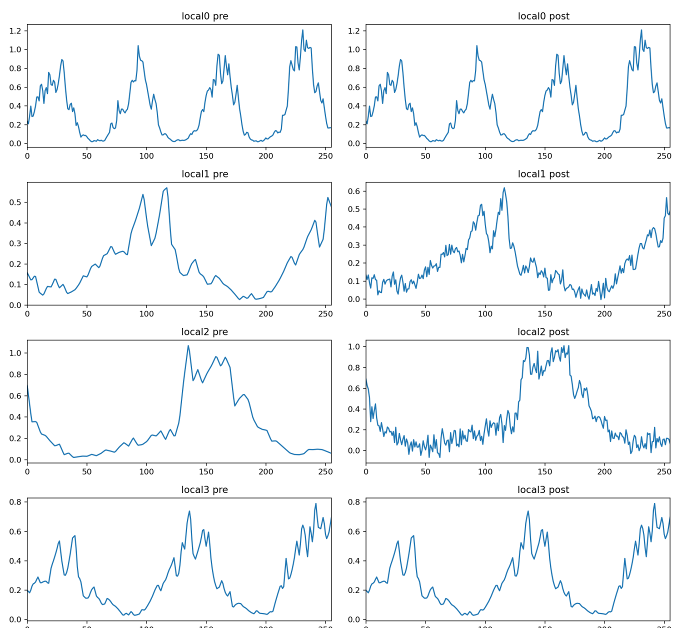

Figure 8: Four local views before and after eventual jittering.

B.5 Masking Strategy

Patch-level masking is applied to the global crops seen by the student to enable the iBOT patch-reconstruction objective. Given $P = 32$ patches:

1. Each sample in the batch is selected for masking with probability $p_{\text{mask}} = 0.5$.
2. For selected samples, the mask ratio is drawn from a linearly spaced schedule between $r_{\min} = 0.1$ and $r_{\max} = 0.7$, yielding between $\lfloor 0.1 \times 32 \rfloor = 3$ and $\lfloor 0.7 \times 32 \rfloor = 22$ masked patches.
3. Masked patch positions are chosen uniformly at random (without replacement).
4. Unselected samples receive an all-zero mask (no patches masked).

Masked patch tokens in the student input are replaced by the learnable `[MASK]` token before entering the Transformer. The teacher always sees unmasked inputs.

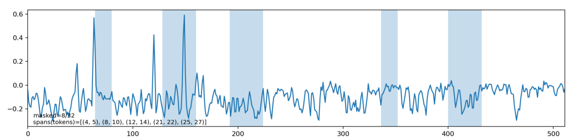

Figure 9: Masked patches example

B.6 Loss Functions

The total loss is a weighted combination of three terms:

$$\mathcal{L} = \lambda_{\text{DINO}}\, \mathcal{L}_{\text{DINO}} + \lambda_{\text{iBOT}}\, \mathcal{L}_{\text{iBOT}} + \lambda_{\text{KoLeo}}\, \mathcal{L}_{\text{KoLeo}},$$

with $\lambda_{\text{DINO}} = 1.0$, $\lambda_{\text{iBOT}} = 1.0$, and $\lambda_{\text{KoLeo}} = 0.1$.

DINO loss. The DINO loss is the cross-entropy between teacher and student probability distributions over $K = 65{,}536$ prototypes, computed on `[CLS]` tokens.

Teacher targets via Sinkhorn–Knopp. Let $\mathbf{z}_t^{(i)} \in \mathbb{R}^K$ denote the teacher head output for sample i in teacher crop t. We form the soft-assignment matrix $\mathbf{Q} \in \mathbb{R}^{K \times B_{\text{eff}}}$ (prototypes $\times$ samples) as:

$$Q_{k,i} = \frac{\exp\bigl(z_{t,k}^{(i)}/\tau_t\bigr)}{\sum_{k',i'} \exp\bigl(z_{t,k'}^{(i')}/\tau_t\bigr)},$$

where the denominator includes a distributed all-reduce so the matrix sums to 1 across all GPUs. Three iterations of Sinkhorn–Knopp alternating row-normalisation (each row sums to $1/K$) and column-normalisation (each column sums to $1/B_{\text{eff}}$) are then applied, where $B_{\text{eff}} = B \times \text{world\_size}$ is the effective batch size. After convergence the columns are rescaled to sum to 1, yielding the teacher probability vectors $\mathbf{p}_t^{(i)} = \mathbf{Q}_{:,i} \in \Delta^{K-1}$.

The teacher temperature τ_t follows a linear warmup from $\tau_{t,0} = 0.04$ to $\tau_t = 0.07$ over the first 2.5 epochs.

Student probabilities. The student head output $\mathbf{z}_s^{(i)}$ for student crop s is converted to log-probabilities:

$$\mathbf{q}_s^{(i)} = \log\operatorname{softmax}\bigl(\mathbf{z}_s^{(i)}/\tau_s\bigr), \qquad \tau_s = 0.1.$$

CROSS-ENTROPY COMPUTATION. The per-pair cross-entropy between student crop s and teacher crop t is:

$$H(s,t) = -\frac{1}{B}\sum_{i=1}^{B}\sum_{k=1}^{K} p_{t,k}^{(i)} \, \log q_{s,k}^{(i)}.$$

The total DINO loss distinguishes two terms, following the DINOv2 scaling convention. Let S_g and S_ℓ denote the number of global and local student crops, and T_g the number of teacher (global) crops.

Global–global term: a global student crop is not compared with the same-index teacher crop. The raw loss matrix $L_{s,t} = H(s,t)$ for $s,t \in \{1,\ldots,S_g\}$ has its diagonal set to zero before summation. The effective number of terms is $n_g = S_g \cdot T_g - \min(S_g, T_g)$:

$$\mathcal{L}_{\text{DINO}}^{\text{global}} = \frac{1}{B\,n_g}\sum_{\substack{s,t=1\\ s\neq t}}^{S_g,\,T_g}\Big(-\sum_{i,k} p_{t,k}^{(i)} \, \log q_{s,k}^{(i)}\Big).$$

Local–global term: all local student crops are compared with all teacher global crops. The number of terms is $n_\ell = S_\ell \cdot T_g$:

$$\mathcal{L}_{\text{DINO}}^{\text{local}} = \frac{1}{B\,n_\ell}\sum_{s=1}^{S_\ell}\sum_{t=1}^{T_g}\Big(-\sum_{i,k} p_{t,k}^{(i)} \, \log q_{s,k}^{(i)}\Big).$$

The two terms are combined with scaling factors that reflect the fraction of cross-crop comparisons each contributes:

$$\alpha_g = \frac{n_g}{n_g + n_\ell}, \qquad \alpha_\ell = \frac{n_\ell}{n_g + n_\ell}, \qquad \mathcal{L}_{\text{DINO}} = \alpha_g\,\mathcal{L}_{\text{DINO}}^{\text{global}} + \alpha_\ell\,\mathcal{L}_{\text{DINO}}^{\text{local}}.$$

iBOT patch loss. The iBOT objective operates on individual patch tokens rather than the `[CLS]` token. It is computed only over patches that are masked for the student.

TEACHER TARGETS VIA SINKHORN–KNOPP (PATCH-LEVEL). Let $M = \sum_{i=1}^{B}\sum_{j=1}^{P} \mathbb{1}[\text{mask}_{i,j} = 1]$ be the total number of masked patches across all samples and GPUs. The teacher backbone processes unmasked global crops; the patch tokens at masked positions are extracted and passed through the iBOT head to produce $\hat{\mathbf{z}}_m \in \mathbb{R}^K$ for each masked patch $m = 1,\ldots,M$.

Sinkhorn–Knopp is applied identically to the DINO case:

$$Q_{k,m}^{\text{patch}} \propto \exp\big(\hat{z}_{m,k}/\tau_t\big),$$

with 3 iterations of alternating row- and column-normalisation using distributed all-reduces. This yields teacher patch probabilities $\hat{\mathbf{p}}_m \in \Delta^{K-1}$.

STUDENT PREDICTIONS. The student receives the global crop with masked patches replaced by the learnable `[MASK]` token. The output patch tokens at masked positions are extracted and passed through the student iBOT head, yielding logits $\hat{\mathbf{s}}_m \in \mathbb{R}^K$.

WEIGHTED CROSS-ENTROPY. The per-patch cross-entropy is:

$$\ell_m = -\sum_{k=1}^{K} \hat{p}_{m,k} \, \log \mathrm{softmax}_k\big(\hat{\mathbf{s}}_m/\tau_s\big), \qquad \tau_s = 0.1.$$

To ensure that each *sample* contributes equally regardless of how many of its patches are masked, each patch loss is weighted by the inverse of the number of masked patches in its parent sample. Concretely, let M_i denote the number of masked patches in sample i, and let $\sigma(m) = i$ map each masked patch index to its parent sample. The weight is:

$$w_m = \frac{1}{\max(M_{\sigma(m)},\ 1)}.$$

The final iBOT loss is normalised by the number of global-crop samples B:

$$\mathcal{L}_{\text{iBOT}} = -\frac{1}{B}\sum_{m=1}^{M} w_m\,\ell_m.$$

KoLeo loss. The Kozachenko–Leonenko entropy estimator is applied to the ℓ_2-normalised pre-head `[CLS]` tokens of the student's global crops:

$$\mathcal{L}_{\text{KoLeo}} = -\frac{1}{B}\sum_{i=1}^{B}\log\bigl(\min_{j\neq i}\|\hat{\mathbf{z}}_i - \hat{\mathbf{z}}_j\|_2 + \epsilon\bigr),$$

where $\hat{\mathbf{z}}$ denotes the ℓ_2-normalised representation and $\epsilon = 10^{-8}$.

B.7 Optimisation

Table 4: Optimisation hyper-parameters.

Parameter	**Value**
Optimiser	AdamW
$(\beta_1,\ \beta_2)$	$(0.9,\ 0.999)$
Base learning rate	1×10^{-3}
Minimum learning rate	1×10^{-7}
Learning rate schedule	Cosine decay
Warmup epochs	0.7
Weight decay (start $\rightarrow$ end)	$0.04 \rightarrow 0.4$ (cosine schedule)
Gradient clipping	3.0
Layer-wise learning rate decay	0.9
Patch embed learning rate multiplier	0.2
Freeze last layer epochs	0.07

The learning rate and weight decay follow cosine schedules. Layer-wise learning rate decay is applied to the Transformer Encoder layers: layer l (out of $L = 6$) receives a learning rate multiplied by 0.9^{L+1-l}. Parameters in the token generator unit additionally receive a $0.2\times$ multiplier. Bias, norm, and gamma parameters have their weight decay set to zero.

The final linear projection in each projection head (the layer mapping from the bottleneck to the $K = 65{,}536$ prototypes) follows a separate learning rate schedule. This schedule is identical to the main cosine learning rate schedule, except that its values are set to zero for the first iterations. During this period the prototype weights receive no gradient updates, while all other parameters train normally. This stabilises early training.

C Downstream Evaluation

C.1 Linear Probing

Linear classifier trained on frozen backbone embeddings.

- Optimizer: AdamW
- Epochs: 100
- Learning rate: Fixed to 1e-3

C.2 Fine-Tuning

Following (Feofanov et al., 2025) protocol for full model fine-tuning:

- Optimizer: AdamW
- Scheduler: Cosine
- Weight decay: 0.05
- Epochs: 100
- Learning rate: grid search on $[1e-4, 2e-4, 1e-3]$, used 20% of training data as validation

D Experimental Results

D.1 Per-Dataset Performance

Table 7: Comparison under linear probing regime on UCR. The best epoch performance is averaged over 3 random seeds and reported with the standard deviation. Bold indicates the best result per row. NuTime, Moment and Mantis results are from (Feofanov et al., 2025).

Dataset	NuTime	Moment	Mantis	Utica
ACSF1	$0.7367_{\pm 0.0351}$	$\mathbf{0.75}_{\pm 0.0173}$	$0.6133_{\pm 0.0208}$	$0.6200_{\pm 0.0000}$
Adiac	$0.7255_{\pm 0.0115}$	$\mathbf{0.7886}_{\pm 0.0074}$	$0.7332_{\pm 0.0039}$	$0.5090_{\pm 0.0142}$
AllGestureWiimoteX	$0.661_{\pm 0.003}$	$0.6105_{\pm 0.0136}$	$0.6705_{\pm 0.0044}$	$\mathbf{0.6824}_{\pm 0.0068}$
AllGestureWiimoteY	$0.6362_{\pm 0.0079}$	$0.6576_{\pm 0.0079}$	$0.6671_{\pm 0.0057}$	$\mathbf{0.7029}_{\pm 0.0071}$
AllGestureWiimoteZ	$0.6024_{\pm 0.0179}$	$0.5767_{\pm 0.0103}$	$\mathbf{0.6695}_{\pm 0.0033}$	$0.6448_{\pm 0.0016}$
ArrowHead	$0.76_{\pm 0.0198}$	$\mathbf{0.8076}_{\pm 0.0144}$	$0.7105_{\pm 0.0175}$	$0.6514_{\pm 0.0618}$
BME	$0.8444_{\pm 0.0168}$	$0.9756_{\pm 0.0139}$	$0.9311_{\pm 0.0038}$	$\mathbf{0.9933}_{\pm 0.0000}$
Beef	$0.6556_{\pm 0.077}$	$\mathbf{0.7444}_{\pm 0.0509}$	$0.6556_{\pm 0.0509}$	$0.6111_{\pm 0.0192}$
BeetleFly	$0.8667_{\pm 0.0289}$	$\mathbf{0.95}_{\pm 0.0}$	$0.85_{\pm 0.05}$	$0.8833_{\pm 0.0577}$
BirdChicken	$0.9667_{\pm 0.0289}$	$0.85_{\pm 0.0}$	$\mathbf{1.0}_{\pm 0.0}$	$0.9000_{\pm 0.0000}$
CBF	$0.9744_{\pm 0.0011}$	$0.9411_{\pm 0.0078}$	$0.993_{\pm 0.0013}$	$\mathbf{0.9952}_{\pm 0.0006}$
Car	$0.75_{\pm 0.0}$	$\mathbf{0.7944}_{\pm 0.0255}$	$0.7722_{\pm 0.0419}$	$0.7611_{\pm 0.0347}$
Chinatown	$0.932_{\pm 0.0017}$	$\mathbf{0.9806}_{\pm 0.0034}$	$0.8737_{\pm 0.0168}$	$0.9660_{\pm 0.0034}$
ChlorineConcentration	$0.6675_{\pm 0.0035}$	$\mathbf{0.6901}_{\pm 0.0042}$	$0.6806_{\pm 0.0019}$	$0.5729_{\pm 0.0021}$
CinCECGTorso	$0.737_{\pm 0.0152}$	$0.6998_{\pm 0.0118}$	$0.6611_{\pm 0.0036}$	$\mathbf{0.7430}_{\pm 0.0123}$
Coffee	$0.9167_{\pm 0.0206}$	$0.8929_{\pm 0.0}$	$0.9524_{\pm 0.0206}$	$\mathbf{0.9881}_{\pm 0.0206}$

Continued on next page

Table 7 – *continued from previous page*

Dataset	NuTime	Moment	Mantis	Utica
Computers	**0.78**$_{\pm 0.004}$	0.6173$_{\pm 0.0162}$	0.7373$_{\pm 0.0092}$	0.7600$_{\pm 0.0040}$
CricketX	0.6701$_{\pm 0.0146}$	0.6795$_{\pm 0.0051}$	0.7368$_{\pm 0.0171}$	**0.7436**$_{\pm 0.0092}$
CricketY	0.6556$_{\pm 0.0053}$	0.6897$_{\pm 0.0077}$	**0.7504**$_{\pm 0.0065}$	0.7350$_{\pm 0.0015}$
CricketZ	0.6863$_{\pm 0.0171}$	0.7128$_{\pm 0.0044}$	**0.7906**$_{\pm 0.0039}$	0.7778$_{\pm 0.0121}$
Crop	0.6683$_{\pm 0.0026}$	**0.7035**$_{\pm 0.0026}$	0.6756$_{\pm 0.0018}$	0.6960$_{\pm 0.0011}$
DiatomSizeReduction	0.8322$_{\pm 0.0115}$	**0.8867**$_{\pm 0.0019}$	0.8845$_{\pm 0.0019}$	0.8322$_{\pm 0.0019}$
DistalPhalanxOutlineAgeGroup	0.7362$_{\pm 0.011}$	0.7506$_{\pm 0.011}$	**0.789**$_{\pm 0.015}$	0.7770$_{\pm 0.0000}$
DistalPhalanxOutlineCorrect	0.7742$_{\pm 0.0055}$	**0.7886**$_{\pm 0.0055}$	0.75$_{\pm 0.0126}$	0.7621$_{\pm 0.0021}$
DistalPhalanxTW	0.6763$_{\pm 0.0}$	0.6571$_{\pm 0.011}$	0.6859$_{\pm 0.011}$	**0.6906**$_{\pm 0.0000}$
DodgerLoopDay	0.5167$_{\pm 0.0361}$	0.4542$_{\pm 0.0144}$	0.55$_{\pm 0.0217}$	**0.6167**$_{\pm 0.0191}$
DodgerLoopGame	0.756$_{\pm 0.0084}$	0.8116$_{\pm 0.0126}$	0.7585$_{\pm 0.0221}$	**0.8720**$_{\pm 0.0042}$
DodgerLoopWeekend	0.9565$_{\pm 0.0072}$	0.9614$_{\pm 0.0111}$	0.9517$_{\pm 0.0084}$	**0.9831**$_{\pm 0.0042}$
ECG200	0.8133$_{\pm 0.0153}$	**0.8967**$_{\pm 0.0153}$	0.82$_{\pm 0.01}$	0.8167$_{\pm 0.0153}$
ECG5000	0.9313$_{\pm 0.0003}$	**0.9384**$_{\pm 0.0008}$	0.9211$_{\pm 0.001}$	0.9381$_{\pm 0.0011}$
ECGFiveDays	0.7801$_{\pm 0.017}$	0.8564$_{\pm 0.0223}$	0.909$_{\pm 0.0218}$	**0.9311**$_{\pm 0.0367}$
EOGHorizontalSignal	0.4346$_{\pm 0.0032}$	0.5571$_{\pm 0.0112}$	**0.5875**$_{\pm 0.0089}$	0.5737$_{\pm 0.0084}$
EOGVerticalSignal	0.2716$_{\pm 0.008}$	0.4595$_{\pm 0.0097}$	0.4751$_{\pm 0.0}$	**0.4908**$_{\pm 0.0112}$
Earthquakes	0.7458$_{\pm 0.0042}$	0.7458$_{\pm 0.0042}$	0.7482$_{\pm 0.0}$	**0.7530**$_{\pm 0.0083}$
ElectricDevices	0.7046$_{\pm 0.0014}$	0.7142$_{\pm 0.001}$	**0.7226**$_{\pm 0.0026}$	0.7107$_{\pm 0.0014}$
EthanolLevel	0.3407$_{\pm 0.0101}$	**0.4227**$_{\pm 0.0058}$	0.2993$_{\pm 0.011}$	0.3273$_{\pm 0.0046}$
FaceAll	0.6363$_{\pm 0.0053}$	0.7398$_{\pm 0.0074}$	**0.7815**$_{\pm 0.0074}$	0.7544$_{\pm 0.0045}$
FaceFour	0.7841$_{\pm 0.0521}$	0.7765$_{\pm 0.0174}$	**0.9508**$_{\pm 0.0066}$	0.9242$_{\pm 0.0347}$
FacesUCR	0.7141$_{\pm 0.003}$	0.7951$_{\pm 0.0034}$	0.8354$_{\pm 0.0054}$	**0.9080**$_{\pm 0.0025}$
FiftyWords	0.5949$_{\pm 0.0125}$	0.6777$_{\pm 0.0111}$	0.6462$_{\pm 0.0096}$	**0.7604**$_{\pm 0.0076}$
Fish	0.9162$_{\pm 0.0033}$	0.8705$_{\pm 0.0201}$	**0.9333**$_{\pm 0.0066}$	0.9010$_{\pm 0.0087}$
FordA	0.8932$_{\pm 0.002}$	**0.9015**$_{\pm 0.0008}$	0.8581$_{\pm 0.0048}$	0.8939$_{\pm 0.0023}$
FordB	0.7642$_{\pm 0.0119}$	0.765$_{\pm 0.0019}$	0.7305$_{\pm 0.0031}$	**0.7790**$_{\pm 0.0012}$
FreezerRegularTrain	0.9738$_{\pm 0.0013}$	0.8994$_{\pm 0.0012}$	0.9374$_{\pm 0.0043}$	**0.9812**$_{\pm 0.0041}$
FreezerSmallTrain	**0.9549**$_{\pm 0.0059}$	0.7758$_{\pm 0.0067}$	0.7942$_{\pm 0.0059}$	0.9421$_{\pm 0.0106}$
Fungi	0.7043$_{\pm 0.0093}$	**0.9964**$_{\pm 0.0062}$	0.8262$_{\pm 0.0164}$	0.9892$_{\pm 0.0000}$
GestureMidAirD1	0.6744$_{\pm 0.0379}$	0.6744$_{\pm 0.0044}$	0.659$_{\pm 0.0044}$	**0.7615**$_{\pm 0.0077}$
GestureMidAirD2	0.5692$_{\pm 0.0}$	0.5744$_{\pm 0.016}$	0.6154$_{\pm 0.0077}$	**0.7103**$_{\pm 0.0089}$
GestureMidAirD3	0.3974$_{\pm 0.0247}$	0.359$_{\pm 0.0118}$	0.3282$_{\pm 0.016}$	**0.4077**$_{\pm 0.0077}$
GesturePebbleZ1	0.8915$_{\pm 0.0067}$	0.8469$_{\pm 0.0067}$	**0.9283**$_{\pm 0.0034}$	0.8760$_{\pm 0.0089}$
GesturePebbleZ2	0.8165$_{\pm 0.011}$	0.8376$_{\pm 0.0073}$	**0.9219**$_{\pm 0.0256}$	0.7932$_{\pm 0.0037}$
GunPoint	0.9444$_{\pm 0.0038}$	**1.0**$_{\pm 0.0}$	0.98$_{\pm 0.0067}$	0.9911$_{\pm 0.0038}$
GunPointAgeSpan	0.9684$_{\pm 0.0032}$	0.962$_{\pm 0.0032}$	0.9905$_{\pm 0.0}$	**0.9926**$_{\pm 0.0037}$
GunPointMaleVersusFemale	0.962$_{\pm 0.0032}$	0.9884$_{\pm 0.0018}$	**0.9958**$_{\pm 0.0018}$	0.9852$_{\pm 0.0018}$
GunPointOldVersusYoung	**1.0**$_{\pm 0.0}$	0.9577$_{\pm 0.008}$	0.9968$_{\pm 0.0}$	0.9937$_{\pm 0.0032}$
Ham	0.7206$_{\pm 0.022}$	**0.7714**$_{\pm 0.0}$	0.673$_{\pm 0.0145}$	0.7302$_{\pm 0.0198}$
HandOutlines	0.9045$_{\pm 0.0062}$	0.909$_{\pm 0.0056}$	**0.9162**$_{\pm 0.0072}$	0.8712$_{\pm 0.0165}$
Haptics	0.4481$_{\pm 0.0056}$	**0.5152**$_{\pm 0.0068}$	0.4968$_{\pm 0.0032}$	0.5076$_{\pm 0.0050}$
Herring	0.599$_{\pm 0.0325}$	0.6146$_{\pm 0.009}$	**0.6667**$_{\pm 0.0239}$	**0.6667**$_{\pm 0.0090}$
HouseTwenty	0.8655$_{\pm 0.0084}$	0.9356$_{\pm 0.0097}$	0.9412$_{\pm 0.0}$	**0.9580**$_{\pm 0.0000}$
InlineSkate	0.357$_{\pm 0.0105}$	0.3176$_{\pm 0.001}$	**0.363**$_{\pm 0.0136}$	0.3388$_{\pm 0.0038}$
InsectEPGRegularTrain	**1.0**$_{\pm 0.0}$	0.9183$_{\pm 0.0061}$	**1.0**$_{\pm 0.0}$	**1.0000**$_{\pm 0.0000}$

Continued on next page

Table 7 – *continued from previous page*

Dataset	NuTime	Moment	Mantis	Utica
InsectEPGSmallTrain	**1.0**$_{\pm 0.0}$	0.8501$_{\pm 0.0061}$	**1.0**$_{\pm 0.0}$	**1.0000**$_{\pm 0.0000}$
InsectWingbeatSound	0.5066$_{\pm 0.0066}$	**0.6158**$_{\pm 0.0037}$	0.519$_{\pm 0.0044}$	0.5614$_{\pm 0.0024}$
ItalyPowerDemand	0.8698$_{\pm 0.007}$	**0.9498**$_{\pm 0.0006}$	0.9077$_{\pm 0.0035}$	0.9326$_{\pm 0.0050}$
LargeKitchenAppliances	0.7227$_{\pm 0.0027}$	0.7333$_{\pm 0.0071}$	0.7804$_{\pm 0.0041}$	**0.8249**$_{\pm 0.0086}$
Lightning2	0.6885$_{\pm 0.0164}$	0.7377$_{\pm 0.0328}$	0.8033$_{\pm 0.0}$	**0.8197**$_{\pm 0.0164}$
Lightning7	0.6895$_{\pm 0.0158}$	0.6758$_{\pm 0.0209}$	0.7763$_{\pm 0.0285}$	**0.8493**$_{\pm 0.0000}$
Mallat	0.8304$_{\pm 0.0025}$	0.859$_{\pm 0.0111}$	0.8903$_{\pm 0.0137}$	**0.8948**$_{\pm 0.0101}$
Meat	0.8889$_{\pm 0.0192}$	**0.9444**$_{\pm 0.0096}$	0.9389$_{\pm 0.0096}$	**0.9444**$_{\pm 0.0096}$
MedicalImages	0.7044$_{\pm 0.0027}$	0.6939$_{\pm 0.0062}$	0.7079$_{\pm 0.0035}$	**0.7303**$_{\pm 0.0039}$
MelbournePedestrian	**0.912**$_{\pm 0.0045}$	0.8662$_{\pm 0.0047}$	0.9016$_{\pm 0.0031}$	0.9077$_{\pm 0.0025}$
MiddlePhalanxOutlineAgeGroup	0.6169$_{\pm 0.0065}$	0.5779$_{\pm 0.0065}$	0.5801$_{\pm 0.0099}$	**0.6494**$_{\pm 0.0112}$
MiddlePhalanxOutlineCorrect	0.7858$_{\pm 0.0099}$	**0.8625**$_{\pm 0.006}$	0.8099$_{\pm 0.0099}$	0.8419$_{\pm 0.0060}$
MiddlePhalanxTW	0.5238$_{\pm 0.0037}$	**0.5952**$_{\pm 0.0037}$	0.5368$_{\pm 0.0099}$	0.5887$_{\pm 0.0246}$
MixedShapesRegularTrain	0.9392$_{\pm 0.0013}$	0.9124$_{\pm 0.0023}$	**0.943**$_{\pm 0.0044}$	0.9340$_{\pm 0.0004}$
MixedShapesSmallTrain	**0.9061**$_{\pm 0.0029}$	0.8389$_{\pm 0.0041}$	0.8961$_{\pm 0.0004}$	0.8841$_{\pm 0.0023}$
MoteStrain	**0.9462**$_{\pm 0.0032}$	0.8914$_{\pm 0.0083}$	0.9137$_{\pm 0.0136}$	0.9271$_{\pm 0.0039}$
NonInvasiveFetalECGThorax1	0.7696$_{\pm 0.0049}$	**0.8887**$_{\pm 0.0035}$	0.6222$_{\pm 0.0037}$	0.8195$_{\pm 0.0075}$
NonInvasiveFetalECGThorax2	0.811$_{\pm 0.0016}$	**0.9138**$_{\pm 0.0006}$	0.6872$_{\pm 0.0025}$	0.8422$_{\pm 0.0028}$
OSULeaf	0.7975$_{\pm 0.0072}$	0.7355$_{\pm 0.0041}$	0.8747$_{\pm 0.0104}$	**0.8926**$_{\pm 0.0072}$
OliveOil	0.7111$_{\pm 0.0192}$	0.9$_{\pm 0.0}$	**0.9333**$_{\pm 0.0}$	0.4000$_{\pm 0.0000}$
PLAID	0.7933$_{\pm 0.0113}$	0.7312$_{\pm 0.0088}$	**0.8181**$_{\pm 0.0047}$	0.7374$_{\pm 0.0085}$
PhalangesOutlinesCorrect	0.7743$_{\pm 0.0041}$	**0.8248**$_{\pm 0.0007}$	0.7786$_{\pm 0.0042}$	0.7541$_{\pm 0.0012}$
Phoneme	0.2802$_{\pm 0.0044}$	0.2751$_{\pm 0.0062}$	**0.323**$_{\pm 0.0034}$	0.3193$_{\pm 0.0032}$
PickupGestureWiimoteZ	0.6933$_{\pm 0.0808}$	0.68$_{\pm 0.06}$	0.74$_{\pm 0.02}$	**0.8467**$_{\pm 0.0115}$
PigAirwayPressure	0.3782$_{\pm 0.01}$	0.1186$_{\pm 0.0028}$	**0.484**$_{\pm 0.0147}$	0.3013$_{\pm 0.0147}$
PigArtPressure	**0.9391**$_{\pm 0.0028}$	0.6106$_{\pm 0.0048}$	0.9103$_{\pm 0.0028}$	0.7532$_{\pm 0.0121}$
PigCVP	**0.8381**$_{\pm 0.0242}$	0.609$_{\pm 0.0373}$	0.7837$_{\pm 0.0127}$	0.7596$_{\pm 0.0083}$
Plane	0.9937$_{\pm 0.0055}$	0.9841$_{\pm 0.0145}$	**1.0**$_{\pm 0.0}$	**1.0000**$_{\pm 0.0000}$
PowerCons	0.9352$_{\pm 0.0032}$	**0.9463**$_{\pm 0.0032}$	0.9093$_{\pm 0.0032}$	0.9222$_{\pm 0.0056}$
ProximalPhalanxOutlineAgeGroup	0.8537$_{\pm 0.0049}$	0.839$_{\pm 0.0049}$	0.8537$_{\pm 0.0049}$	**0.8780**$_{\pm 0.0000}$
ProximalPhalanxOutlineCorrect	0.8385$_{\pm 0.0034}$	**0.8751**$_{\pm 0.0052}$	0.8202$_{\pm 0.0086}$	0.7915$_{\pm 0.0130}$
ProximalPhalanxTW	0.8065$_{\pm 0.0056}$	0.8114$_{\pm 0.0028}$	0.7691$_{\pm 0.0028}$	**0.8130**$_{\pm 0.0028}$
RefrigerationDevices	**0.5564**$_{\pm 0.0041}$	0.536$_{\pm 0.0046}$	0.504$_{\pm 0.0122}$	0.5227$_{\pm 0.0027}$
Rock	0.6067$_{\pm 0.0306}$	**0.84**$_{\pm 0.02}$	0.7133$_{\pm 0.0115}$	0.6933$_{\pm 0.0115}$
ScreenType	**0.5324**$_{\pm 0.0077}$	0.4436$_{\pm 0.0178}$	0.4649$_{\pm 0.0134}$	0.5253$_{\pm 0.0053}$
SemgHandGenderCh2	0.8928$_{\pm 0.0042}$	0.8028$_{\pm 0.0079}$	**0.9189**$_{\pm 0.0086}$	0.8739$_{\pm 0.0042}$
SemgHandMovementCh2	0.72$_{\pm 0.0038}$	0.5237$_{\pm 0.0013}$	**0.797**$_{\pm 0.0056}$	0.6111$_{\pm 0.0022}$
SemgHandSubjectCh2	0.7741$_{\pm 0.0134}$	0.6815$_{\pm 0.0026}$	**0.8622**$_{\pm 0.0135}$	0.7444$_{\pm 0.0059}$
ShakeGestureWiimoteZ	**0.9267**$_{\pm 0.0115}$	0.8533$_{\pm 0.0115}$	0.8867$_{\pm 0.0115}$	0.8467$_{\pm 0.0115}$
ShapeletSim	0.8833$_{\pm 0.0111}$	0.9648$_{\pm 0.0064}$	0.9278$_{\pm 0.0111}$	**0.9741**$_{\pm 0.0140}$
ShapesAll	0.8194$_{\pm 0.0025}$	0.8239$_{\pm 0.0092}$	0.8194$_{\pm 0.0054}$	**0.8789**$_{\pm 0.0025}$
SmallKitchenAppliances	0.8027$_{\pm 0.0071}$	0.72$_{\pm 0.0141}$	0.8089$_{\pm 0.0081}$	**0.8436**$_{\pm 0.0015}$
SmoothSubspace	0.8933$_{\pm 0.0067}$	0.92$_{\pm 0.0133}$	0.9067$_{\pm 0.0115}$	**0.9311**$_{\pm 0.0102}$
SonyAIBORobotSurface1	0.7987$_{\pm 0.0044}$	0.8087$_{\pm 0.006}$	0.787$_{\pm 0.0294}$	**0.8569**$_{\pm 0.0152}$
SonyAIBORobotSurface2	0.8297$_{\pm 0.0034}$	0.8353$_{\pm 0.0106}$	**0.8552**$_{\pm 0.0168}$	0.8475$_{\pm 0.0121}$
StarLightCurves	**0.9792**$_{\pm 0.0002}$	0.9734$_{\pm 0.0004}$	0.9759$_{\pm 0.0005}$	0.9779$_{\pm 0.0002}$

Continued on next page

Table 7 – *continued from previous page*

Dataset	NuTime	Moment	Mantis	Utica
Strawberry	$0.9378_{\pm 0.0072}$	$\mathbf{0.9604}_{\pm 0.0016}$	$0.9514_{\pm 0.0027}$	$0.9144_{\pm 0.0016}$
SwedishLeaf	$0.9205_{\pm 0.0065}$	$0.9205_{\pm 0.0051}$	$\mathbf{0.9275}_{\pm 0.0009}$	$0.9029_{\pm 0.0040}$
Symbols	$0.9407_{\pm 0.0126}$	$0.938_{\pm 0.0126}$	$0.9698_{\pm 0.0056}$	$\mathbf{0.9749}_{\pm 0.0017}$
SyntheticControl	$0.9667_{\pm 0.0033}$	$0.9433_{\pm 0.0}$	$0.9767_{\pm 0.0067}$	$\mathbf{0.9900}_{\pm 0.0000}$
ToeSegmentation1	$0.8596_{\pm 0.0044}$	$0.924_{\pm 0.0127}$	$\mathbf{0.9635}_{\pm 0.0067}$	$0.9532_{\pm 0.0110}$
ToeSegmentation2	$0.7256_{\pm 0.016}$	$0.8462_{\pm 0.0077}$	$\mathbf{0.9282}_{\pm 0.0044}$	$0.8923_{\pm 0.0133}$
Trace	$\mathbf{1.0}_{\pm 0.0}$	$0.99_{\pm 0.0}$	$\mathbf{1.0}_{\pm 0.0}$	$\mathbf{1.0000}_{\pm 0.0000}$
TwoLeadECG	$0.8712_{\pm 0.0303}$	$0.9649_{\pm 0.0116}$	$0.9962_{\pm 0.0005}$	$\mathbf{0.9965}_{\pm 0.0009}$
TwoPatterns	$0.8414_{\pm 0.0041}$	$0.8919_{\pm 0.0079}$	$0.8802_{\pm 0.0079}$	$\mathbf{0.9693}_{\pm 0.0005}$
UMD	$0.9329_{\pm 0.0223}$	$0.9699_{\pm 0.004}$	$0.9722_{\pm 0.0069}$	$\mathbf{0.9931}_{\pm 0.0000}$
UWaveGestureLibraryAll	$0.8773_{\pm 0.0028}$	$0.8975_{\pm 0.0021}$	$0.8458_{\pm 0.0057}$	$\mathbf{0.9181}_{\pm 0.0002}$
UWaveGestureLibraryX	$0.8053_{\pm 0.0007}$	$0.7829_{\pm 0.0071}$	$0.7696_{\pm 0.0028}$	$\mathbf{0.8142}_{\pm 0.0009}$
UWaveGestureLibraryY	$\mathbf{0.7338}_{\pm 0.0032}$	$0.7022_{\pm 0.0026}$	$0.6874_{\pm 0.0022}$	$0.7261_{\pm 0.0032}$
UWaveGestureLibraryZ	$0.7427_{\pm 0.0036}$	$0.73_{\pm 0.002}$	$0.7324_{\pm 0.0036}$	$\mathbf{0.7639}_{\pm 0.0046}$
Wafer	$\mathbf{0.9933}_{\pm 0.0006}$	$0.9867_{\pm 0.0003}$	$0.9903_{\pm 0.0003}$	$0.9921_{\pm 0.0011}$
Wine	$0.7469_{\pm 0.0107}$	$\mathbf{0.8457}_{\pm 0.0283}$	$0.7901_{\pm 0.0107}$	$0.5802_{\pm 0.0107}$
WordSynonyms	$0.5199_{\pm 0.0127}$	$\mathbf{0.6003}_{\pm 0.0078}$	$0.5502_{\pm 0.0081}$	$0.5664_{\pm 0.0065}$
Worms	$\mathbf{0.7403}_{\pm 0.0}$	$0.7056_{\pm 0.015}$	$0.6537_{\pm 0.027}$	$0.6450_{\pm 0.0075}$
WormsTwoClass	$0.7835_{\pm 0.0075}$	$0.7706_{\pm 0.0075}$	$\mathbf{0.8139}_{\pm 0.0198}$	$0.7273_{\pm 0.0225}$
Yoga	$0.8219_{\pm 0.0015}$	$\mathbf{0.8264}_{\pm 0.003}$	$0.815_{\pm 0.0012}$	$0.6860_{\pm 0.0111}$
# Wins	18	34	33	**52**
Avg. Rank	2.88	2.67	2.37	**2.08**
Avg. Acc.	0.7714	0.7786	0.7922	**0.7944**

Table 8: Comparison under the fine tuning regime on UCR. The best epoch performance is averaged over 3 random seeds and reported with the standard deviation. Bold indicates the best result per row. GPT4S, NuTime, Moment and Mantis results are from (Feofanov et al., 2025).

Dataset	GPT4TS	NuTime	Moment	Mantis	Utica
ACSF1	$0.4900_{\pm 0.0265}$	$0.6733_{\pm 0.0321}$	$0.6033_{\pm 0.0379}$	$\mathbf{0.7433}_{\pm 0.0115}$	$0.6733_{\pm 0.0419}$
Adiac	$0.3572_{\pm 0.0059}$	$0.7349_{\pm 0.0141}$	$0.5823_{\pm 0.0296}$	$0.7766_{\pm 0.0030}$	$\mathbf{0.8159}_{\pm 0.0130}$
AllGestureWiimoteX	$0.4895_{\pm 0.0105}$	$0.6386_{\pm 0.0038}$	$0.7000_{\pm 0.0103}$	$\mathbf{0.7619}_{\pm 0.0050}$	$0.7395_{\pm 0.0115}$
AllGestureWiimoteY	$0.5062_{\pm 0.0116}$	$0.7152_{\pm 0.0103}$	$0.7081_{\pm 0.0343}$	$\mathbf{0.7948}_{\pm 0.0097}$	$0.7390_{\pm 0.0106}$
AllGestureWiimoteZ	$0.4719_{\pm 0.0119}$	$0.6419_{\pm 0.0058}$	$0.6976_{\pm 0.0218}$	$\mathbf{0.7300}_{\pm 0.0100}$	$0.6967_{\pm 0.0141}$
ArrowHead	$0.7638_{\pm 0.0033}$	$0.8305_{\pm 0.0033}$	$0.7790_{\pm 0.0682}$	$0.8210_{\pm 0.0389}$	$\mathbf{0.8457}_{\pm 0.0123}$
BME	$0.9444_{\pm 0.0102}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9800_{\pm 0.0231}$	$0.9956_{\pm 0.0077}$	$\mathbf{1.0000}_{\pm 0.0000}$
Beef	$\mathbf{0.8000}_{\pm 0.0577}$	$\mathbf{0.8222}_{\pm 0.0192}$	$\mathbf{0.8000}_{\pm 0.0577}$	$0.7000_{\pm 0.0333}$	$0.7889_{\pm 0.0157}$
BeetleFly	$0.8167_{\pm 0.0289}$	$0.8833_{\pm 0.0764}$	$0.8000_{\pm 0.1500}$	$0.8833_{\pm 0.0764}$	$\mathbf{0.9500}_{\pm 0.0000}$
BirdChicken	$0.6167_{\pm 0.0289}$	$0.8500_{\pm 0.0500}$	$0.7667_{\pm 0.0289}$	$\mathbf{0.9000}_{\pm 0.0500}$	$\mathbf{0.9000}_{\pm 0.0000}$
CBF	$0.8596_{\pm 0.0083}$	$0.9652_{\pm 0.0046}$	$0.9767_{\pm 0.0172}$	$0.9848_{\pm 0.0074}$	$\mathbf{0.9993}_{\pm 0.0005}$
Car	$0.8000_{\pm 0.0167}$	$0.8444_{\pm 0.0096}$	$\mathbf{0.8778}_{\pm 0.0419}$	$0.8722_{\pm 0.0096}$	$0.8389_{\pm 0.0079}$
Chinatown	$0.9718_{\pm 0.0089}$	$0.9738_{\pm 0.0000}$	$0.9708_{\pm 0.0127}$	$0.9718_{\pm 0.0017}$	$\mathbf{0.9796}_{\pm 0.0024}$
ChlorineConcentration	$0.5733_{\pm 0.0091}$	$0.6718_{\pm 0.0080}$	$0.6289_{\pm 0.0178}$	$0.6971_{\pm 0.0112}$	$\mathbf{0.7218}_{\pm 0.0145}$
CinCECGTorso	$0.7930_{\pm 0.0353}$	$\mathbf{0.9302}_{\pm 0.0067}$	$0.7430_{\pm 0.0080}$	$0.7814_{\pm 0.0173}$	$0.8517_{\pm 0.0188}$
Coffee	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9881_{\pm 0.0206}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$
Computers	$0.6200_{\pm 0.0106}$	$\mathbf{0.8120}_{\pm 0.0174}$	$0.6747_{\pm 0.0295}$	$0.7813_{\pm 0.0162}$	$0.7733_{\pm 0.0105}$

Continued on next page

Table 8 – *continued from previous page*

Dataset	GPT4TS	NuTime	Moment	Mantis	Utica
CricketX	$0.5231_{\pm 0.0143}$	$0.7453_{\pm 0.0039}$	$0.7915_{\pm 0.0192}$	$0.7966_{\pm 0.0090}$	$\mathbf{0.8120}_{\pm 0.0032}$
CricketY	$0.5419_{\pm 0.0090}$	$0.7479_{\pm 0.0104}$	$0.7487_{\pm 0.0271}$	$0.8060_{\pm 0.0141}$	$\mathbf{0.8555}_{\pm 0.0105}$
CricketZ	$0.5496_{\pm 0.0039}$	$0.7701_{\pm 0.0246}$	$0.8068_{\pm 0.0218}$	$0.8043_{\pm 0.0415}$	$\mathbf{0.8350}_{\pm 0.0134}$
Crop	$0.7311_{\pm 0.0033}$	$0.7574_{\pm 0.0022}$	$0.7170_{\pm 0.0071}$	$0.7444_{\pm 0.0039}$	$\mathbf{0.7618}_{\pm 0.0011}$
DiatomSizeReduction	$0.9401_{\pm 0.0361}$	$0.9314_{\pm 0.0425}$	$0.9325_{\pm 0.0576}$	$\mathbf{0.9684}_{\pm 0.0136}$	$0.9564_{\pm 0.0152}$
DistalPhalanxOutlineAgeGroup	$0.7218_{\pm 0.0042}$	$0.7386_{\pm 0.0150}$	$0.7362_{\pm 0.0272}$	$0.7698_{\pm 0.0259}$	$\mathbf{0.7722}_{\pm 0.0034}$
DistalPhalanxOutlineCorrect	$0.6932_{\pm 0.0084}$	$0.7705_{\pm 0.0302}$	$0.7452_{\pm 0.0200}$	$0.7717_{\pm 0.0158}$	$\mathbf{0.7995}_{\pm 0.0062}$
DistalPhalanxTW	$0.6978_{\pm 0.0125}$	$0.6978_{\pm 0.0216}$	$0.6451_{\pm 0.0300}$	$0.6954_{\pm 0.0231}$	$\mathbf{0.7218}_{\pm 0.0136}$
DodgerLoopDay	$0.5542_{\pm 0.0564}$	$0.5458_{\pm 0.0688}$	$0.5292_{\pm 0.0641}$	$0.6250_{\pm 0.0250}$	$\mathbf{0.6625}_{\pm 0.0102}$
DodgerLoopGame	$0.8551_{\pm 0.0192}$	$0.8213_{\pm 0.0649}$	$0.8599_{\pm 0.0233}$	$\mathbf{0.8841}_{\pm 0.0145}$	$0.8575_{\pm 0.0068}$
DodgerLoopWeekend	$\mathbf{0.9855}_{\pm 0.0000}$	$0.9783_{\pm 0.0126}$	$0.9783_{\pm 0.0126}$	$0.9783_{\pm 0.0000}$	$\mathbf{0.9855}_{\pm 0.0000}$
ECG200	$0.8467_{\pm 0.0551}$	$0.8700_{\pm 0.0100}$	$\mathbf{0.8867}_{\pm 0.0231}$	$0.8567_{\pm 0.0058}$	$0.8833_{\pm 0.0094}$
ECG5000	$\mathbf{0.9408}_{\pm 0.0019}$	$0.9379_{\pm 0.0001}$	$0.9325_{\pm 0.0003}$	$0.9335_{\pm 0.0081}$	$0.9403_{\pm 0.0007}$
ECGFiveDays	$0.8784_{\pm 0.0760}$	$\mathbf{0.9621}_{\pm 0.0084}$	$0.7944_{\pm 0.0373}$	$0.9148_{\pm 0.0292}$	$0.9241_{\pm 0.0054}$
Earthquakes	$0.7338_{\pm 0.0072}$	$0.7482_{\pm 0.0000}$	$\mathbf{0.7578}_{\pm 0.0166}$	$0.7482_{\pm 0.0000}$	$0.7506_{\pm 0.0034}$
ElectricDevices	$0.5776_{\pm 0.0173}$	$0.7112_{\pm 0.0109}$	$0.6666_{\pm 0.0204}$	$0.7454_{\pm 0.0049}$	$\mathbf{0.7700}_{\pm 0.0013}$
FaceAll	$0.7247_{\pm 0.0042}$	$\mathbf{0.8385}_{\pm 0.0098}$	$0.8012_{\pm 0.0080}$	$0.8308_{\pm 0.0031}$	$0.8008_{\pm 0.0254}$
FaceFour	$0.8295_{\pm 0.0301}$	$0.9318_{\pm 0.0227}$	$0.8371_{\pm 0.0853}$	$\mathbf{0.9773}_{\pm 0.0114}$	$0.8977_{\pm 0.0093}$
FacesUCR	$0.7932_{\pm 0.0047}$	$0.8846_{\pm 0.0127}$	$0.8350_{\pm 0.0014}$	$0.9148_{\pm 0.0071}$	$\mathbf{0.9260}_{\pm 0.0039}$
FiftyWords	$0.6615_{\pm 0.0058}$	$0.7875_{\pm 0.0083}$	$0.7861_{\pm 0.0222}$	$0.8139_{\pm 0.0083}$	$\mathbf{0.8411}_{\pm 0.0089}$
Fish	$0.8133_{\pm 0.0119}$	$0.9524_{\pm 0.0119}$	$0.9162_{\pm 0.0201}$	$\mathbf{0.9714}_{\pm 0.0099}$	$0.9486_{\pm 0.0093}$
FordA	$0.7207_{\pm 0.0798}$	$0.9225_{\pm 0.0095}$	$0.9202_{\pm 0.0271}$	$0.9343_{\pm 0.0046}$	$\mathbf{0.9391}_{\pm 0.0019}$
FordB	$0.6107_{\pm 0.0567}$	$0.7922_{\pm 0.0126}$	$0.8173_{\pm 0.0125}$	$0.7979_{\pm 0.0135}$	$\mathbf{0.8297}_{\pm 0.0046}$
FreezerRegularTrain	$0.8884_{\pm 0.0329}$	$0.9942_{\pm 0.0014}$	$0.9876_{\pm 0.0056}$	$0.9927_{\pm 0.0041}$	$\mathbf{0.9967}_{\pm 0.0012}$
FreezerSmallTrain	$0.6614_{\pm 0.0094}$	$0.9537_{\pm 0.0290}$	$0.8206_{\pm 0.0695}$	$\mathbf{0.9667}_{\pm 0.0098}$	$0.9213_{\pm 0.0121}$
Fungi	$0.7061_{\pm 0.0124}$	$0.7294_{\pm 0.0135}$	$0.6900_{\pm 0.0350}$	$0.7778_{\pm 0.0031}$	$\mathbf{0.7867}_{\pm 0.0134}$
GestureMidAirD1	$0.5615_{\pm 0.0353}$	$0.7410_{\pm 0.0222}$	$0.6487_{\pm 0.0311}$	$0.7692_{\pm 0.0407}$	$\mathbf{0.7872}_{\pm 0.0158}$
GestureMidAirD2	$0.4846_{\pm 0.0428}$	$0.6513_{\pm 0.0347}$	$0.5615_{\pm 0.0077}$	$0.6744_{\pm 0.0118}$	$\mathbf{0.7359}_{\pm 0.0130}$
GestureMidAirD3	$0.3308_{\pm 0.0400}$	$0.4359_{\pm 0.0089}$	$0.3615_{\pm 0.0277}$	$0.4077_{\pm 0.0231}$	$\mathbf{0.4897}_{\pm 0.0221}$
GesturePebbleZ1	$0.8217_{\pm 0.0034}$	$0.8953_{\pm 0.0058}$	$0.9109_{\pm 0.0089}$	$0.9322_{\pm 0.0089}$	$\mathbf{0.9477}_{\pm 0.0047}$
GesturePebbleZ2	$0.7553_{\pm 0.0159}$	$0.8861_{\pm 0.0219}$	$0.8861_{\pm 0.0063}$	$\mathbf{0.9241}_{\pm 0.0219}$	$0.8628_{\pm 0.0196}$
GunPoint	$0.8733_{\pm 0.0067}$	$0.9933_{\pm 0.0000}$	$0.9867_{\pm 0.0115}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9933_{\pm 0.0000}$
GunPointAgeSpan	$0.8755_{\pm 0.0174}$	$0.9916_{\pm 0.0018}$	$0.9662_{\pm 0.0048}$	$\mathbf{0.9979}_{\pm 0.0037}$	$0.9958_{\pm 0.0015}$
GunPointMaleVersusFemale	$0.9610_{\pm 0.0391}$	$0.9989_{\pm 0.0018}$	$0.9916_{\pm 0.0048}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9937_{\pm 0.0000}$
GunPointOldVersusYoung	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9820_{\pm 0.0048}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9947_{\pm 0.0054}$
Ham	$0.6698_{\pm 0.0240}$	$0.7238_{\pm 0.0190}$	$\mathbf{0.7746}_{\pm 0.0306}$	$0.7238_{\pm 0.0415}$	$0.7302_{\pm 0.0119}$
Haptics	$0.4426_{\pm 0.0167}$	$0.4870_{\pm 0.0142}$	$0.4167_{\pm 0.0179}$	$\mathbf{0.5227}_{\pm 0.0149}$	$0.5217_{\pm 0.0110}$
Herring	$0.6458_{\pm 0.0325}$	$0.6458_{\pm 0.0393}$	$0.5677_{\pm 0.0325}$	$0.6354_{\pm 0.0325}$	$\mathbf{0.6510}_{\pm 0.0266}$
HouseTwenty	$0.8123_{\pm 0.0049}$	$0.9272_{\pm 0.0097}$	$0.8992_{\pm 0.0084}$	$0.9804_{\pm 0.0049}$	$\mathbf{0.9832}_{\pm 0.0000}$
InsectEPGRegularTrain	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9331_{\pm 0.0267}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$
InsectEPGSmallTrain	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.8969_{\pm 0.0023}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$
InsectWingbeatSound	$0.6165_{\pm 0.0069}$	$0.6190_{\pm 0.0048}$	$0.6019_{\pm 0.0034}$	$0.5961_{\pm 0.0051}$	$\mathbf{0.6328}_{\pm 0.0075}$
ItalyPowerDemand	$0.9637_{\pm 0.0011}$	$\mathbf{0.9666}_{\pm 0.0020}$	$0.9624_{\pm 0.0006}$	$0.9624_{\pm 0.0015}$	$0.9543_{\pm 0.0016}$
LargeKitchenAppliances	$0.5156_{\pm 0.0285}$	$0.8240_{\pm 0.0116}$	$0.8693_{\pm 0.0232}$	$0.8516_{\pm 0.0111}$	$\mathbf{0.8951}_{\pm 0.0063}$
Lightning2	$0.7268_{\pm 0.0250}$	$0.7760_{\pm 0.0250}$	$0.7978_{\pm 0.0250}$	$0.8087_{\pm 0.0189}$	$\mathbf{0.8580}_{\pm 0.0077}$
Lightning7	$0.5982_{\pm 0.0079}$	$0.7032_{\pm 0.0519}$	$0.7717_{\pm 0.0209}$	$0.7397_{\pm 0.0362}$	$\mathbf{0.8310}_{\pm 0.0065}$
Mallat	$0.9200_{\pm 0.0258}$	$0.9508_{\pm 0.0191}$	$0.9205_{\pm 0.0260}$	$0.9404_{\pm 0.0072}$	$\mathbf{0.9598}_{\pm 0.0064}$
Meat	$0.6944_{\pm 0.2269}$	$\mathbf{0.9611}_{\pm 0.0255}$	$0.6444_{\pm 0.0674}$	$0.9333_{\pm 0.0289}$	$0.9222_{\pm 0.0157}$
MedicalImages	$0.6110_{\pm 0.0201}$	$0.7513_{\pm 0.0149}$	$0.7206_{\pm 0.0290}$	$0.7662_{\pm 0.0178}$	$\mathbf{0.7943}_{\pm 0.0075}$
MelbournePedestrian	$0.9351_{\pm 0.0037}$	$0.9601_{\pm 0.0019}$	$0.8795_{\pm 0.0055}$	$0.9552_{\pm 0.0010}$	$\mathbf{0.9692}_{\pm 0.0015}$
MiddlePhalanxOutlineAgeGroup	$0.6104_{\pm 0.0065}$	$0.5368_{\pm 0.0198}$	$0.6126_{\pm 0.0327}$	$0.5996_{\pm 0.0209}$	$\mathbf{0.6580}_{\pm 0.0031}$

Continued on next page

Table 8 – *continued from previous page*

Dataset	GPT4TS	NuTime	Moment	Mantis	Utica
MiddlePhalanxOutlineCorrect	$0.6518_{\pm 0.0436}$	$0.7938_{\pm 0.0248}$	$0.8076_{\pm 0.0034}$	$0.8339_{\pm 0.0139}$	$\mathbf{0.8625}_{\pm 0.0028}$
MiddlePhalanxTW	$\mathbf{0.5909}_{\pm 0.0000}$	$0.4632_{\pm 0.0150}$	$0.5606_{\pm 0.0037}$	$0.4827_{\pm 0.0300}$	$0.5866_{\pm 0.0061}$
MixedShapesSmallTrain	$0.8115_{\pm 0.0111}$	$0.9281_{\pm 0.0054}$	$0.8357_{\pm 0.0363}$	$\mathbf{0.9531}_{\pm 0.0038}$	$0.9434_{\pm 0.0037}$
MoteStrain	$0.8243_{\pm 0.0062}$	$\mathbf{0.9609}_{\pm 0.0037}$	$0.8341_{\pm 0.0264}$	$0.9068_{\pm 0.0174}$	$0.9297_{\pm 0.0078}$
NonInvasiveFetalECGThorax1	$0.7774_{\pm 0.0152}$	$\mathbf{0.9342}_{\pm 0.0076}$	$0.7535_{\pm 0.0237}$	$0.9086_{\pm 0.0063}$	$0.9318_{\pm 0.0054}$
NonInvasiveFetalECGThorax2	$0.8390_{\pm 0.0098}$	$0.9381_{\pm 0.0046}$	$0.7995_{\pm 0.0033}$	$0.9216_{\pm 0.0055}$	$\mathbf{0.9454}_{\pm 0.0036}$
OSULeaf	$0.4669_{\pm 0.0180}$	$0.8967_{\pm 0.0286}$	$0.8017_{\pm 0.0219}$	$\mathbf{0.9642}_{\pm 0.0024}$	$0.9229_{\pm 0.0186}$
OliveOil	$0.4889_{\pm 0.0385}$	$0.7778_{\pm 0.0385}$	$0.4667_{\pm 0.0667}$	$\mathbf{0.8889}_{\pm 0.0509}$	$0.6778_{\pm 0.0956}$
PhalangesOutlinesCorrect	$0.6437_{\pm 0.0041}$	$0.8263_{\pm 0.0031}$	$0.8030_{\pm 0.0047}$	$0.8162_{\pm 0.0128}$	$\mathbf{0.8400}_{\pm 0.0038}$
Phoneme	$0.1466_{\pm 0.0023}$	$0.2973_{\pm 0.0081}$	$0.2904_{\pm 0.0090}$	$\mathbf{0.3214}_{\pm 0.0059}$	$0.3087_{\pm 0.0051}$
PickupGestureWiimoteZ	$0.7133_{\pm 0.0231}$	$0.7267_{\pm 0.0115}$	$0.7200_{\pm 0.0000}$	$\mathbf{0.7600}_{\pm 0.0529}$	$0.7400_{\pm 0.0000}$
Plane	$0.9778_{\pm 0.0055}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9873_{\pm 0.0145}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$
PowerCons	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9889_{\pm 0.0111}$	$0.9167_{\pm 0.0111}$	$0.9944_{\pm 0.0096}$	$0.9926_{\pm 0.0026}$
ProximalPhalanxOutlineAgeGroup	$0.8016_{\pm 0.0203}$	$0.8553_{\pm 0.0102}$	$0.8455_{\pm 0.0123}$	$0.8455_{\pm 0.0246}$	$\mathbf{0.8780}_{\pm 0.0040}$
ProximalPhalanxOutlineCorrect	$0.8179_{\pm 0.0034}$	$\mathbf{0.9072}_{\pm 0.0060}$	$0.8259_{\pm 0.0221}$	$0.8763_{\pm 0.0209}$	$0.8923_{\pm 0.0113}$
ProximalPhalanxTW	$0.7967_{\pm 0.0102}$	$0.8179_{\pm 0.0056}$	$0.7870_{\pm 0.0102}$	$0.7772_{\pm 0.0314}$	$\mathbf{0.8212}_{\pm 0.0083}$
RefrigerationDevices	$0.4453_{\pm 0.0092}$	$\mathbf{0.5511}_{\pm 0.0227}$	$0.4996_{\pm 0.0336}$	$0.4916_{\pm 0.0111}$	$0.5396_{\pm 0.0163}$
Rock	$0.5933_{\pm 0.0503}$	$0.6800_{\pm 0.0200}$	$0.6733_{\pm 0.0611}$	$\mathbf{0.7133}_{\pm 0.0231}$	$0.5667_{\pm 0.0189}$
ScreenType	$0.3956_{\pm 0.0126}$	$0.5289_{\pm 0.0434}$	$0.4213_{\pm 0.0116}$	$0.4782_{\pm 0.0041}$	$\mathbf{0.5520}_{\pm 0.0058}$
ShakeGestureWiimoteZ	$0.6867_{\pm 0.0115}$	$0.9133_{\pm 0.0231}$	$0.9067_{\pm 0.0115}$	$\mathbf{0.9200}_{\pm 0.0000}$	$0.8267_{\pm 0.0340}$
ShapeletSim	$0.5074_{\pm 0.0128}$	$0.8852_{\pm 0.0534}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$
ShapesAll	$0.6833_{\pm 0.0029}$	$0.8856_{\pm 0.0067}$	$0.8039_{\pm 0.0108}$	$0.8906_{\pm 0.0059}$	$\mathbf{0.9067}_{\pm 0.0085}$
SmallKitchenAppliances	$0.5529_{\pm 0.0171}$	$0.7822_{\pm 0.0015}$	$0.6756_{\pm 0.0161}$	$0.7929_{\pm 0.0031}$	$\mathbf{0.8516}_{\pm 0.0063}$
SmoothSubspace	$0.8911_{\pm 0.0038}$	$0.9889_{\pm 0.0038}$	$0.7733_{\pm 0.0291}$	$\mathbf{0.9978}_{\pm 0.0038}$	$0.9867_{\pm 0.0054}$
SonyAIBORobotSurface1	$0.6267_{\pm 0.0215}$	$0.8525_{\pm 0.0189}$	$0.7604_{\pm 0.0204}$	$0.8142_{\pm 0.0113}$	$\mathbf{0.9567}_{\pm 0.0151}$
SonyAIBORobotSurface2	$0.8454_{\pm 0.0205}$	$0.8506_{\pm 0.0115}$	$0.8765_{\pm 0.0276}$	$0.8996_{\pm 0.0112}$	$\mathbf{0.9266}_{\pm 0.0082}$
Strawberry	$0.7261_{\pm 0.0788}$	$\mathbf{0.9712}_{\pm 0.0031}$	$0.9459_{\pm 0.0054}$	$0.9667_{\pm 0.0078}$	$0.9595_{\pm 0.0117}$
SwedishLeaf	$0.8203_{\pm 0.0305}$	$0.9573_{\pm 0.0024}$	$0.9376_{\pm 0.0070}$	$\mathbf{0.9712}_{\pm 0.0042}$	$0.9504_{\pm 0.0114}$
Symbols	$0.8003_{\pm 0.0243}$	$0.9779_{\pm 0.0053}$	$0.9481_{\pm 0.0192}$	$\mathbf{0.9889}_{\pm 0.0017}$	$0.9689_{\pm 0.0079}$
SyntheticControl	$0.9622_{\pm 0.0117}$	$0.9889_{\pm 0.0051}$	$0.9889_{\pm 0.0019}$	$\mathbf{0.9944}_{\pm 0.0038}$	$0.9900_{\pm 0.0000}$
ToeSegmentation1	$0.5687_{\pm 0.0051}$	$0.9459_{\pm 0.0101}$	$0.9284_{\pm 0.0483}$	$0.9664_{\pm 0.0067}$	$\mathbf{0.9752}_{\pm 0.0021}$
ToeSegmentation2	$0.6923_{\pm 0.0353}$	$0.8974_{\pm 0.0089}$	$\mathbf{0.9308}_{\pm 0.0204}$	$0.9231_{\pm 0.0077}$	$0.9000_{\pm 0.0218}$
Trace	$0.8567_{\pm 0.0577}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$
TwoLeadECG	$0.7934_{\pm 0.0272}$	$0.9104_{\pm 0.0273}$	$0.8879_{\pm 0.0509}$	$\mathbf{0.9941}_{\pm 0.0018}$	$0.9936_{\pm 0.0042}$
TwoPatterns	$0.9676_{\pm 0.0173}$	$0.9998_{\pm 0.0001}$	$0.9998_{\pm 0.0002}$	$0.9999_{\pm 0.0001}$	$\mathbf{1.0000}_{\pm 0.0000}$
UMD	$0.9699_{\pm 0.0080}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9815_{\pm 0.0200}$	$0.9907_{\pm 0.0040}$	$0.9977_{\pm 0.0033}$
UWaveGestureLibraryX	$0.7319_{\pm 0.0069}$	$0.8516_{\pm 0.0044}$	$0.8062_{\pm 0.0076}$	$\mathbf{0.8679}_{\pm 0.0028}$	$0.8655_{\pm 0.0033}$
UWaveGestureLibraryY	$0.6145_{\pm 0.0027}$	$0.7701_{\pm 0.0050}$	$0.7378_{\pm 0.0127}$	$0.7917_{\pm 0.0073}$	$\mathbf{0.7930}_{\pm 0.0021}$
UWaveGestureLibraryZ	$0.6224_{\pm 0.0217}$	$0.7813_{\pm 0.0073}$	$0.7570_{\pm 0.0158}$	$0.8053_{\pm 0.0011}$	$\mathbf{0.8119}_{\pm 0.0040}$
Wafer	$0.9948_{\pm 0.0022}$	$0.9987_{\pm 0.0003}$	$0.9982_{\pm 0.0004}$	$\mathbf{0.9991}_{\pm 0.0003}$	$0.9981_{\pm 0.0003}$
Wine	$0.5185_{\pm 0.0321}$	$\mathbf{0.8025}_{\pm 0.0466}$	$0.5000_{\pm 0.0000}$	$0.6605_{\pm 0.0107}$	$0.5864_{\pm 0.0231}$
WordSynonyms	$0.5397_{\pm 0.0096}$	$0.6917_{\pm 0.0065}$	$0.7079_{\pm 0.0220}$	$\mathbf{0.7429}_{\pm 0.0031}$	$0.7132_{\pm 0.0201}$
Worms	$0.4416_{\pm 0.0225}$	$0.7229_{\pm 0.0198}$	$0.7013_{\pm 0.0225}$	$0.7143_{\pm 0.0260}$	$\mathbf{0.7489}_{\pm 0.0324}$
WormsTwoClass	$0.6364_{\pm 0.0130}$	$0.7143_{\pm 0.0225}$	$0.7532_{\pm 0.0450}$	$\mathbf{0.8052}_{\pm 0.0344}$	$0.7922_{\pm 0.0106}$
Yoga	$0.7467_{\pm 0.0025}$	$0.8053_{\pm 0.0118}$	$0.8477_{\pm 0.0057}$	$0.8709_{\pm 0.0132}$	$\mathbf{0.8775}_{\pm 0.0062}$
# Wins	8	21	7	38	**60**
Avg. Acc.	0.7434	0.8387	0.7991	0.8504	**0.8569**
Avg. Rank	4.4248	2.2301	3.5133	1.9469	**1.4071**

Table 5: Comparison under linear probing regime on UEA. The best epoch performance is averaged over 3 random seeds and reported with the standard deviation. Bold indicates the best result per row. NuTime, Moment and Mantis results are from (Feofanov et al., 2025). We exclude datasets whose memory requirements exceed our available GPU memory (24 GB).

Dataset	**NuTime**	**Moment**	**Mantis**	**Utica**
ArticularyWordRecognition	**0.9933** ± 0.0000	0.9800 ± 0.0000	0.9930 ± 0.0030	0.9900 ± 0.0027
BasicMotions	**1.0000** ± 0.0000	0.9920 ± 0.0140	**1.0000** ± 0.0000	**1.0000** ± 0.0000
CharacterTrajectories	0.9649 ± 0.0038	0.9700 ± 0.0020	0.9400 ± 0.0010	**0.9847** ± 0.0006
Cricket	0.9907 ± 0.0080	0.9770 ± 0.0080	**1.0000** ± 0.0000	0.9907 ± 0.0066
ERing	**0.9716** ± 0.0057	0.9690 ± 0.0080	0.9410 ± 0.0100	0.9605 ± 0.0017
Eigenworms	0.7786 ± 0.0202	0.7350 ± 0.0160	0.7530 ± 0.0160	**0.7837** ± 0.0190
Epilepsy	**1.0000** ± 0.0000	0.9880 ± 0.0040	0.9950 ± 0.0040	0.9855 ± 0.0000
EthanolConcentration	**0.3942** ± 0.0158	0.2780 ± 0.0080	0.2700 ± 0.0100	0.3561 ± 0.0100
FingerMovements	0.5133 ± 0.0153	0.4970 ± 0.0230	0.5400 ± 0.0350	**0.5733** ± 0.0094
HandMovementDirection	0.3108 ± 0.0234	0.3150 ± 0.0080	0.2120 ± 0.0410	**0.3514** ± 0.0441
Handwriting	0.2035 ± 0.0082	0.2360 ± 0.0050	0.3390 ± 0.0100	**0.4227** ± 0.0095
JapaneseVowels	0.9405 ± 0.0072	0.8800 ± 0.0090	0.9630 ± 0.0060	**0.9784** ± 0.0022
LSST	0.5668 ± 0.0068	0.6210 ± 0.0010	0.6050 ± 0.0020	**0.6919** ± 0.0019
Libras	0.8852 ± 0.0064	0.8500 ± 0.0200	0.8980 ± 0.0080	**0.9000** ± 0.0046
NATOPS	0.8426 ± 0.0140	0.8220 ± 0.0240	0.9070 ± 0.0130	**0.9185** ± 0.0026
PhonemeSpectra	0.2588 ± 0.0065	0.2150 ± 0.0010	0.2730 ± 0.0080	**0.2982** ± 0.0007
RacketSports	0.9101 ± 0.0100	0.8200 ± 0.0100	**0.9230** ± 0.0040	0.8838 ± 0.0135
SelfRegulationSCP1	0.7702 ± 0.0120	0.7540 ± 0.0120	0.8040 ± 0.0110	**0.8612** ± 0.0126
SelfRegulationSCP2	0.5000 ± 0.0242	0.4960 ± 0.0200	0.4760 ± 0.0450	**0.5315** ± 0.0052
SpokenArabicDigits	0.8968 ± 0.0032	0.9350 ± 0.0020	0.8390 ± 0.0050	**0.9891** ± 0.0004
UWaveGestureLibrary	0.8792 ± 0.0018	0.8730 ± 0.0110	0.8140 ± 0.0100	**0.8802** ± 0.0103
Total Wins	5	0	3	**15**
Average Rank	2.45	3.29	2.67	**1.60**
Average Accuracy	0.7415	0.7240	0.7374	**0.7777**

Table 6: Comparison under the fine tuning regime on UEA. The best epoch performance is averaged over 3 random seeds and reported with the standard deviation. Bold indicates the best result per row. GPT4S, NuTime, Moment and Mantis results are from (Feofanov et al., 2025). We exclude datasets whose memory requirements exceed our available GPU memory (24 GB).

Dataset	GPT4TS	NuTime	Moment	Mantis	Utica
ArticularyWordRecognition	$0.9633_{\pm 0.0058}$	$0.9833_{\pm 0.0033}$	$0.9733_{\pm 0.0033}$	$\mathbf{0.9944}_{\pm 0.0019}$	$0.9900_{\pm 0.0033}$
BasicMotions	$0.8833_{\pm 0.0144}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$	$\mathbf{1.0000}_{\pm 0.0000}$
CharacterTrajectories	$0.9731_{\pm 0.0056}$	$0.9800_{\pm 0.0036}$	$0.9791_{\pm 0.0028}$	$0.9893_{\pm 0.0008}$	$\mathbf{0.9937}_{\pm 0.0007}$
Cricket	$0.9213_{\pm 0.0080}$	$0.9907_{\pm 0.0080}$	$0.9491_{\pm 0.0160}$	$0.9954_{\pm 0.0080}$	$\mathbf{1.0000}_{\pm 0.0000}$
ERing	$0.9074_{\pm 0.0098}$	$0.9506_{\pm 0.0077}$	$0.9358_{\pm 0.0077}$	$0.9827_{\pm 0.0043}$	$\mathbf{0.9864}_{\pm 0.0021}$
Epilepsy	$0.8092_{\pm 0.0233}$	$0.9952_{\pm 0.0042}$	$0.9976_{\pm 0.0042}$	$\mathbf{1.0000}_{\pm 0.0000}$	$0.9952_{\pm 0.0042}$
HandMovementDirection	$\mathbf{0.4685}_{\pm 0.0680}$	$0.3153_{\pm 0.0281}$	$0.3108_{\pm 0.0358}$	$0.3108_{\pm 0.0702}$	$0.4189_{\pm 0.0135}$
Handwriting	$0.2922_{\pm 0.0049}$	$0.2545_{\pm 0.0118}$	$0.3635_{\pm 0.0101}$	$\mathbf{0.4631}_{\pm 0.0147}$	$0.4451_{\pm 0.0226}$
LSST	$0.1080_{\pm 0.0611}$	$0.3335_{\pm 0.0282}$	$0.5827_{\pm 0.0162}$	$0.6035_{\pm 0.0254}$	$\mathbf{0.7091}_{\pm 0.0075}$
Libras	$0.7333_{\pm 0.0096}$	$0.8407_{\pm 0.0064}$	$0.7852_{\pm 0.0064}$	$0.8722_{\pm 0.0056}$	$\mathbf{0.9019}_{\pm 0.0032}$
RacketSports	$0.7961_{\pm 0.0174}$	$0.8794_{\pm 0.0137}$	$0.7895_{\pm 0.0132}$	$\mathbf{0.9189}_{\pm 0.0100}$	$\mathbf{0.9189}_{\pm 0.0082}$
UWaveGestureLibrary	$0.8396_{\pm 0.0065}$	$0.8646_{\pm 0.0188}$	$0.9156_{\pm 0.0094}$	$0.9281_{\pm 0.0125}$	$\mathbf{0.9406}_{\pm 0.0125}$
Wins	1	1	1	5	**7**
Avg Rank	4.50	3.17	3.42	1.75	**1.50**
Avg Accuracy	0.7246	0.7823	0.7985	0.8382	**0.8583**