Time Series Representations for Classification Lie Hidden in Pretrained Vision Transformers

Anonymous Author(s)
Affiliation
Address
email

Abstract

Time series classification is a fundamental task in healthcare and industry, yet the development of time series foundation models (TSFMs) remains limited by the scarcity of publicly available time series datasets. In this work, we propose **Ti**me **Vi**sion **T**ransformer (**TiViT**), a framework that converts time series into images to leverage the representational power of frozen Vision Transformers (ViTs) pretrained on large-scale image datasets. First, we show that the 2D patching of ViTs for time series can increase the number of label-relevant tokens and reduce the sample complexity. Second, we demonstrate that TiViT achieves state-of-the-art performance on time series classification benchmarks by utilizing the hidden representations of large OpenCLIP models. We explore the structure of TiViT representations and find that intermediate layers with high intrinsic dimension are the most effective for time series classification. Finally, we assess the alignment between TiViT and TSFM representations and identify a strong complementarity, with further performance gains achieved by combining their features.

1 Introduction

Inspired by the success of foundation models in natural language processing (NLP) and computer vision (CV), similar models have recently been developed for the analysis of time series following two different approaches. The first one is to pretrain time series foundation models (TSFMs) in a self-supervised way [Ansari et al., 2024, Das et al., 2024, Feofanov et al., 2025, Goswami et al., 2024, Lin et al., 2023] using a large-scale real-world time series dataset. The second one is to repurpose powerful foundation models from other domains, such as NLP [Jin et al., 2024, Zhou et al., 2023] and CV [Chen et al., 2024, Li et al., 2023b], for time series tasks. The idea behind these approaches is to benefit from the vast amount of samples that large vision and language models are trained on.

Time series can be transformed into images in many ways, including line plots, heatmaps, or spectrograms [Ni et al., 2025]. Wu et al. [2023] trained TimesNet end-to-end on heatmaps generated from time series. Li et al. [2023b] finetuned SwinTransformer on line plots of irregularly sampled time series. In contrast, we are the first to demonstrate that frozen vision foundation models such as OpenCLIP [Cherti et al., 2023, Ilharco et al., 2021], SigLIP 2 [Tschannen et al., 2025], and DINOv2 [Oquab et al., 2024], pretrained solely on natural images or image-text pairs, can be directly applied to time series classification without any pretraining or fine-tuning on time series data.

Our main contributions are as follows: (1) We show that pretrained ViTs of foundation models can be superior to TSFMs in time series classification. We achieve this by transforming time series into images and by further using hidden layer representations of vision models. (2) We propose a theoretical insight showing that image-based time series modeling can be efficient when used with Transformers since it reduces sample complexity during training. (3) We show that representations

Under review at the NeurIPS 2025 Workshop on Recent Advances in Time Series Foundation Models (BERT^{2}S). Do not distribute.

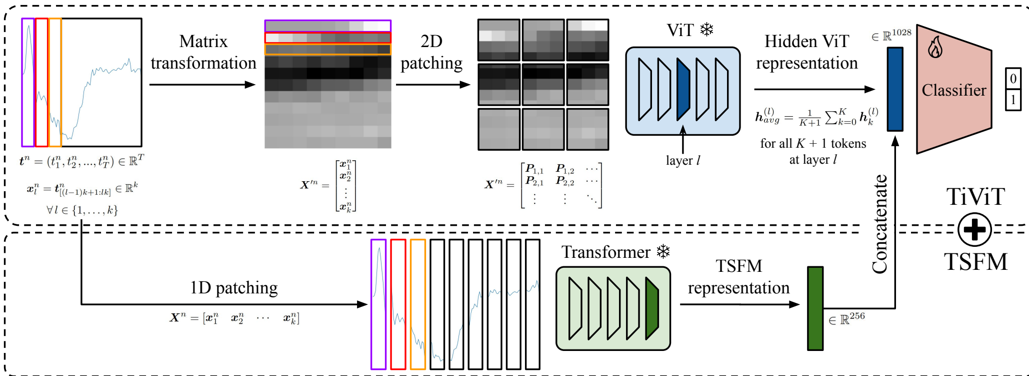


Figure 1: Illustration of TiViT on a time series sample from ECG200 [Olszewski, 2001]. We split the time series into segments and stack them to form a grayscale image. Then, we patch the image in 2D and feed it into a frozen ViT pretrained on large-scale image datasets. We average the hidden representations from a specific layer and pass them to a learnable classification head. Combining the representations of TiViT and TSFMs such as Mantis further improves classification accuracy.

from TSFMs and TiViT can be concatenated to provide an average improvement of +3% on 128 UCR [Dau et al., 2019] time series datasets, highlighting the complementarity of these models.

2 Modeling Time Series as Images

Although previous studies [Chen et al., 2024, Lin et al., 2024, Wu et al., 2023] have modeled time series as 2D matrices, there is no theoretical understanding of why such an approach may be beneficial in practice.

Theoretical Analysis We motivate the representation of time series as heatmap images by comparing the 1D and 2D patching of a periodic time series $\boldsymbol{t} \in \mathbb{R}^T$ (with $T = k^2$, period $p = k$). Our analysis focuses on the number of label-relevant tokens each method produces, which in turn determines the sample complexity of a Transformer [Li et al., 2023a]. The patching methods are:

- **1D patching**: The series $\boldsymbol{t}$ is split into k contiguous, non-overlapping tokens $\boldsymbol{x}_l \in \mathbb{R}^k$.
- **2D patching**: The series $\boldsymbol{t}$ is reshaped into a $k \times k$ matrix, then divided into k non-overlapping $\sqrt{k} \times \sqrt{k}$ patches, which are flattened to form tokens $\boldsymbol{x}'_{(i,j)} \in \mathbb{R}^k$.

Following the data model from Li et al. [2023a] for binary classification, we assume tokens are noisy versions of two class-specific patterns, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. A token is label-relevant if it is closer to the pattern of the correct class. The sample complexity of a shallow Transformer scales as $\mathcal{O}(1/\alpha_*^2)$ where α_* denotes the fraction of label-relevant tokens. Our key insight is that 2D patching increases this fraction, which we formalize in the following proposition. The full proof is postponed to Appendix B.2 and illustrated in Appendix B.3.

Proposition 1. *Let a time series $\boldsymbol{t} \in \mathbb{R}^T$ be composed of k segments, where each segment is either a non-discriminative pattern $\boldsymbol{\mu}_1$ or a label-relevant pattern $\boldsymbol{\mu}_2$. Let $|\{i : \boldsymbol{x}_i = \boldsymbol{\mu}_2\}| = n'$ and assume that $2\boldsymbol{x}' \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \leq ||\boldsymbol{\mu}_1||^2 - ||\boldsymbol{\mu}_2||^2$ whenever $|\{i : x'_i \in \boldsymbol{\mu}_2\}| \geq \sqrt{k}$. Then, it holds: $\alpha_*^{2D} \geq \alpha_*^{1D} = \frac{n'}{k}$, and the inequality is strict if $n' \bmod \sqrt{k} > 0$.*

Empirical Validation To verify our theoretical insight, we compare the two patching strategies on the UCR benchmark using a fixed Transformer architecture and pretraining paradigm. Details are provided in Appendix C. As shown in Table 1, 2D patching consistently outperforms 1D patching. Subsequently, we build on this idea of modeling time series as images and further leverage pretrained vision models for feature extraction.

Table 1: Comparison of patching strategies on the UCR benchmark.

Patching	Non-overlap		Overlap	
	1D	2D	1D	2D
Accuracy	76.4	76.8	76.6	**77.4**

3 TiViT: Time series classification with pretrained Vision Transformers

We introduce TiViT leveraging pretrained frozen ViTs from the vision or vision-language domain for time series classification. Figure 1 illustrates our approach. We are given a time series dataset $\mathcal{T} = \{\boldsymbol{t}^n | \boldsymbol{t}^n \in \mathbb{R}^{T\times D}\}_{n=1}^{N}$ containing N samples, each of length T and dimensionality D. The corresponding targets $\mathcal{Y} = \{y^n\}_{n=1}^{N}$ are labels $y^n \in \{1, ..., C\}$ from C classes.

Time series-to-image transformation Following the channel independence assumption proposed by Nie et al. [2023], we first split a multivariate time series $\boldsymbol{t}^n \in \mathbb{R}^{T\times D}$ into D univariate time series $\{\boldsymbol{t}_d^n \in \mathbb{R}^{T}\}_{d=1}^{D}$. We then normalize each univariate time series $\boldsymbol{t}_d^n$ using robust scaling, defined as: $\frac{\boldsymbol{t}_d^n - Q_2}{Q_3 - Q_1}$, where Q_1, Q_2, Q_3 are the first, second (median), and third quartiles, respectively. We apply padding at the beginning of each time series by replicating its first value and subsequently segment it into M patches $\{\boldsymbol{x}_m\}_{m=1}^{M}$ of size P. Given a patch length P and stride S, the total number of patches is: $M = \left\lfloor \frac{T-P}{S} \right\rfloor + 1$. We stack the patches to generate a 2D representation $\boldsymbol{X}' \in \mathbb{R}^{M\times P}$, which we then render into a grayscale image $\boldsymbol{X}' \in \mathbb{R}^{M\times P\times 3}$ by replicating its signals across three channels. To align with the square input resolution (R, R) expected by the ViT, we resize the image.

Time series classification We feed each grayscale image $\boldsymbol{X}'$ representing a univariate time series into a pretrained and frozen ViT v with L hidden layers. The ViT inherent 2D patching yields a sequence $\{\boldsymbol{x}'_k \in \mathbb{R}^{U^2}\}_{k=1}^{K}$ of flattened patches where (U, U) is the resolution per patch and $K = R^2/U^2$ is the resulting number of patches. ViTs generally prepend a classification token to this sequence. The ViT consumes all input tokens and produces a sequence of features at every layer: $v(\boldsymbol{X}') = \left\{[\boldsymbol{h}_0^{(l)}, \boldsymbol{h}_1^{(l)}, ..., \boldsymbol{h}_K^{(l)}]\right\}_{l=0}^{L}$. To obtain a single embedding vector $\boldsymbol{e}$ per image, we select a specific layer l and average its $K+1$ representations: $\boldsymbol{e} = \boldsymbol{h}_{avg}^{(l)} = \frac{1}{K+1}\sum_{k=0}^{K} \boldsymbol{h}_k^{(l)}$. For multivariate time series, we feed per-channel image representations $\{\boldsymbol{X}'_d\}_{d=1}^{D}$ separately into the ViT and concatenate the resulting embeddings for a specified layer: $\text{Concat}(\boldsymbol{e}_1, ..., \boldsymbol{e}_D)$. We only train a linear classifier on the ViT representations and their corresponding class labels.

4 Experimental evaluation

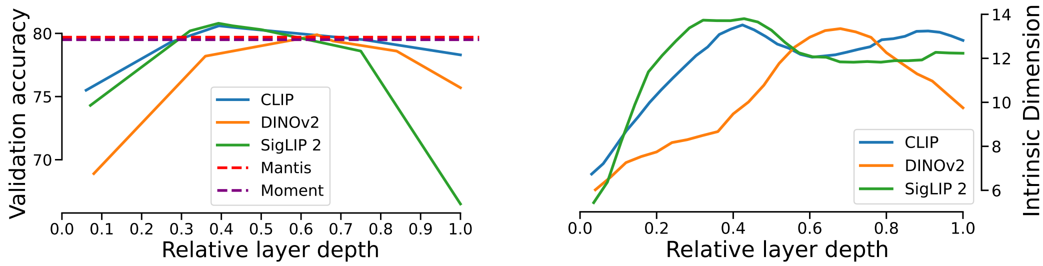
We evaluate TiViT with three different ViT backbones (CLIP [Radford et al., 2021, Cherti et al., 2023, Ilharco et al., 2021], SigLIP 2 [Tschannen et al., 2025], DINOv2 [Oquab et al., 2024]) on the UCR [Dau et al., 2019] and UEA [Bagnall et al., 2018] benchmarks for time series classification. We compare the performance of TiViT to two state-of-the-art TSFMs: Mantis [Feofanov et al., 2025] and Moment [Goswami et al., 2024]. Our experimental setup is detailed in Appendix D.

4.1 Transforming time series into images for ViT feature extraction

The performance of our time series-to-image transformation is sensitive to the patch size P, as extreme values can create redundant visual tokens during resizing to the ViT input resolution. To avoid a computationally expensive hyperparameter search for the optimal patch size P^* per dataset, we propose the heuristic $P = \sqrt{T}$ for any series of length T. This choice yields a square-shaped matrix representation prior to resizing, which minimizes distortion and preserves patch diversity (see Figure 5c). While an exhaustive search for P^* offers a marginal accuracy improvement (see Table 5a), our heuristic provides a strong baseline at a fraction of the computational cost. We further observe that introducing overlap between patches consistently boosts performance (see Table 5b). Consequently, the following experiments use a patch size of $P = \sqrt{T}$ and a stride of $S = P/10$.

4.2 Hidden representations are most effective in time series classification

While the final representations of ViTs typically capture high level semantics, intermediate layers encode lower level information [Dorszewski et al., 2025]. Our study reveals that the intermediate representations of ViTs are the most effective for downstream classification. In Figure 2a we report the classification performance of TiViT with pretrained ViTs from DINOv2, CLIP, and SigLIP 2 on the validation split of the UCR benchmark. For each dataset, we extract representations from the hidden layers of ViTs, average them, and train a linear classifier. The intermediate representations of ViTs, between 40% and 70% of the layer depth, achieve the highest classification accuracy.



(a) Validation accuracy

(b) Intrinsic dimension

Figure 2: (a) Validation accuracy and (b) Intrinsic dimensionality using hidden representations at different depth of ViTs (CLIP, DINOv2 and SigLIP 2). Results are averaged over 128 datasets from the UCR benchmark and three random seeds.

Intrinsic dimension To better understand the hidden representations of ViTs, we analyze their intrinsic dimension (see Figure 2b) and principal components (see Appendix E.4). Valeriani et al. [2023] have previously investigated the geometry of hidden representations of Transformers for in-domain vision and language applications. We measure the intrinsic dimension of ViTs applied on time series from the UCR archive using the DADApy [Glielmo et al., 2022] implementation of the TWO-NN estimator [Facco et al., 2017]. Figure 2b displays for three different ViT backbones the intrinsic dimensionality of their representations at varying layer depth. The best performing layers often exhibit the highest intrinsic dimensionality.

Benchmark A full comparison of TiViT and TSFMs on the UCR and UEA test set is reported in Table 2. The state-of-the-art TSFM Mantis achieves a linear classification accuracy of 80.1% on the UCR benchmark. Our statistical analysis with a paired t-test and a significance level of 0.05 confirms that TiViT significantly outperforms ($p = 0.03$) Mantis across the 128 datasets of the UCR benchmark, achieving 81.3% accuracy. We further extend our analysis to the classification of multivariate time series. TiViT reaches a classification accuracy of 72.0%, which is statistically on par with Mantis on the UEA benchmark.

Table 2: Classification accuracy of TSFMs and TiViT per benchmark.

Model	UCR	UEA
Moment	79.0	69.9
Mantis	80.1	72.4
TiViT *(Ours)*	81.3	72.0
TiViT + Moment *(Ours)*	82.5	72.6
TiViT + Mantis *(Ours)*	**83.0**	**73.7**

4.3 Alignment and fusion of TiViT and TSFM representations

We further explore the complementarity of TiViT and TSFM representations when concatenating their features for joint classification. As depicted in Table 2, the combination of TiViT and TSFM consistently improves the classification performance over any standalone model. While the combination of two TSFMs yields 81.5% accuracy, fusing TiViT with Moment and Mantis leads to even higher accuracies of 82.5% and 83.0%, respectively. These results underscore the potential of multimodal time series analysis. To uncover the differences between representations learned by ViTs and TSFMs, we additionally assess the alignment of their representation spaces using the mutual k-nearest neighbor metric [Huh et al., 2024] in Appendix E.5.

5 Conclusion

In this paper, we showed that modeling time series in 2D rather than 1D benefits time series classification with Transformers. Building on this insight, we introduced TiViT, leveraging large pretrained ViTs for feature extraction on images generated from time series. Our analysis revealed that the hidden representations of ViTs characterized by high intrinsic dimensionality are most effective in time series classification. TiViT significantly outperformed state-of-the-art TSFMs in time series classification on UCR, and reached comparable performance on UEA. Furthermore, we investigated multimodal time series analysis by merging the representations of TiViT and TSFMs, and achieved state-of-the-art results for foundation models in zero-shot and linear classification.

References

Ralph G. Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 437–442, 2013.

Abdul Fatir Ansari, Lorenzo Stella, Ali 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 Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

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.

Sathya Kamesh Bhethanabhotla, Omar Swelam, Julien Siems, David Salinas, and Frank Hutter. Mamba4Cast: Efficient zero-shot time series forecasting with state space models. *arXiv preprint arXiv:2410.09385*, 2024.

Defu Cao, Furong Jia, Sercan O. Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. TEMPO: Prompt-based generative pre-trained transformer for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. LLM4TS: Aligning pre-trained llms as data-efficient time-series forecasters. *ACM Transactions on Intelligent Systems and Technology*, 16(3), 2025.

Mouxiang Chen, Lefei Shen, Zhuo Li, Xiaoyun Joy Wang, Jianling Sun, and Chenghao Liu. VisionTS: Visual masked autoencoders are free-lunch zero-shot time series forecasters. *arXiv preprint arXiv:2408.17253*, 2024.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

Gari D. Clifford, Chengyu Liu, Benjamin Moody, Li-wei H. Lehman, Ikaro Silva, Qiao Li, Alistair E. Johnson, and Roger G. Mark. AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)*, pages 1–4, 2017.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, pages 10148–10167, 2024.

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. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

Teresa Dorszewski, Lenka Tětková, Robert Jenssen, Lars Kai Hansen, and Kristoffer Knutsen Wickstrøm. From colors to classes: Emergence of concepts in vision transformers. *arXiv preprint arXiv:2503.24071*, 2025.

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. In *International Conference on Learning Representations*, 2021.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7:12140, 2017.

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.

Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. UniTS: A unified multi-task time series model. In *Advances in Neural Information Processing Systems*, pages 140589–140631, 2024.

Aldo Glielmo, Iuri Macocco, Diego Doimo, Matteo Carli, Claudio Zeni, Romina Wild, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. DADApy: Distance-based analysis of data-manifolds in python. *Patterns*, 3 (10):100589, 2022.

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

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. MOMENT: A family of open time-series foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 16115–16152, 2024.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. Large language models are zero shot time series forecasters. In *Advances in Neural Information Processing Systems*, 2023.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognitio*, pages 16000–16009, 2022.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In *Proceedings of the 41st International Conference on Machine Learning*, pages 20617–20642, 2024.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021. Version 0.1. URL: `https://doi.org/10.5281/zenodo.5143773`.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. Time-LLM: Time series forecasting by reprogramming large language models. In *International Conference on Learning Representations*, 2024.

Bastiaan Kemp, Aeilko H. Zwinderman, Bert Tuk, Hilbert A. C. Kamphuisen, and Josefien J. L. Oberye. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194, 2000.

Christian Lessmeier, James K. Kimotho, Detmar Zimmer, and Walter Sextro. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. *PHM Society European Conference*, 3(1), 2016.

Hongkang Li, Meng Wang, Sijia Liu, and Pin-Yu Chen. A theoretical understanding of shallow vision transformers: Learning, generalization, and sample complexity. In *International Conference on Learning Representations*, 2023a.

Zekun Li, Shiyang Li, and Xifeng Yan. Time series as images: Vision transformer for irregularly sampled time series. In *Advances in Neural Information Processing Systems*, pages 49187–49204, 2023b.

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. *arXiv preprint arXiv:2310.07402*, 2023.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. SparseTSF: Modeling long-term time series forecasting with 1k parameters. In *Proceedings of the 41st International Conference on Machine Learning*, pages 30211–30226, 2024.

Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. In *2009 IEEE International Conference on Pervasive Computing and Communications*, 2009.

Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-MoE: Empowering time series foundation models with sparse mixture of experts. *arXiv preprint arXiv:2410.10469*, 2024a.

Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 32369–32399, 2024b.

Jingchao Ni, Ziming Zhao, ChengAo Shen, Hanghang Tong, Dongjin Song, Wei Cheng, Dongsheng Luo, and Haifeng Chen. Harnessing vision models for time series analysis: A survey. *arXiv preprint arXiv:2502.08869*, 2025.

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

Robert Thomas Olszewski. *Generalized feature extraction for structural pattern recognition in time-series data*. PhD thesis, Carnegie Mellon University, 2001.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

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. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763, 2021.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-Llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2024.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, pages 25278–25294, 2022.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. The geometry of hidden representations of large transformer models. In *Advances in Neural Information Processing Systems*, pages 51234–51252, 2023.

Yihang Wang, Yuying Qiu, Peng Chen, Kai Zhao, Yang Shu, Zhongwen Rao, Lujia Pan, Bin Yang, and Chenjuan Guo. ROSE: Register assisted general time series forecasting with decomposed frequency learning. *arXiv preprint arXiv:2405.17478*, 2024.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Hao Xue and Flora D. Salim. PromptCast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6851–6864, 2024.

Tian Zhou, Peisong Niu, Liang Sun, and Rong Jin. One fits all: Power general time series analysis by pretrained lm. In *Advances in Neural Information Processing Systems*, volume 36, pages 43322–43355, 2023.

Appendix

In Section A, we outline related work on time series foundation models and on transforming time series into images. In Section B, we summarize the theoretical analysis of Li et al. [2023a] on learning and generalization for Vision Transformers and detail our proof of label relevance for 2D patching. In Section C, we describe the model and pretraining setup used in our comparison of 1D and 2D patching for Transformers. In Section D, we explain the setup of our experimental evaluation of TiViT. In Section E, we further analyze the size and type of TiViT backbones. In Section F, we provide the benchmark results for each dataset from the UCR and UEA archive. Finally, we discuss the broader impacts of our work in Section G.

A Related work

Time series foundation models Recently, the research community has witnessed an impressive surge in the number and variety of TSFMs. At first, such models were based on repurposing large language models (LLMs) for time series tasks [Cao et al., 2024, Chang et al., 2025, Gruver et al., 2023, Jin et al., 2024, Xue and Salim, 2024, Zhou et al., 2023] by leveraging the ability of LLMs to efficiently handle text sequences. A different approach that gained in popularity later was to train TSFMs from the ground up on extensive and diverse datasets [Ansari et al., 2024, Bhethanabhotla et al., 2024, Das et al., 2024, Feofanov et al., 2025, Gao et al., 2024, Goswami et al., 2024, Lin et al., 2023, Liu et al., 2024a,b, Rasul et al., 2024, Wang et al., 2024]. While most of the models were designed for time series forecasting, several of them also specifically tackled time series classification [Feofanov et al., 2025, Gao et al., 2024, Goswami et al., 2024, Lin et al., 2023, Zhou et al., 2023]. These models are on par with or exceed the performance of other popular deep learning models proposed for time series classification, such as the famous TimesNet [Wu et al., 2023] architecture.

Transforming time series into images Time series can be transformed into images in many ways, either based on the 1D representation of the time series in the original (line plot) or transformed (frequency) space, or by using a 2D modeling (heatmap, Gramian angular field, recurrence plot) that stacks segments of the input time series based on a chosen periodicity. Vision models, often based on CNNs and their variations, were used on such image-based representations of time series since as early as 2013 (see Ni et al. [2025] for a recent survey). Most of them, however, are trained in a supervised way to fit a dataset at hand. This work explores how pretrained vision models can be used as powerful feature extractors without training or fine-tuning. Li et al. [2023b] showed that pretrained ViTs can be efficient in the classification of irregular time series from their line plot representations after full fine-tuning. In a similar vein, Chen et al. [2024] applied a masked auto-encoder with a pretrained frozen ViT to 2D transformed time series to perform univariate time series forecasting. Different from these works, we explain why vision models can be more efficient in time series analysis compared to Vanilla Transformers. Moreover, our TiViT model surpasses the performance of frontier TSFMs across a broad set of common classification benchmarks.

B Details on the theoretical analysis

We first review the shallow ViT and data model introduced by Li et al. [2023a] in their theoretical analysis of training a ViT. Their Theorem B.1 shows that the sample complexity for ViTs to achieve a zero generalization error is inversely correlated with the fraction of label-relevant tokens. Building on this insight, we provide a detailed proof of our Proposition 1 from the main paper, showing that 2D patching can increase the number of label-relevant tokens compared to 1D patching. We further illustrate our Proposition 1 with various examples of time series and their corresponding 2D representations.

B.1 Background

Model and setup Following the setup of Li et al. [2023a], we study a binary classification problem with N training samples $\{(\boldsymbol{X}^n, y^n)\}_{n=1}^{N}$. Each input $\boldsymbol{X}^n \in \mathbb{R}^{d\times L}$ contains L tokens $\{\boldsymbol{x}_1^n, \dots, \boldsymbol{x}_L^n\}$. Labels $y^n \in \{\pm 1\}$ are determined by majority vote over discriminative tokens. A simplified Vision

Table 3: Key Notations

Notation	Description
α_*	Fraction of label-relevant tokens
σ, δ, τ	Initialization/token noise parameters
κ	Minimum pattern distance
M	Total number of patterns

Transformer (ViT) [Dosovitskiy et al., 2021] model is defined as:

$$F(\boldsymbol{X}^n) = \frac{1}{|\mathcal{S}^n|} \sum_{l \in \mathcal{S}^n} \boldsymbol{a}_{(l)}^\top \text{ReLU}\left(\boldsymbol{W}_O \boldsymbol{W}_V \boldsymbol{X}^n \text{softmax}\left(\boldsymbol{X}^{n^\top} \boldsymbol{W}_K^\top \boldsymbol{W}_Q \boldsymbol{x}_l^n\right)\right),$$

where $\psi = (\boldsymbol{A} = \{\boldsymbol{a}_{(l)}\}_l, \boldsymbol{W}_O, \boldsymbol{W}_V, \boldsymbol{W}_K, \boldsymbol{W}_Q)$ are trainable parameters. The empirical risk minimization problem is:

$$\min_{\psi} f_N(\psi) = \frac{1}{N} \sum_{n=1}^{N} \max\left\{1 - y^n \cdot F(\boldsymbol{X}^n), 0\right\}.$$

Training uses mini-batch SGD with fixed output layer weights $\boldsymbol{A}$, following standard NTK initialization practices.

Data model Tokens $\boldsymbol{x}_l^n$ are noisy versions of M patterns $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M\}$, where $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ are discriminative. Label y^n depends on majority vote over tokens closest to $\boldsymbol{\mu}_1/\boldsymbol{\mu}_2$. Noise level τ satisfies $\tau < \kappa/4$, with $\kappa - 4\tau = \Theta(1)$.

Generalization of ViT We now recap the main results from Li et al. [2023a] from which we derive our result, along with the main notations in Table 3.

Assumption (Initial Model Conditions, [Li et al., 2023a]). *Initial weights $\boldsymbol{W}_V^{(0)}, \boldsymbol{W}_K^{(0)}, \boldsymbol{W}_Q^{(0)}$ satisfy:*

$$\|\boldsymbol{W}_V^{(0)} \boldsymbol{\mu}_j - \boldsymbol{p}_j\| \leq \sigma, \quad \|\boldsymbol{W}_K^{(0)} \boldsymbol{\mu}_j - \boldsymbol{q}_j\| \leq \delta, \quad \|\boldsymbol{W}_Q^{(0)} \boldsymbol{\mu}_j - \boldsymbol{r}_j\| \leq \delta,$$

for orthonormal bases $\mathcal{P}, \mathcal{Q}, \mathcal{R}$ and $\sigma = O(1/M), \delta < 1/2$.

Theorem (Generalization of ViT, [Li et al., 2023a]). *Under Assumption 1, with sufficient model width $m \gtrsim \epsilon^{-2} M^2 \log N$, fraction*

$$\alpha_* \geq \alpha_\#/(\epsilon_S e^{-(\delta+\tau)}(1 - (\sigma + \tau)),$$

and sample size

$$N \geq \Omega\left((\alpha_* - c'(1 - \zeta) - c''(\sigma + \tau))^{-2}\right),$$

SGD achieves zero generalization error after

$$T = \Theta\left(\frac{1}{(1 - \epsilon - (\sigma + \tau)M/\pi)\eta\alpha_*}\right)$$

iterations.

Proposition (Generalization without Self-Attention, [Li et al., 2023a]). *Without self-attention, achieving zero error requires $N \geq \Omega\left((\alpha_*(\alpha_* - \sigma - \tau))^{-2}\right)$, demonstrating ViT's sample complexity reduction by $1/\alpha_*^2$.*

B.2 Proof of label relevance in 2D patches

We remind Proposition 1 from the main paper and provide a detailed proof.

Proposition 1. *For an arbitrary $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^k$, let $\mathbf{t} = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \cdots \ \boldsymbol{x}_k]^\top \in \mathbb{R}^T$ where $\forall i \in [k], \boldsymbol{x}_i \in \mathbb{R}^k$ and either $\boldsymbol{x}_i = \boldsymbol{\mu}_1$ or $\boldsymbol{x}_i = \boldsymbol{\mu}_2$ with $\boldsymbol{\mu}_2$ being a label-relevant pattern. Let $|\{i : \boldsymbol{x}_i = \boldsymbol{\mu}_2\}| = n'$ and assume that $2\boldsymbol{x}' \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \leq ||\boldsymbol{\mu}_1||^2 - ||\boldsymbol{\mu}_2||^2$ whenever $|\{i : x_i' \in \boldsymbol{\mu}_2\}| \geq \sqrt{k}$. Then, it holds that*

$$\alpha_*^{2D} \geq \alpha_*^{1D} = \frac{n'}{k},$$

and the inequality is strict if n' mod $\sqrt{k} > 0$.

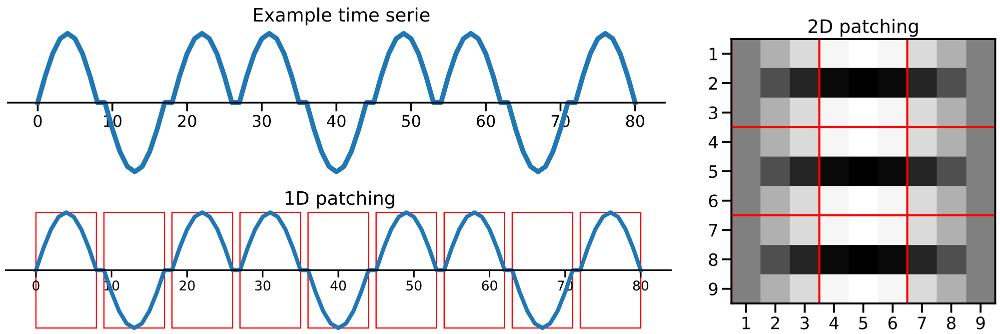


Figure 3: Benefits of 2D patching for time series. We consider a binary classification problem with two distinct patterns: a sine function over $[0, \pi]$, either positive or negative. Here, the negative sine function represents the label-relevant pattern. Tokens should cover at least $1/\sqrt{k}$ of the label-relevant pattern to be considered label-relevant, i.e., all tokens in 2D (red), only one third of tokens in 1D.

Proof. For a token $\boldsymbol{x}'^n$ to be label-relevant (aligned with $\boldsymbol{\mu}_2$), it must satisfy:

$$\|\boldsymbol{x}'^n - \boldsymbol{\mu}_2\| \leq \|\boldsymbol{x}'^n - \boldsymbol{\mu}_1\|.$$

Expanding both sides, we have that:

$$\|\boldsymbol{x}'^n\|^2 + 2\boldsymbol{x}'^n \cdot \boldsymbol{\mu}_1 + \|\boldsymbol{\mu}_1\|^2 \leq \|\boldsymbol{x}'^n\|^2 - 2\boldsymbol{x}'^n \cdot \boldsymbol{\mu}_2 + \|\boldsymbol{\mu}_2\|^2.$$

Regrouping the terms gives us the desired condition:

$$2\boldsymbol{x}'^n \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \leq ||\boldsymbol{\mu}_1||^2 - ||\boldsymbol{\mu}_2||^2. \tag{1}$$

Recall that n' denotes the number of segments of $\boldsymbol{\mu}_2$ in time series $\boldsymbol{t}$. Each such segment spans $\sqrt{k}$ tokens, contributing at least $\sqrt{k}$ elements to each of them. Under the assumption of the proposition, it implies (1) and makes each of these $\sqrt{k}$ tokens label-relevant.

We now need to carefully consider how the $\boldsymbol{\mu}_2$ segments can be placed within $\boldsymbol{t}$ to understand how many tokens become label-relevant thanks to each $\boldsymbol{\mu}_2$. We consider two cases: 1) $n' = c\sqrt{k}$ for some $c \in \mathbb{N}$ satisfying $n' \in (0, k]$, and 2) $n' = c\sqrt{k} + b$ for some $a, b \in \mathbb{N}$, $\sqrt{k} > b > 0$ such that $n' \in (0, k]$. In the first case, $\alpha_*^{1\text{D}} = c\sqrt{k}/k$. In the case of 2D patching, in the worst case, $\boldsymbol{\mu}_2$ segments can be placed such that they will contribute to $c\sqrt{k}$ tokens. In this case, $\alpha_*^{2\text{D}} \geq c\sqrt{k}/k$ and $\alpha_*^{1\text{D}} \leq \alpha_*^{2\text{D}}$. If n' is not a multiple of $\sqrt{k}$, the same analysis applies for the $c\sqrt{k}$ segments of $\boldsymbol{\mu}_2$. To account for the remainder b, we note that for any $b > 0$, in 2D case, it adds $\sqrt{k}$ label-relevant tokens to the fraction $\alpha_*^{2\text{D}}$ so that $\alpha_*^{2\text{D}} \geq \frac{c\sqrt{k}+\sqrt{k}}{k}$. In the case of 1D patching, $\alpha_*^{1\text{D}} = \frac{c\sqrt{k}+b}{k}$. Given that $b < \sqrt{k}$, this concludes the proof. $\square$

To better illustrate this proposition, we visualize it using a concrete example. We define $\boldsymbol{\mu}_1 = \sin(x)$ for $x \in [0, \pi]$ and let $\boldsymbol{\mu}_2 = -\boldsymbol{\mu}_1$. Figure 3 (more examples are provided in Appendix B.3) displays the input time series $\boldsymbol{t}$ with $k = 9$ and $n' = 3$. In this case, the assumption $2\boldsymbol{x}' \cdot (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \leq ||\boldsymbol{\mu}_1||^2 - ||\boldsymbol{\mu}_2||^2$ simplifies to $\boldsymbol{x}' \cdot \boldsymbol{\mu}_1 \leq 0$ and is verified for all tokens in 2D case and only for n' tokens in 1D case. On a higher level, this proposition formalizes the idea that having a discriminative signal spread across more tokens (each $\boldsymbol{\mu}_2$ contributes to $\sqrt{k}$ tokens in 2D case) makes it easier for a Transformer model to pick up this signal and to learn the classification task better. In the case of 1D patching, this signal is less spread, making it harder for the model to attend to important tokens during training.

B.3 Additional illustrations of Proposition 1

To illustrate the benefits of 2D modeling and patching, we present several examples of time series in Figure 4. We define $\boldsymbol{\mu}_1$ using functions such as log, cosine, and sine. We then set $\boldsymbol{\mu}_2 = \mathbf{1}_k$, $n' = 3$ and randomly shuffle $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ segments within the generated input time series.

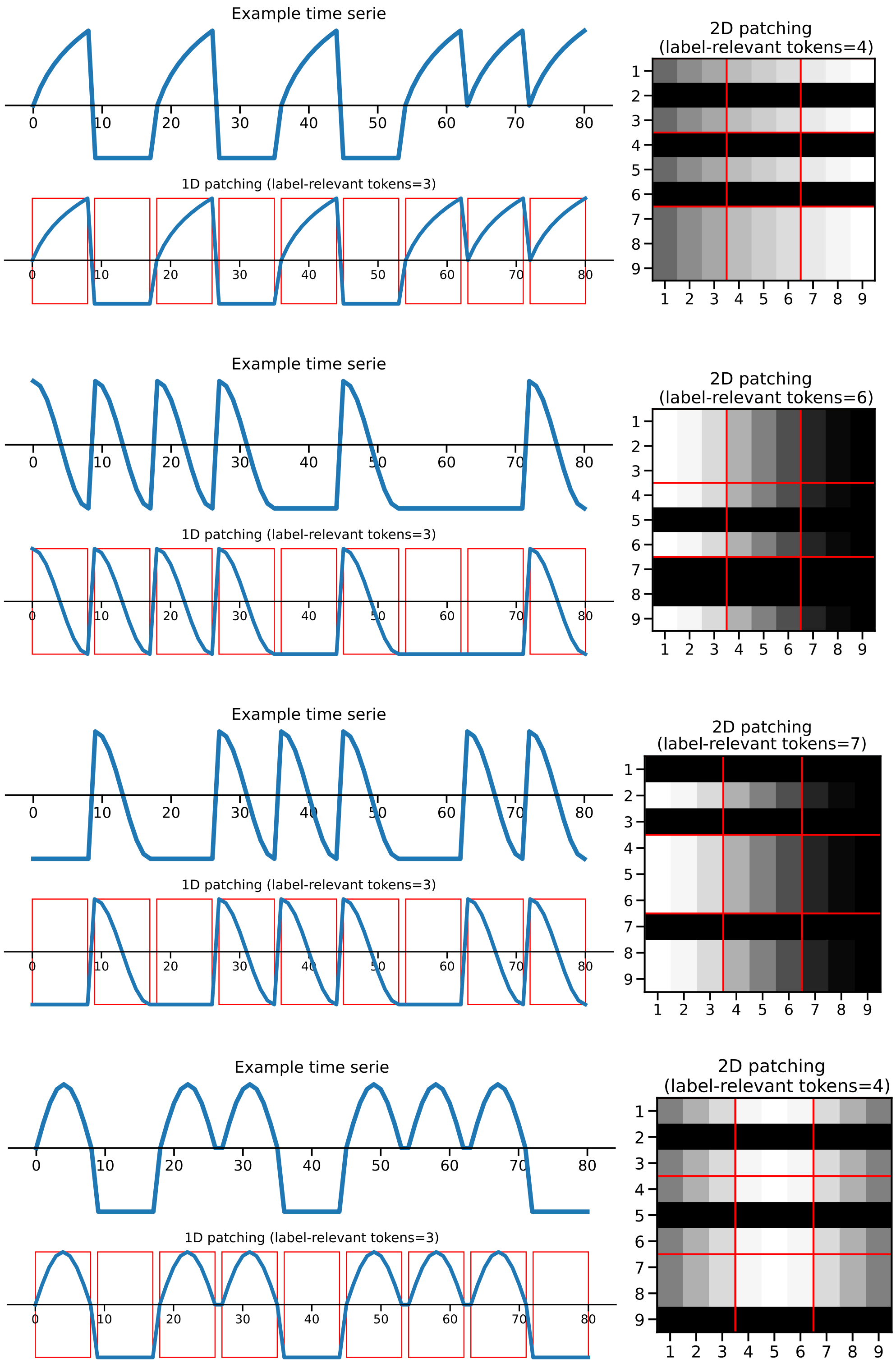


Figure 4: Illustration of Proposition 1 on more generated time series. In each example considered, 2D patching is more beneficial due the higher number of label-relevant tokens.

Table 4: Data used to pretrain Transformers for comparison of 1D and 2D patching.

Dataset	Number of examples	Prop. of taken examples
ECG	20835	45.7%
EMG	163	100%
Epilepsy	11480	100%
FD-A	10912	100%
FD-B	13619	100%
Gesture	1320	100%
HAR	20835	78.7%
SleepEEG	20836	4.5%

C Details on the comparison of 1D and 2D patching for Transformers

C.1 Architecture and pretraining

To evaluate the effect of 1D versus 2D patching on representations learned by Transformers, we fix the Transformer architecture and pretraining strategy, and only change the patching approach for generating input tokens. We adopt the setup of Feofanov et al. [2025] since their Transformer block implementation (ViTUnit class here) for time series classification is similar to the classical ViT. Specifically, the model comprises 6 Transformer layers, each with 8 attention heads and an embedding dimension of 256.

For pretraining, we employ contrastive learning following [Feofanov et al., 2025, He et al., 2020]. The augmentation technique to generate positive pairs is RandomCropResize with a crop rate varying within $[0\%, 20\%]$. All time series are resized to a fixed length $T = 512$ using interpolation.

We examine both non-overlapping and overlapping patches following [Goswami et al., 2024, Nie et al., 2023]. For non-overlapping 1D patching, we generate 32 patches of size 16. For non-overlapping 2D patching, we first arrange the 1D patches in a matrix of size 32×16 and then extract 32 patches of size 2×8. After flattening, we obtain 32 patches of size 16, similar to the 1D setting, but semantically different. For overlapping 1D patching, we apply a stride of 8, which yields 64 patches of size 16. For overlapping 2D patching, we rearrange these 1D patches again in a matrix of size 64×16 and then extract 32 patches of size 4×8. Flattening yields 32 patches of size 32.

C.2 Dataset

To pretrain the different models, we first generate a pretraining dataset from publicly available datasets that are not part of the evaluation benchmark. In detail, we consider a concatenation of the following datasets: ECG [Clifford et al., 2017], EMG [Goldberger et al., 2000], Epilepsy [Andrzejak et al., 2001], FD-A and FD-B [Lessmeier et al., 2016], Gesture [Liu et al., 2009], HAR [Anguita et al., 2013], SleepEEG [Kemp et al., 2000]. To reduce computation time, we construct a subset of the full dataset containing 100 000 samples, with a sufficiently balanced distribution across the individual source datasets. We give more details in Table 4 on how many samples were taken from each dataset to form the pretraining corpus.

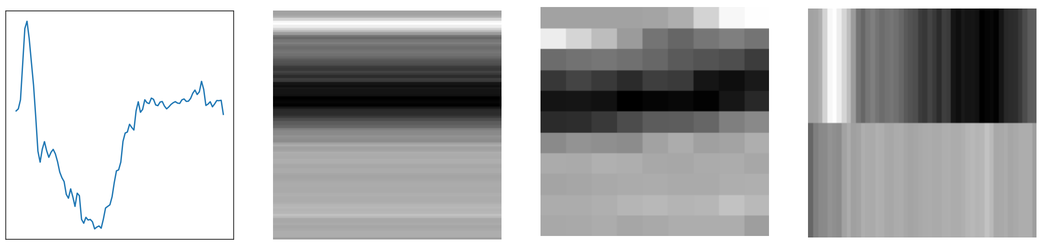
Table 5: Comparison of the effects on validation accuracy of (a) Patch size P and (b) Patch overlap. Results are averaged across the 128 datasets of UCR benchmark for 3 random seeds.

(a) Selecting patch size P

Patch size	$\sqrt{T}$	P^*
Val accuracy	78.2	**79.5**

(b) Effect of patch overlap on validation accuracy

Overlap	0.0	0.25	0.5	0.75	0.9	0.95
Val accuracy	78.2	79.3	80.2	80.0	**80.4**	80.0



(a) ECG200 sample (b) $P = 1$ (c) $P = \sqrt{T}$ (d) $P = \frac{T}{2}$

Figure 5: Effect of patch size P on the time series-to-image transformation on a sample from ECG200[Olszewski, 2001]. To match the ViT input resolution, a small patch size ($P = 1$) requires horizontal stretching, while a large patch size ($P = \frac{T}{2}$) requires vertical stretching. Both scenarios result in redundant tokens.

D Experimental setup

Datasets UCR [Dau et al., 2019] comprises 128 univariate time series datasets of varying sample size ($16 \leq N_{\text{train}} \leq 8926$) and series length ($15 \leq T \leq 2844$). UEA [Bagnall et al., 2018] consists of 30 multivariate time series datasets. Following Feofanov et al. [2025], we exclude three datasets (AtrialFibrillation, StandWalkJump, PenDigits) from UEA due to their short sequence length or small test size.

Vision Transformers Our study examines three differently pretrained ViTs. CLIP [Radford et al., 2021] performs contrastive learning of image and text encoders on image-text pairs. We reuse the ViT image encoders of OpenCLIP [Cherti et al., 2023, Ilharco et al., 2021] models trained with the LAION-2B English subset of LAION-5B [Schuhmann et al., 2022]. SigLIP 2 [Tschannen et al., 2025] adopts contrastive learning on image-text pairs, but with a Sigmoid loss, complemented by captioning-based pretraining, self-distillation, and masked prediction. In contrast, DINOv2 [Oquab et al., 2024] is solely pretrained on images through self-distillation with a student-teacher architecture and masked modeling. For each pretraining approach, we consider multiple vision model sizes (ViT-B, ViT-L, ViT-H) with varying layer depth (12, 24, and 32 layers).

Baselines We compare TiViT to two state-of-the-art TSFMs exclusively pretrained on time series. Mantis [Feofanov et al., 2025] is a Transformer model (8 M parameters) comprising 6 layers and 8 heads per layer, pretrained on 2 million time series with contrastive learning. Moment [Goswami et al., 2024] is a family of Transformers pretrained on 13 million time series with masked modeling. In our study, we consider Moment-base with 12 layers and 125 M parameters.

Implementation To assess the effectiveness of TiViT and TSFM representations in time series classification, we train a logistic regressor with the LBFGS solver per dataset. Our evaluation adheres to the standard train-test splits provided by the UCR and UEA archive and reserves 20% of the train split for validation. For the time series-to-image transformation, we resize the grayscale images to the resolution expected by the ViT with nearest interpolation and adjust the contrast with a factor of 0.8. All experiments can be performed on a single NVIDIA V100 GPU with 16 GB memory. Our results are averaged over three random seeds.

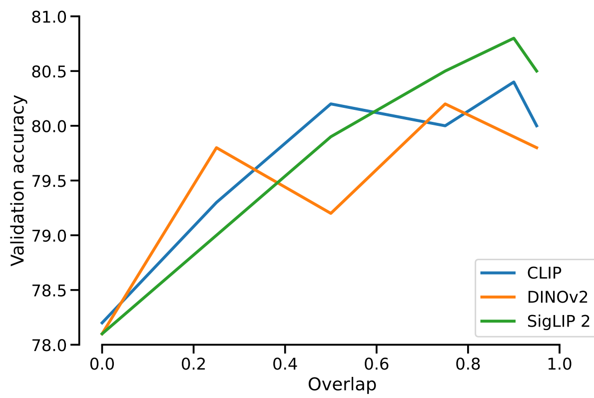


Figure 6: Effect of patch overlap on the classification accuracy of TiViT with different backbones.

Table 6: Linear classification with TiViT on UCR. For each model, we report the test accuracy achieved with the best performing hidden layer representation.

Model	Architecture	Layer	Parameters	Data	Accuracy
TiViT-DINOv2	ViT-L/14	15	178 M	LVD-142M	80.0
TiViT-SigLIP 2	SoViT-400m/14	10	138 M	WebLI (10B)	80.6
TiViT-CLIP	ViT-H/14	14	257 M	LAION-2B	**81.3**

E Additional analysis on TiViT

E.1 Patch size and overlap

In Section 4.1, we analyze the time series-to-image transformation for TiViT-CLIP and show that a patch size $P = \sqrt{T}$ and a stride $S = \frac{P}{10}$ yields high classification accuracy for any time series of length T. Figure 6 displays the effect of patch overlap for TiViT with CLIP, DINOv2, and SigLIP 2 backbones while fixing the patch size at $P = \sqrt{T}$. All versions of TiViT achieve high classification accuracy when utilizing an overlap of 0.9 (corresponding to stride $S = \frac{P}{10}$).

E.2 Different vision foundation models

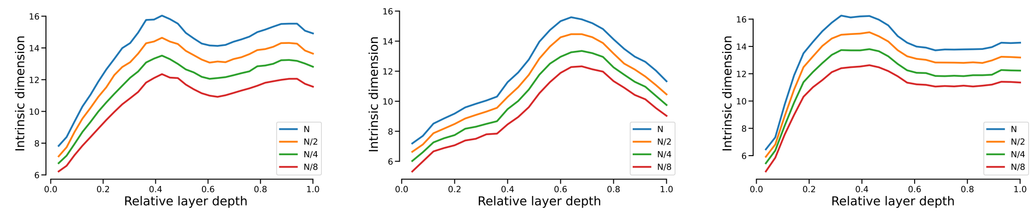
Table 6 displays the best performing hidden layers for various vision foundation models. CLIP and SigLIP 2, both optimized with a contrastive loss on image-text pairs, reach best performance in their earlier layers: layer 14 of 33 for CLIP (ViT-H) and layer 10 of 28 for SigLIP 2 (SoViT-400m). In contrast, DINOv2 (ViT-L) trained with contrastive learning and masked modeling on images only, reaches the highest classification accuracy with representations from a later layer (15 of 25). Our selection of architectures per pretraining paradigm ensures that TiViT exhibits a similar number of layers and parameters up to the best performing hidden layer. For each ViT, we determine the optimal

Table 7: Linear classification accuracy of TiViT on the UCR dataset with different ways of aggregating the hidden representations per layer. We report the total number of layers including the output layer and the index of the best performing layer starting from 0.

Model	# Layers	Average of tokens		CLS token	
		Layer	Acc	Layer	Acc
TiViT-DINOv2	25	15	80.0	17	79.1
TiViT-SigLIP 2	28	10	80.6	14	71.7
TiViT-CLIP	33	14	**81.3**	18	78.6

Table 8: Linear classification of TiViT-CLIP with varying size of the ViT backbone. For each model, we report the test accuracy on the UCR dataset achieved with the best performing hidden layer representation and the number of parameters up to this layer.

Architecture	Layer (total number)	Parameters	Accuracy
ViT-B/32	8 (13)	52 M	79.8
ViT-B/16	6 (13)	36 M	80.8
ViT-L/14	10 (24)	178 M	80.3
ViT-H/14	14 (32)	257 M	**81.3**



(a) TiViT-CLIP (b) TiViT-DINOv2 (c) TiViT-SigLIP 2

Figure 7: Intrinsic dimension of hidden representations per layer from CLIP, DINOv2, and SigLIP computed for subsamples of the dataset in $\{N, \frac{N}{2}, \frac{N}{4}, \frac{N}{8}\}$.

hidden layer based on its highest validation accuracy across the 128 datasets of the UCR benchmark. This best performing layer per ViT is consistently used in all subsequent experiments.

E.3 Aggregation of hidden token representations

As described in Section 3, we obtain a single embedding for each time series by averaging the ViT hidden representations in a particular layer. We now evaluate the performance of TiViT when using the CLS token from each layer instead. Table 7 compares the linear classification performance on the UCR dataset using either the CLS token or the mean of all tokens. To ensure a fair comparison, we determine the best performing layer for each approach based on the validation accuracy. Across all backbones, the CLS token consistently results in lower test accuracy, confirming our choice to use the mean hidden representation in TiViT. Interestingly, the best performing CLS tokens appear in later layers compared to the best performing mean tokens. Therefore, utilizing the mean representations does not only enhance classification accuracy, but also reduce computational cost.

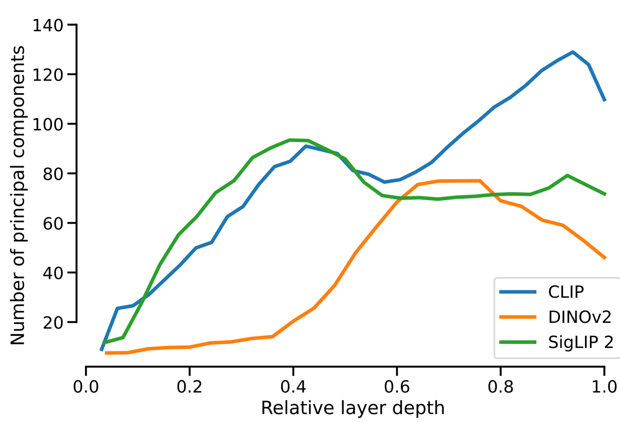
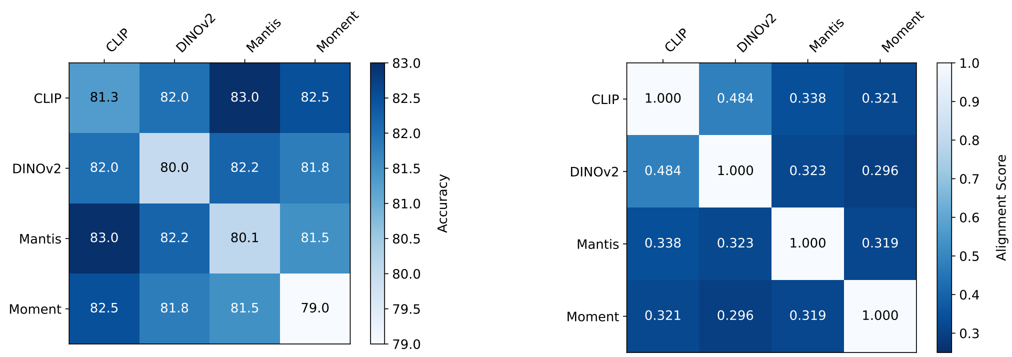


Figure 8: Number of principal components necessary to cover 95% of variance in the ViT representations per layer averaged across UCR datasets.

Table 9: Joint linear classification with TiViT and TSFMs on the UCR benchmark. We measure the alignment of the representation spaces using the mutual k-NN metric.

TiViT		TSFM		Joint	Alignment
CLIP	DINOv2	Mantis	Moment	Accuracy	Score
–	–	80.1	79.0	81.5	0.319
–	80.0	–	79.0	81.8	0.296
81.3	80.0	–	–	82.0	**0.484**
–	80.0	80.1	–	82.2	0.323
81.3	–	–	79.0	82.5	0.321
81.3	–	80.1	–	**83.0**	0.338



(a) Pairwise joint classification accuracy.

(b) Pairwise alignment score (mutual kNN).

Figure 9: The representations of frozen ViTs and TSFMs are concatenated and used in linear classification. Results are averaged over 128 datasets from the UCR benchmark.

E.4 Intrinsic dimension and principal components of hidden representations

The intrinsic dimension quantifies the minimum number of variables required to represent a local neighborhood of samples in the representation space. To estimate the intrinsic dimension, the TWO-NN estimator introduced by Facco et al. [2017] leverages the distance of each data point to its first and second nearest neighbor. As noted by the authors, a larger number of data points reduces the average distance to the second neighbor, and thus increases the intrinsic dimension. To mitigate this effect, they propose to subsample the dataset. Given a dataset of size N, we report the intrinsic dimension for $\frac{N}{4}$ subsamples in the main paper, which is in line with Valeriani et al. [2023]. In Figure 7, we compare the intrinsic dimension of average representations from hidden layers using N, $\frac{N}{2}$, $\frac{N}{4}$, and $\frac{N}{8}$ samples for estimation. The layer with the highest intrinsic dimension, which is central to our analysis, remains the same regardless of the subsampling ratio.

Since the intrinsic dimension only characterizes the local geometry of the representation space, we further provide a global analysis using principal components. Specifically, in Figure 8, we determine the number of principal components that are necessary to cover 95% of the variance in the data. For DINOv2, we observe a peak in the number of principal components in the middle layers that corresponds to the layers achieving the best classification accuracy. Interestingly, CLIP and SigLIP 2 exhibit two peaks in the number of principal components across the layers. The middle-layers corresponding to the first peak yield the highest time series classification accuracy.

E.5 Alignment and fusion of TiViT and TSFM representations

For each sample in the dataset, we find the $k = 10$ nearest neighbors in the embedding space of two different models and measure the intersection between the two neighbor sets. The final alignment score between two models is an average across all samples from the UCR benchmark. Table 9 presents the alignment scores for CLIP, DINOv2, Mantis, and Moment. Interestingly, the alignment score of the two TSFMs is relatively low. We hypothesize that this discrepancy arises from their different pretraining paradigms: Mantis is trained contrastively while Moment is trained with masked

Table 10: Linear classification accuracy of TiViT with varying MAE backbone size and aggregation of hidden representations per layer. We report the total number of layers including the output layer and the index of the best performing layer starting from 0.

Architecture	# Layers	Average of tokens		CLS token	
		Layer	Acc	Layer	Acc
MAE Base	13	8	72.7	9	73.8
MAE Large	25	14	74.3	18	75.6
MAE Huge	33	20	75.9	20	**76.7**

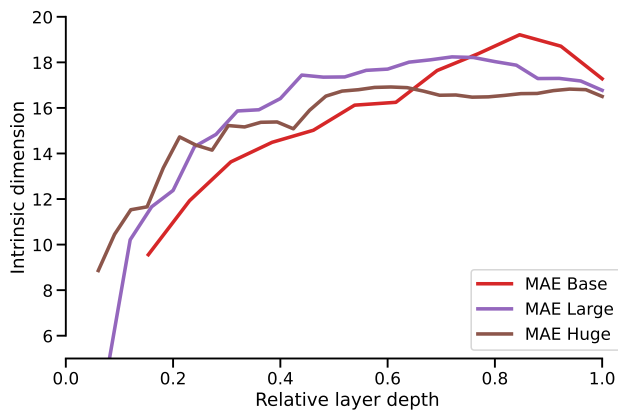


Figure 10: Intrinsic dimensionality of CLS tokens per MAE layer averaged across UCR datasets.

modeling. A similarly low alignment score is observed between any TiViT and TSFM, which we attribute to their domain gap. TiViT and Mantis extract different representations for the same time series, which is beneficial for joint classification. The highest alignment is measured between TiViT-CLIP and TiViT-DINOv2, both of which are pretrained contrastively on image datasets. Figure 9 is an additional visualization of the pairwise scores as heatmaps.

E.6 Size of ViT backbone

We report the performance of TiViT with CLIP ViT-H backbone in Section 4.2 of the main paper. Table 8 provides a detailed analysis of how the performance of TiViT varies with the size of the ViT backbone, including ViT-B (with two patch sizes), ViT-L, and ViT-H. Remarkably, with only 6 Transformer layers from ViT-B, TiViT achieves an accuracy of 80.8%. While matching the number of Transformer layers in Mantis, TiViT surpasses Mantis (80.1%) in classification accuracy. However, the hidden dimensionality is higher for the ViT-B backbone used in TiViT. By utilizing a larger backbone, specifically 14 hidden layers of ViT-H/14, we achieve the highest accuracy of 81.3%, significantly outperforming conventional TSFMs.

E.7 Masked autoencoder backbone

In the main paper, we analyze the reusability of ViT backbones from CLIP [Radford et al., 2021, Schuhmann et al., 2022], DINOv2 [Oquab et al., 2024], and SigLIP 2 [Tschannen et al., 2025] in time series classification. In contrast, Chen et al. [2024] repurpose Masked Autoencoders (MAEs) [He et al., 2022] for time series forecasting. To enable a direct comparison, we now utilize the hidden representations of MAE Base, Large, and Huge in time series classification.

Our analysis in Table 10 shows that for MAEs using the CLS token yields better performance in time series classification than averaging token representations. Moreover, Table 10 presents a comparison across MAEs of different sizes, showing that larger backbones consistently achieve higher accuracy. Different from contrastively pretrained models, summarized in Table 6 of the main paper, the best representations for time series classification with MAE lie in later layers. We further observe that the hidden representations of the later MAE layers up to the output layer perform similar in time

Table 11: Linear classification accuracy on UCR subsets (left) and classifier comparison (right). We consider logistic regression (LR), nearest centroid (NC), and random forest (RF).

	UCR subsets				Classifier comparison		
Model	Smallest	Largest	Shortest	Longest	LR	NC	RF
Moment	85.7	85.5	86.9	65.8	79.0	68.6	75.6
Mantis	86.6	82.8	88.1	70.5	80.1	71.2	77.5
TiViT *(Ours)*	89.8	85.3	87.5	75.0	81.3	71.6	77.4
TiViT + Moment *(Ours)*	89.9	**87.1**	**88.8**	74.9	82.5	73.3	79.4
TiViT + Mantis *(Ours)*	**90.9**	86.2	**88.8**	**77.7**	**83.0**	**73.4**	**79.8**

series classification, while there is a significant gap between hidden representations and output representations for TiViT-CLIP (see Figure 2a in the main paper). Figure 10 illustrates the intrinsic dimension of the CLS tokens per layer averaged across the UCR datasets. We observe that the intrinsic dimension increases up to 60% of the layer depth, while the later layers mostly exhibit a similar intrinsic dimension, explaining their similar classification performance.

It is worth noting that MAE has only been pretrained on ImageNet-1k [Deng et al., 2009] with 1.5 million samples, whereas CLIP has been pretrained on the significantly larger LAION-2B [Schuhmann et al., 2022] dataset with 2 billion samples. We hypothesize that being exposed to a larger set of images during training enhances the capacity of a vision model to extract discriminative patterns from 2D time series representations.

E.8 UCR subsets and classifier comparison

In Section 4.2, we report the performance of TiViT across all 128 UCR datasets. To further explore its capabilities, we now select four UCR subsets: 10 datasets with the fewest training samples ($16 \leq N_{train} \leq 20$), the most training samples ($1000 \leq N_{train} \leq 8926$), the shortest time series ($15 \leq T \leq 80$), and the longest time series ($1500 \leq T \leq 2844$). The results are displayed in Table 11. TiViT significantly outperforms Mantis on subsets with a small training set (89.8% vs. 86.6%) and long time series (75.0% vs. 70.5%). These findings demonstrate that TiViT excels in generalizing from limited training data and in modeling long-range dependencies. On the remaining two subsets, TiViT is on par with TSFMs. Combining the representations of TiViT and TSFMs achieves the highest classification accuracy across all subsets, once again underscoring their complementarity.

While the previous experiments require to train a logistic regressor for classification, we finally investigate the effectiveness of TiViT in zero-shot classification. Here, we employ a nearest centroid classifier, where each class is represented by the centroid of its representations, and samples are assigned to the class of their nearest centroid. On the UCR benchmark, TiViT achieves a zero-shot classification accuracy of 71.6%. Our approach is on par with Mantis (71.2%) and outperforms Moment (68.6%), highlighting the ability of TiViT to extract generalizable representations. We further merge the representations of TiViT and Mantis, reaching a zero-shot accuracy of 73.4%. Following Feofanov et al. [2025], we also adopt a random forest classifier. We observe that TiViT performs on par with Mantis, and that once again combining the representation of both models achieves state-of-the-art classification performance. Feofanov et al. [2025] have demonstrated that Mantis surpasses other TSFMs such as NuTime [Lin et al., 2023] when evaluated with a random forest classifier. This conclusion can now be extended to TiViT.

F Detailed results on UCR and UEA benchmarks

In the main paper, we report the average accuracy of TiViT and TSFM across 128 univariate datasets from the UCR archive and 27 multivariate datasets from the UEA archive. Here, we report the full linear classification benchmark with accuracy scores for Mantis, Moment, TiViT, and their combinations on each dataset. Table 12 presents the performance on the UCR dataset, while Table 13 reports the results on the UEA dataset. Additionally, Table 14 provides the mean rank of all five methods on both benchmarks. If multiple element share the same rank, we assign them the lowest rank in the group.

Table 12: Classification accuracy for 128 univariate datasets from the UCR benchmark. We report the mean and standard deviation across three random seeds.

Dataset	Moment	Mantis	TiViT	TiViT + Moment	TiViT + Mantis
ACSF1	0.673 ± 0.012	0.667 ± 0.021	**0.777** ± 0.015	**0.777** ± 0.012	0.763 ± 0.021
Adiac	0.731 ± 0.003	0.728 ± 0.010	0.695 ± 0.017	**0.740** ± 0.005	0.714 ± 0.003
AllGestureWiimoteX	0.680 ± 0.004	0.666 ± 0.007	0.653 ± 0.016	**0.702** ± 0.002	0.673 ± 0.019
AllGestureWiimoteY	0.711 ± 0.024	0.699 ± 0.007	0.715 ± 0.010	0.733 ± 0.013	**0.740** ± 0.010
AllGestureWiimoteZ	0.583 ± 0.013	0.650 ± 0.004	0.649 ± 0.017	0.664 ± 0.011	**0.667** ± 0.019
ArrowHead	0.804 ± 0.012	0.745 ± 0.007	0.806 ± 0.045	**0.840** ± 0.023	0.825 ± 0.035
BME	0.900 ± 0.075	0.987 ± 0.012	**0.998** ± 0.004	0.987 ± 0.018	0.996 ± 0.008
Beef	**0.756** ± 0.038	0.700 ± 0.033	0.733 ± 0.033	**0.756** ± 0.038	0.733 ± 0.033
BeetleFly	0.833 ± 0.029	0.900 ± 0.000	0.900 ± 0.050	0.883 ± 0.029	**0.933** ± 0.029
BirdChicken	0.850 ± 0.087	**0.933** ± 0.076	0.850 ± 0.087	0.850 ± 0.087	0.850 ± 0.087
CBF	0.943 ± 0.012	0.994 ± 0.010	**0.999** ± 0.001	0.998 ± 0.003	**0.999** ± 0.001
Car	0.817 ± 0.000	0.794 ± 0.051	0.794 ± 0.010	0.806 ± 0.025	**0.822** ± 0.025
Chinatown	0.966 ± 0.009	0.962 ± 0.003	0.965 ± 0.009	**0.976** ± 0.012	0.970 ± 0.007
ChlorineConcentration	0.723 ± 0.001	0.643 ± 0.004	0.721 ± 0.011	**0.739** ± 0.016	0.737 ± 0.009
CinCECGTorso	0.733 ± 0.031	0.737 ± 0.004	**0.895** ± 0.013	0.863 ± 0.019	**0.895** ± 0.012
Coffee	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000
Computers	0.712 ± 0.036	0.735 ± 0.021	0.748 ± 0.016	**0.772** ± 0.024	0.767 ± 0.012
CricketX	0.706 ± 0.020	0.726 ± 0.015	0.763 ± 0.010	0.755 ± 0.005	**0.766** ± 0.011
CricketY	0.693 ± 0.018	0.732 ± 0.017	0.767 ± 0.011	**0.779** ± 0.007	0.777 ± 0.011
CricketZ	0.740 ± 0.016	0.721 ± 0.009	0.773 ± 0.015	0.779 ± 0.012	**0.797** ± 0.017
Crop	0.709 ± 0.003	0.695 ± 0.001	0.673 ± 0.003	**0.712** ± 0.002	0.707 ± 0.003
DiatomSizeReduction	0.900 ± 0.030	0.881 ± 0.032	**0.938** ± 0.048	0.932 ± 0.049	**0.938** ± 0.048
DistalPhalanxOutlineAgeGroup	0.743 ± 0.011	**0.746** ± 0.017	0.715 ± 0.004	0.724 ± 0.011	0.700 ± 0.011
DistalPhalanxOutlineCorrect	**0.762** ± 0.017	0.728 ± 0.007	0.755 ± 0.006	0.756 ± 0.014	0.743 ± 0.007
DistalPhalanxTW	0.643 ± 0.004	**0.698** ± 0.007	0.652 ± 0.027	0.688 ± 0.011	0.640 ± 0.007
DodgerLoopDay	0.467 ± 0.031	**0.504** ± 0.014	0.475 ± 0.022	0.500 ± 0.033	0.496 ± 0.031
DodgerLoopGame	0.720 ± 0.051	0.749 ± 0.008	0.768 ± 0.045	0.756 ± 0.053	**0.783** ± 0.040
DodgerLoopWeekend	**0.971** ± 0.000	0.964 ± 0.000	0.957 ± 0.000	0.969 ± 0.004	**0.971** ± 0.000
ECG200	0.843 ± 0.006	**0.853** ± 0.012	0.837 ± 0.012	**0.853** ± 0.015	0.847 ± 0.012
ECG5000	0.933 ± 0.005	0.924 ± 0.003	0.936 ± 0.002	0.937 ± 0.002	**0.939** ± 0.002
ECGFiveDays	0.957 ± 0.007	0.977 ± 0.004	0.983 ± 0.001	**0.995** ± 0.001	0.986 ± 0.001
EOGHorizontalSignal	0.561 ± 0.008	0.562 ± 0.018	0.603 ± 0.014	0.644 ± 0.015	**0.649** ± 0.006
EOGVerticalSignal	0.463 ± 0.012	**0.507** ± 0.007	0.465 ± 0.009	0.493 ± 0.014	0.491 ± 0.008
Earthquakes	**0.722** ± 0.034	0.719 ± 0.007	0.707 ± 0.015	0.717 ± 0.032	**0.722** ± 0.029
ElectricDevices	0.631 ± 0.008	0.701 ± 0.003	**0.762** ± 0.002	0.744 ± 0.005	0.751 ± 0.002
EthanolLevel	**0.631** ± 0.010	0.439 ± 0.010	0.579 ± 0.023	0.614 ± 0.007	0.583 ± 0.012
FaceAll	0.733 ± 0.014	**0.794** ± 0.010	0.745 ± 0.007	0.747 ± 0.004	0.766 ± 0.006
FaceFour	0.784 ± 0.041	**0.958** ± 0.007	0.777 ± 0.093	0.811 ± 0.046	0.879 ± 0.046
FacesUCR	0.791 ± 0.009	0.886 ± 0.005	0.863 ± 0.011	0.870 ± 0.011	**0.902** ± 0.009
FiftyWords	0.727 ± 0.021	0.740 ± 0.013	0.747 ± 0.011	0.767 ± 0.006	**0.777** ± 0.012
Fish	0.947 ± 0.003	0.958 ± 0.007	0.949 ± 0.006	0.958 ± 0.012	**0.970** ± 0.009
FordA	0.914 ± 0.003	0.911 ± 0.002	0.909 ± 0.004	**0.928** ± 0.005	0.914 ± 0.005
FordB	0.800 ± 0.005	0.769 ± 0.002	**0.801** ± 0.004	0.796 ± 0.011	0.795 ± 0.005
FreezerRegularTrain	0.973 ± 0.012	0.976 ± 0.012	**0.995** ± 0.001	**0.995** ± 0.004	**0.995** ± 0.002
FreezerSmallTrain	0.840 ± 0.012	0.870 ± 0.020	**0.981** ± 0.004	0.970 ± 0.008	0.980 ± 0.005
Fungi	0.753 ± 0.033	0.810 ± 0.025	0.794 ± 0.020	0.810 ± 0.020	**0.815** ± 0.025
GestureMidAirD1	0.656 ± 0.012	0.669 ± 0.023	0.726 ± 0.025	0.721 ± 0.018	**0.756** ± 0.031
GestureMidAirD2	0.567 ± 0.016	0.574 ± 0.032	0.646 ± 0.043	0.628 ± 0.019	**0.669** ± 0.028
GestureMidAirD3	0.359 ± 0.019	0.385 ± 0.013	0.474 ± 0.009	0.441 ± 0.018	**0.479** ± 0.035
GesturePebbleZ1	0.893 ± 0.015	0.911 ± 0.003	0.891 ± 0.003	0.924 ± 0.010	**0.932** ± 0.007
GesturePebbleZ2	0.846 ± 0.018	**0.905** ± 0.006	0.835 ± 0.011	0.876 ± 0.032	0.892 ± 0.011
GunPoint	0.984 ± 0.027	0.987 ± 0.007	0.991 ± 0.004	**0.993** ± 0.007	**0.993** ± 0.007
GunPointAgeSpan	0.980 ± 0.008	**0.998** ± 0.002	0.997 ± 0.000	0.995 ± 0.002	0.997 ± 0.000
GunPointMaleVersusFemale	**1.000** ± 0.000	0.999 ± 0.002	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000
GunPointOldVersusYoung	**1.000** ± 0.000	**1.000** ± 0.000	0.989 ± 0.004	**1.000** ± 0.000	**1.000** ± 0.000
Ham	**0.752** ± 0.025	0.667 ± 0.010	0.698 ± 0.049	0.730 ± 0.048	0.740 ± 0.044
HandOutlines	0.930 ± 0.007	0.931 ± 0.006	0.936 ± 0.004	**0.942** ± 0.006	0.931 ± 0.004
Haptics	0.491 ± 0.026	0.462 ± 0.002	0.487 ± 0.027	0.521 ± 0.033	**0.523** ± 0.022
Herring	**0.698** ± 0.018	0.682 ± 0.024	0.615 ± 0.018	0.620 ± 0.039	0.635 ± 0.033
HouseTwenty	0.947 ± 0.010	0.961 ± 0.010	**0.980** ± 0.005	0.975 ± 0.008	**0.980** ± 0.005
InlineSkate	0.364 ± 0.019	0.334 ± 0.021	0.393 ± 0.008	**0.403** ± 0.005	0.396 ± 0.008
InsectEPGRegularTrain	0.987 ± 0.014	**1.000** ± 0.000	0.997 ± 0.005	**1.000** ± 0.000	**1.000** ± 0.000
InsectEPGSmallTrain	0.953 ± 0.008	**1.000** ± 0.000	0.985 ± 0.008	0.981 ± 0.014	**1.000** ± 0.000

Continuation of Table 12

Dataset	Moment	Mantis	TiViT	TiViT + Moment	TiViT + Mantis
InsectWingbeatSound	0.539 ± 0.003	0.469 ± 0.019	0.524 ± 0.016	**0.553** ± 0.010	0.531 ± 0.013
ItalyPowerDemand	**0.938** ± 0.005	0.911 ± 0.007	0.928 ± 0.015	0.937 ± 0.013	0.928 ± 0.014
LargeKitchenAppliances	0.859 ± 0.005	0.820 ± 0.010	0.880 ± 0.012	**0.884** ± 0.014	0.874 ± 0.009
Lightning2	0.760 ± 0.041	0.781 ± 0.025	0.820 ± 0.000	**0.836** ± 0.016	**0.836** ± 0.033
Lightning7	0.836 ± 0.036	0.749 ± 0.021	0.836 ± 0.014	**0.868** ± 0.008	0.845 ± 0.008
Mallat	0.915 ± 0.010	0.868 ± 0.028	0.930 ± 0.033	**0.957** ± 0.017	0.939 ± 0.023
Meat	0.911 ± 0.038	**0.939** ± 0.019	0.806 ± 0.019	0.900 ± 0.029	0.872 ± 0.051
MedicalImages	0.731 ± 0.003	0.705 ± 0.024	0.741 ± 0.011	**0.778** ± 0.009	0.762 ± 0.013
MelbournePedestrian	**0.933** ± 0.004	0.908 ± 0.006	0.860 ± 0.005	0.930 ± 0.005	0.920 ± 0.006
MiddlePhalanxOutlineAgeGroup	0.481 ± 0.028	**0.563** ± 0.042	0.552 ± 0.023	0.530 ± 0.023	0.550 ± 0.014
MiddlePhalanxOutlineCorrect	0.813 ± 0.028	**0.844** ± 0.007	0.784 ± 0.019	0.795 ± 0.019	0.818 ± 0.019
MiddlePhalanxTW	0.515 ± 0.019	0.455 ± 0.019	**0.517** ± 0.004	0.498 ± 0.004	0.509 ± 0.014
MixedShapesRegularTrain	0.947 ± 0.002	0.956 ± 0.003	0.975 ± 0.001	0.974 ± 0.001	**0.978** ± 0.001
MixedShapesSmallTrain	0.876 ± 0.011	0.897 ± 0.010	0.944 ± 0.006	0.935 ± 0.006	**0.947** ± 0.009
MoteStrain	0.879 ± 0.011	0.887 ± 0.015	0.899 ± 0.004	**0.922** ± 0.012	0.918 ± 0.013
NonInvasiveFetalECGThorax1	0.918 ± 0.001	0.799 ± 0.004	0.890 ± 0.008	**0.921** ± 0.005	0.887 ± 0.002
NonInvasiveFetalECGThorax2	0.927 ± 0.002	0.817 ± 0.004	0.915 ± 0.003	**0.933** ± 0.002	0.918 ± 0.003
OSULeaf	0.920 ± 0.009	0.902 ± 0.006	**0.988** ± 0.007	0.986 ± 0.005	0.985 ± 0.002
OliveOil	0.889 ± 0.019	**0.944** ± 0.019	0.700 ± 0.033	0.856 ± 0.019	0.789 ± 0.051
PLAID	0.741 ± 0.005	0.819 ± 0.005	0.911 ± 0.005	0.901 ± 0.007	**0.929** ± 0.007
PhalangesOutlinesCorrect	**0.800** ± 0.004	0.796 ± 0.006	0.789 ± 0.005	**0.800** ± 0.012	0.794 ± 0.005
Phoneme	0.276 ± 0.014	0.294 ± 0.013	0.377 ± 0.008	0.377 ± 0.009	**0.386** ± 0.011
PickupGestureWiimoteZ	0.760 ± 0.040	0.807 ± 0.012	0.853 ± 0.042	0.840 ± 0.060	**0.887** ± 0.042
PigAirwayPressure	0.117 ± 0.017	0.579 ± 0.012	0.535 ± 0.011	0.474 ± 0.007	**0.612** ± 0.032
PigArtPressure	0.750 ± 0.019	0.811 ± 0.015	0.798 ± 0.024	0.808 ± 0.021	**0.845** ± 0.024
PigCVP	0.723 ± 0.018	**0.777** ± 0.012	0.670 ± 0.028	0.734 ± 0.012	**0.777** ± 0.007
Plane	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000
PowerCons	0.930 ± 0.012	0.941 ± 0.017	0.898 ± 0.006	**0.952** ± 0.014	0.915 ± 0.003
ProximalPhalanxOutlineAgeGroup	0.800 ± 0.015	**0.850** ± 0.014	0.837 ± 0.007	0.833 ± 0.010	0.837 ± 0.012
ProximalPhalanxOutlineCorrect	0.875 ± 0.010	**0.885** ± 0.005	0.861 ± 0.008	0.877 ± 0.002	0.875 ± 0.005
ProximalPhalanxTW	**0.751** ± 0.013	0.727 ± 0.013	0.740 ± 0.007	0.738 ± 0.010	0.740 ± 0.010
RefrigerationDevices	0.520 ± 0.023	0.517 ± 0.014	**0.568** ± 0.019	0.552 ± 0.023	0.564 ± 0.029
Rock	0.640 ± 0.087	0.607 ± 0.110	0.833 ± 0.099	0.807 ± 0.095	**0.840** ± 0.106
ScreenType	0.477 ± 0.018	0.465 ± 0.013	0.523 ± 0.012	0.542 ± 0.019	**0.548** ± 0.006
SemgHandGenderCh2	0.742 ± 0.010	0.877 ± 0.010	0.877 ± 0.008	0.866 ± 0.013	**0.916** ± 0.010
SemgHandMovementCh2	0.414 ± 0.019	0.657 ± 0.012	0.547 ± 0.005	0.533 ± 0.007	**0.692** ± 0.009
SemgHandSubjectCh2	0.662 ± 0.002	0.834 ± 0.013	0.840 ± 0.002	0.819 ± 0.006	**0.884** ± 0.008
ShakeGestureWiimoteZ	0.907 ± 0.031	0.907 ± 0.012	0.840 ± 0.035	**0.913** ± 0.012	0.867 ± 0.012
ShapeletSim	0.963 ± 0.006	0.924 ± 0.008	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000
ShapesAll	0.893 ± 0.008	0.851 ± 0.007	0.899 ± 0.003	**0.915** ± 0.002	0.909 ± 0.002
SmallKitchenAppliances	0.720 ± 0.012	0.784 ± 0.012	**0.815** ± 0.015	**0.815** ± 0.019	0.808 ± 0.017
SmoothSubspace	0.891 ± 0.020	**0.976** ± 0.004	**0.976** ± 0.014	0.967 ± 0.007	**0.976** ± 0.010
SonyAIBORobotSurface1	0.829 ± 0.015	**0.881** ± 0.027	0.845 ± 0.021	0.840 ± 0.020	0.854 ± 0.019
SonyAIBORobotSurface2	0.829 ± 0.032	0.876 ± 0.032	0.901 ± 0.028	0.904 ± 0.044	**0.910** ± 0.024
StarLightCurves	0.969 ± 0.001	0.969 ± 0.000	0.974 ± 0.001	**0.976** ± 0.001	**0.976** ± 0.001
Strawberry	**0.972** ± 0.002	0.959 ± 0.003	0.958 ± 0.002	0.968 ± 0.010	0.964 ± 0.004
SwedishLeaf	0.919 ± 0.011	0.939 ± 0.004	0.953 ± 0.001	**0.960** ± 0.002	0.958 ± 0.001
Symbols	0.965 ± 0.006	0.984 ± 0.002	**0.987** ± 0.000	0.986 ± 0.000	0.986 ± 0.001
SyntheticControl	0.967 ± 0.006	0.989 ± 0.004	0.999 ± 0.002	0.996 ± 0.004	**1.000** ± 0.000
ToeSegmentation1	0.953 ± 0.022	**0.968** ± 0.013	0.923 ± 0.009	0.950 ± 0.015	0.952 ± 0.008
ToeSegmentation2	0.897 ± 0.016	**0.962** ± 0.008	0.913 ± 0.016	0.913 ± 0.009	0.923 ± 0.008
Trace	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000
TwoLeadECG	0.916 ± 0.020	0.997 ± 0.001	**1.000** ± 0.000	0.999 ± 0.001	**1.000** ± 0.000
TwoPatterns	0.989 ± 0.001	0.949 ± 0.003	**0.998** ± 0.001	**0.998** ± 0.000	0.997 ± 0.001
UMD	**0.993** ± 0.000	0.988 ± 0.008	**0.993** ± 0.000	**0.993** ± 0.000	**0.993** ± 0.000
UWaveGestureLibraryAll	0.924 ± 0.001	0.872 ± 0.004	0.937 ± 0.002	**0.948** ± 0.003	0.944 ± 0.001
UWaveGestureLibraryX	0.793 ± 0.003	0.778 ± 0.009	0.825 ± 0.002	0.836 ± 0.005	**0.838** ± 0.003
UWaveGestureLibraryY	0.708 ± 0.010	0.677 ± 0.009	0.755 ± 0.002	**0.765** ± 0.002	0.764 ± 0.002
UWaveGestureLibraryZ	0.729 ± 0.005	0.737 ± 0.005	0.761 ± 0.006	0.773 ± 0.010	**0.788** ± 0.005
Wafer	0.992 ± 0.002	0.996 ± 0.000	**0.999** ± 0.000	**0.999** ± 0.000	**0.999** ± 0.000
Wine	**0.901** ± 0.028	0.833 ± 0.037	0.673 ± 0.057	0.759 ± 0.037	0.759 ± 0.032
WordSynonyms	0.644 ± 0.017	0.623 ± 0.016	0.643 ± 0.017	**0.677** ± 0.020	0.675 ± 0.028
Worms	0.749 ± 0.033	0.697 ± 0.037	0.753 ± 0.047	**0.805** ± 0.022	0.784 ± 0.067
WormsTwoClass	0.775 ± 0.037	0.740 ± 0.000	0.775 ± 0.033	0.784 ± 0.040	**0.805** ± 0.022
Yoga	0.833 ± 0.008	0.771 ± 0.014	0.819 ± 0.005	**0.841** ± 0.006	0.838 ± 0.006

End of Table

Table 13: Classification accuracy for 27 multivariate datasets from the UEA benchmark. We report the mean and standard deviation across three random seeds.

Dataset	Moment	Mantis	TiViT	TiViT + Moment	TiViT + Mantis
ArticularyWordRecognition	0.988 ± 0.002	**0.991** ± 0.002	0.977 ± 0.003	0.977 ± 0.003	0.974 ± 0.005
BasicMotions	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000
CharacterTrajectories	**0.982** ± 0.001	0.973 ± 0.001	0.964 ± 0.005	**0.982** ± 0.001	0.978 ± 0.005
Cricket	**1.000** ± 0.000	0.986 ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000
DuckDuckGeese	**0.467** ± 0.081	0.433 ± 0.023	0.393 ± 0.081	0.413 ± 0.064	0.433 ± 0.050
ERing	0.895 ± 0.022	0.905 ± 0.025	0.975 ± 0.014	0.977 ± 0.006	**0.981** ± 0.007
EigenWorms	0.746 ± 0.022	0.746 ± 0.016	**0.911** ± 0.016	0.880 ± 0.009	**0.911** ± 0.012
Epilepsy	**1.000** ± 0.000	0.990 ± 0.004	**1.000** ± 0.000	**1.000** ± 0.000	**1.000** ± 0.000
EthanolConcentration	0.445 ± 0.013	0.269 ± 0.044	**0.485** ± 0.012	0.473 ± 0.030	0.465 ± 0.019
FaceDetection	0.584 ± 0.007	0.592 ± 0.006	0.598 ± 0.004	0.584 ± 0.007	**0.607** ± 0.005
FingerMovements	**0.633** ± 0.045	0.593 ± 0.025	0.517 ± 0.040	0.620 ± 0.036	0.553 ± 0.050
HandMovementDirection	**0.279** ± 0.051	0.212 ± 0.021	0.275 ± 0.016	0.257 ± 0.036	0.257 ± 0.027
Handwriting	0.296 ± 0.018	**0.425** ± 0.013	0.307 ± 0.034	0.340 ± 0.002	0.385 ± 0.021
Heartbeat	0.735 ± 0.007	**0.800** ± 0.017	0.732 ± 0.008	0.717 ± 0.022	0.769 ± 0.003
InsectWingbeat	0.231 ± 0.012	**0.573** ± 0.017	0.355 ± 0.008	0.332 ± 0.018	0.443 ± 0.020
JapaneseVowels	0.918 ± 0.006	**0.978** ± 0.003	0.940 ± 0.002	0.938 ± 0.012	0.933 ± 0.008
LSST	0.571 ± 0.005	0.607 ± 0.009	0.604 ± 0.005	0.610 ± 0.009	**0.652** ± 0.003
Libras	0.861 ± 0.017	0.887 ± 0.026	0.907 ± 0.006	**0.922** ± 0.022	0.920 ± 0.018
MotorImagery	0.530 ± 0.026	**0.563** ± 0.012	**0.563** ± 0.049	0.560 ± 0.044	0.553 ± 0.042
NATOPS	0.900 ± 0.029	**0.931** ± 0.014	0.869 ± 0.006	0.889 ± 0.006	0.878 ± 0.006
PEMS-SF	0.705 ± 0.029	**0.788** ± 0.029	0.709 ± 0.084	0.763 ± 0.044	0.742 ± 0.087
PhonemeSpectra	0.186 ± 0.004	0.272 ± 0.006	0.245 ± 0.007	0.265 ± 0.007	**0.286** ± 0.008
RacketSports	0.829 ± 0.007	**0.919** ± 0.004	0.846 ± 0.010	0.871 ± 0.008	0.879 ± 0.027
SelfRegulationSCP1	0.762 ± 0.010	0.825 ± 0.022	0.858 ± 0.008	0.840 ± 0.003	**0.891** ± 0.010
SelfRegulationSCP2	0.509 ± 0.031	0.491 ± 0.018	**0.526** ± 0.038	0.506 ± 0.017	0.517 ± 0.020
SpokenArabicDigits	**0.981** ± 0.003	0.907 ± 0.006	0.969 ± 0.001	0.979 ± 0.003	0.972 ± 0.002
UWaveGestureLibrary	0.846 ± 0.010	0.879 ± 0.015	0.910 ± 0.005	0.902 ± 0.004	**0.919** ± 0.009

Table 14: Mean rank of TiViT and TSFMs across datasets from the UCR and UEA archive.

Model	UCR	UEA
Moment	3.66	3.33
Mantis	3.44	2.85
TiViT *(Ours)*	2.97	2.85
TiViT + Moment *(Ours)*	2.16	2.63
TiViT + Mantis *(Ours)*	**1.92**	**2.22**

G Broader impacts

Since this paper presents foundational machine learning research, we do not see any direct societal risks. The broader impact of our work will depend on its specific application.

We demonstrate that our method TiViT significantly improves classification accuracy. This advancement can be beneficial in healthcare where the analysis of physiological signals is crucial for early diagnosis and treatment or in industry where the accurate monitoring of sensor data enables predictive maintenance and reduces downtime.

However, deep learning models including TiViT operate as black boxes with limited interpretability. In safety-critical domains or applications directly impacting humans, such models necessitate careful deployment and oversight. Further research into interpretability and human-in-the-loop frameworks is essential to make deep learning models trustworthy for real-world settings.