LEARNING REPRESENTATIONS FOR MULTIVARIATE TIME SERIES CLASSIFICATION VIA CONTEXT PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised representation learning for multivariate time series classification is dominated by contrastive learning and masked modeling. However, these paradigms incur substantial computational overhead and often rely on complex data augmentations or reconstruction decoders that introduce domain-specific biases. In this work, we propose **C-Pred**, a lightweight self-supervised framework based on next-patch context prediction. C-Pred formulates self-supervised learning as a K-way classification problem: given a reference patch, the model identifies its immediate temporal neighbor from K candidate patches within the same instance. This lightweight formulation, requiring only patch-level encoding rather than full-sequence processing, makes **channel-independent encoding** practical: each channel is processed independently by a shared univariate backbone, allowing a single model to accommodate arbitrary channel dimensions without increasing the backbone size. By requiring only a small number of temporal patches and no data augmentation, cross-instance negative sampling, or signal reconstruction, C-Pred substantially reduces the computational cost of self-supervision. Across six large-scale benchmarks (>20k instances) and 28 UEA multivariate datasets, C-Pred matches or exceeds state-of-the-art representation quality while achieving up to a **7.5× speedup** in training time over competitive baselines. On the large-scale benchmarks, channel-independent encoding further improves accuracy by 2.99 percentage points over channel mixing, achieving 84.36% average accuracy.

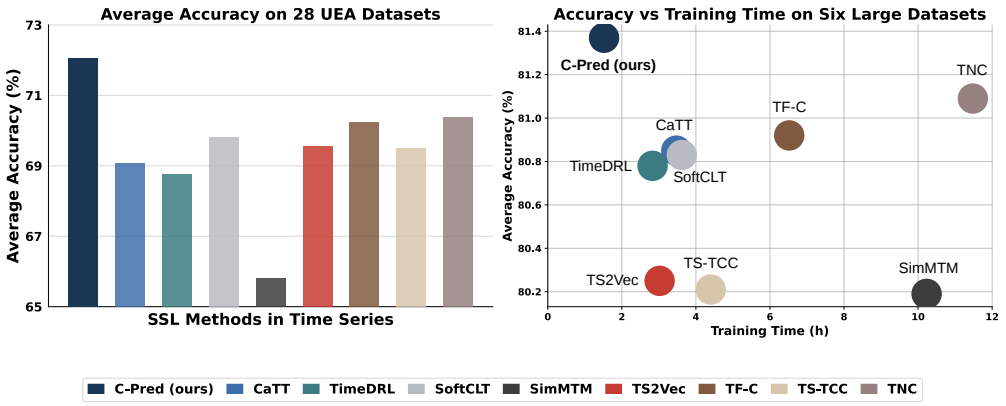

Figure 1: Training efficiency and downstream performance of SSL methods. *Left:* Average accuracy across 28 UEA benchmark datasets. *Right:* Average accuracy vs total training time across six large-scale datasets (>20k samples).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Two families of Self-supervised learning (SSL) methods have proven particularly effective for time series classification. *Contrastive learning* methods (Eldele et al., 2021; Yue et al., 2022; Tonekaboni et al., 2021; Zhang et al., 2022; Lee et al., 2024) learn representations by pulling together embeddings of semantically similar views (positive pairs) while pushing apart dissimilar ones (negative pairs). *Masked modelling* methods (Zerveas et al., 2021; Dong et al., 2023; Nie et al., 2023) mask portions of the input and train the encoder to reconstruct the missing content, encouraging the model to capture the underlying temporal structure. Both paradigms have demonstrated strong empirical performance and have advanced the state of the art on several time series classification benchmarks.

However, these approaches have notable limitations. Contrastive learning depends heavily on the construction of positive and negative pairs. Augmentation-based methods may distort the semantic structure of the signal, while temporal adjacency methods can fail on periodic signals where distant segments are more similar than neighbouring ones, leading to false negatives and noisy supervision. Masked modelling avoids augmentations but introduces other challenges, including sensitive masking strategies, higher computational cost, and reconstruction objectives that may overemphasise low-level signal details instead of meaningful semantic representations.

In this work, we propose *C-Pred*, a simple and efficient SSL framework for multivariate time series classification based on a context prediction pretext task. Rather than constructing positive and negative pairs or learning to reconstruct masked inputs, C-Pred formulates the SSL objective as a *K-way classification task*: given a time series instance partitioned into P non-overlapping patches of length L, a reference patch i is selected, and the encoder must identify which of K candidate patches drawn from the same instance is the immediate temporal neighbour of the reference. This admits a simple and tractable pretext task that has exactly one correct answer per instance.

This formulation offers three advantages over existing approaches. **First, no positive pair selection is required.** C-Pred avoids both augmentation-based and adjacency-based contrastive learning by defining the task directly from the temporal structure of each instance, eliminating assumptions about semantic-preserving transformations and reducing false negatives in periodic signals. **Second, no masking or reconstruction is needed.** The task is formulated as a simple K-way classification problem, avoiding decoders, reconstruction losses, and masking strategies. **Third, C-Pred is computationally efficient.** Instead of processing the full sequence or multiple augmented views, the encoder operates only on a reference patch and K candidate patches, substantially reducing computation, particularly in the binary setting ($K = 2$).

We evaluate C-Pred on six large-scale real-world benchmarks with more than 20k instances each, and on 28 UEA multivariate archive spanning diverse temporal domains. C-Pred achieves competitive performance across all settings while maintaining substantially reduced training time compared to contrastive and masked modelling baselines.

Our main contributions are summarised as follows:

- We propose C-Pred, a pretext task for self-supervised multivariate time series representation learning based on next-patch identification via a K-way classification objective, requiring no positive pair selection, no cross-instance negative sampling, and no masking or reconstruction.
- We provide empirical analysis of the effect of channel independence, channel mixing, encoder architecture, batch size, and prediction head, on the speed and downstream classification performance on multivariate time series.
- We evaluate C-Pred on six large-scale real-world benchmarks and 28 UEA multivariate datasets on multivariate time series classification, demonstrating strong and consistent performance across diverse temporal dynamics with substantially reduced training time compared to existing SSL methods.

2 RELATED WORK

Self-Supervised Learning in Time Series. Motivated by advances of SSL in computer vision (Doersch et al., 2015; Noroozi & Favaro, 2016; Chen et al., 2020; He et al., 2020; Caron et al., 2021;

He et al., 2022), a growing body of work has adapted SSL to the time series domain, primarily through contrastive learning and masked modelling. TNC (Tonekaboni et al., 2021) exploits temporal neighbourhood structure by treating signals from nearby time windows as positives and distant windows as negatives. TS-TCC (Eldele et al., 2021) combines temporal and contextual contrasting with weak and strong augmentations to learn representations robust to temporal perturbations. TF-C (Zhang et al., 2022) aligns representations in both the time and frequency domains through cross-domain contrastive learning. TS2Vec (Yue et al., 2022) introduces a hierarchical contrastive framework operating at both the instance and temporal level using randomly masked overlapping windows, producing dense timestep-level representations. SoftCLT (Lee et al., 2024) relaxes the hard positive/negative assignment by introducing soft similarity labels based on temporal distance. TimeDRL (Chang et al., 2024) proposes a disentangled representation learning framework that jointly optimizes timestamp-level predictive objectives and instance-level contrastive objectives. CaTT (Shamba et al., 2025) eliminates augmentation and multiple encoder passes by contrasting temporally adjacent representations within a sequence, treating nearby timestamps as positives. On the masked modelling side, SimMTM (Dong et al., 2023) frames SSL as a masked reconstruction task over manifold-aware patch aggregations, recovering masked time points from multiple masked series. While these methods achieve strong empirical performance, they share a common dependency on either carefully designed augmentation strategies, which embed domain-specific assumptions and risk distorting the signal, or complex masking and reconstruction objectives that introduce computational overhead. Our work takes a different approach, formulating the SSL objective as a simple K-way classification pretext task that requires neither augmentation nor reconstruction.

Proxy Objectives via Temporal Context Prediction. In contrast to the dominance of contrastive and reconstruction-based objectives, early proxy-based pretext tasks (Doersch et al., 2015; Noroozi & Favaro, 2016; Pathak et al., 2016; Gidaris et al., 2018), have received limited attention in the time series domain. This is largely because time series SSL emerged after the shift in computer vision toward contrastive learning and masked modeling, leading to a direct adoption of these paradigms without revisiting earlier alternatives. The question of whether these simpler pretext tasks offer advantages in the time series setting remains largely unexplored.

Our work is inspired by the context prediction pretext task of Doersch et al. (2015), which uses spatial relationships between image patches as a supervisory signal for representation learning. However, directly adapting this idea to time series is ineffective, since temporal patches from the same instance are highly similar, making fine-grained position prediction ambiguous and weakly supervised (Fig. 2).

We instead reformulate the task: given a reference patch and K candidate patches, the model predicts which candidate is the immediate temporal neighbour. This converts an ill-posed regression problem into a simple K-way classification task with a unique correct answer. To our knowledge, C-Pred is the first to systematically adapt context prediction to multivariate time series representation learning in this form.

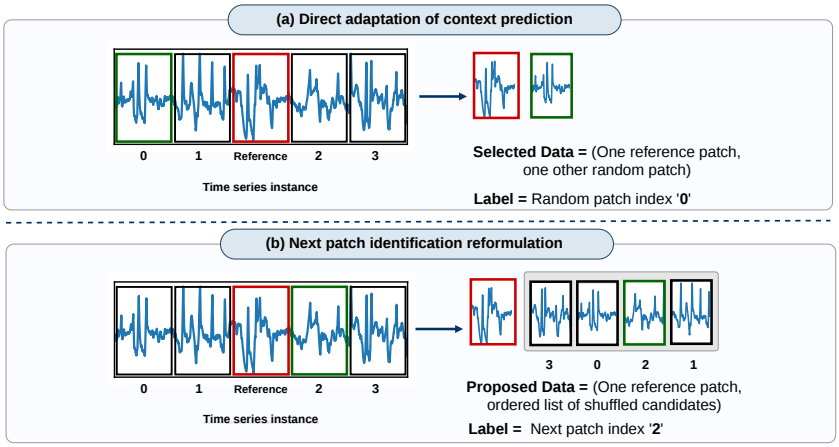

Figure 2: (a) Direct adaptation of image-based context prediction to time series. (b) C-Pred reformulation as next-patch identification via K-way classification.

3 C-PRED METHOD

3.1 PROBLEM DEFINITION

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$ denote a collection of N unlabelled time series instances, where each $x_i \in \mathbb{R}^{T \times C}$, with T denoting the sequence length and C the number of channels. Our objective is to learn an encoder $f_\theta : \mathbb{R}^{L \times C} \to \mathbb{R}^d$ that maps a patch of length $L < T$ to a d-dimensional representation, such that the learned embeddings are discriminative and transferable to downstream classification tasks without any labelled supervision. Further details on the channel-independent aggregation mechanism and variable-length inference are provided in Appendix B.

We adopt a pretext task framework in which the supervisory signal is derived entirely from the temporal structure of individual instances, without requiring cross-instance negative sampling, data augmentation, or signal reconstruction. The key design choice, and the central focus of this work, is the formulation of temporal context prediction as a tractable K-way classification objective.

3.2 WHY PRETEXT TASKS FOR TIME SERIES

The design of the self-supervised objective is the most consequential decision in SSL, as it determines what structure the encoder is required to learn. For the learned representation to support downstream classification, the pretext task must satisfy three competing constraints.

Label Preservation. The pretext task must not require the model to learn invariances that destroy class-relevant information. Augmentation-based contrastive methods risk this when transformations alter the temporal dynamics that distinguish classes.

Tractability. The pretext task must be neither too easy nor too hard. A task that is too easy, such as predicting whether two identical patches are the same, provides no useful learning signal. A task that is too hard, such as predicting the precise temporal position of a patch among all P possible positions in a long sequence, provides a signal too sparse for the encoder to learn from efficiently.

Signal Density. The pretext task should provide dense supervisory signal per training step. Contrastive methods with a single positive pair per instance provide $2B$ positive assignments per batch update. Masked modeling provides a dense signal at the token level but at the cost of a decoder and reconstruction loss. A well-designed pretext task should provide comparable signal density with lower architectural and computational overhead.

These constraints together motivate our reformulation of context prediction for time series.

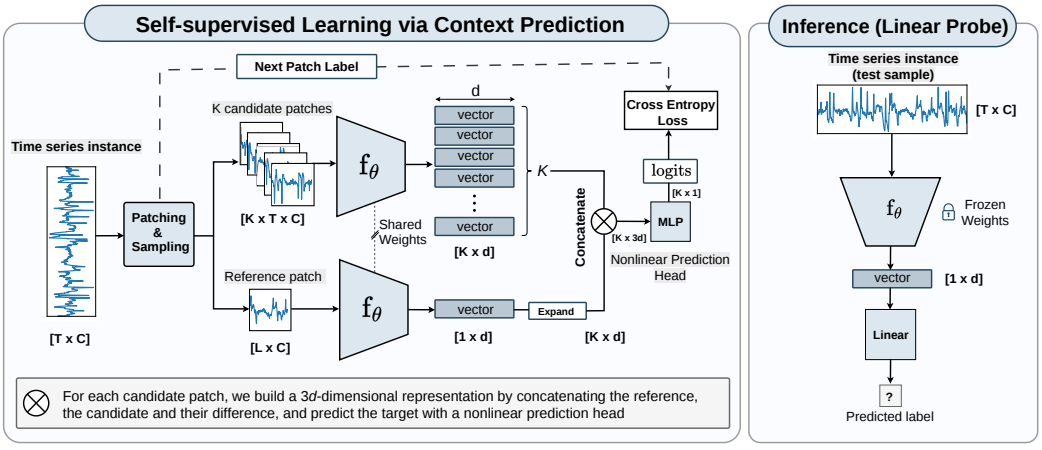

Figure 3: Overview of the proposed C-Pred framework. Given a reference patch, the model learns representations by identifying the temporally adjacent patch from a set of candidate patches via a K-way cross-entropy objective.

3.3 Patch Construction and Context Prediction

Given a time series instance $x \in \mathbb{R}^{T \times C}$, we partition it into P non-overlapping patches of equal length:

$$L = \left\lfloor \frac{T}{P} \right\rfloor, \qquad x^{(p)} = x_{(p-1)L\,:\,pL} \in \mathbb{R}^{L \times C}, \quad p = 1, \ldots, P. \tag{1}$$

From the P patches, we select a reference patch $x^{(r)}$ uniformly at random from $\{x^{(1)}, \ldots, x^{(P-1)}\}$, excluding the last patch to ensure a valid right neighbour always exists. The true neighbour is defined as the immediately next patch: $x^{(r+1)} \in \mathbb{R}^{L \times C}$.

We then sample $K-1$ distractor patches uniformly at random from the remaining $P-2$ patches $\{x^{(p)} : p \neq r,\, p \neq r+1\}$, forming a candidate pool of K patches:

$$\mathcal{C} = \{x^{(r+1)}\} \cup \{x^{(d_1)}, \ldots, x^{(d_{K-1})}\}, \tag{2}$$

where $d_1, \ldots, d_{K-1}$ are the distractor indices. The candidates are randomly shuffled, and the label $y^* \in \{0, \ldots, K-1\}$ records the position of the true neighbour in the shuffled pool. This shuffling is essential to prevent the model from exploiting position bias in the candidate pool.

3.4 Encoder and Prediction Head

Each patch is encoded independently using a shared encoder f_θ:

$$z^{(r)} = f_\theta(x^{(r)}) \in \mathbb{R}^d, \qquad z^{(c_k)} = f_\theta(x^{(c_k)}) \in \mathbb{R}^d, \quad k = 0, \ldots, K-1, \tag{3}$$

where $x^{(r)}$ denotes the reference patch and $\{x^{(c_k)}\}_{k=0}^{K-1}$ are the K candidate patches in the shuffled pool.

To model interactions between the reference and candidates, we first expand the reference representation and construct pairwise feature vectors: $\tilde{z}^{(r)} = \text{repeat}(z^{(r)}, K) \in \mathbb{R}^{K \times d}$,

$$h_k = \left[\tilde{z}_k^{(r)} \parallel z^{(c_k)} \parallel \left(\tilde{z}_k^{(r)} - z^{(c_k)} \right) \right] \in \mathbb{R}^{3d}, \tag{4}$$

where $\parallel$ denotes concatenation.

The resulting matrix $H = [h_0, \ldots, h_{K-1}] \in \mathbb{R}^{K \times 3d}$ is passed through a nonlinear prediction head g_ϕ to produce logits:

$$s_k = g_\phi(h_k), \quad k = 0, \ldots, K-1, \tag{5}$$

yielding $\mathbf{s} = [s_0, \ldots, s_{K-1}] \in \mathbb{R}^K$.

These logits define a categorical distribution over the candidate pool, enabling direct K-way classification without relying on predefined similarity functions (Figure 3).

3.5 Training Objective

The pretext task is formulated as a K-way classification problem. Given a reference patch and a shuffled pool of K candidate patches, the model is trained to identify the true neighbour by assigning the highest logit to the correct candidate. We optimise the standard cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{B} \sum_{b=1}^{B} \log \frac{\exp(s_{y_b^*}^{(b)})}{\sum_{k=0}^{K-1} \exp(s_k^{(b)})}, \tag{6}$$

where B is the batch size, $s_k^{(b)}$ denotes the logit for the k-th candidate in instance b, and y_b^* is the index of the true neighbour.

This objective requires no data augmentation, cross-instance negative sampling, or reconstruction decoder. The encoder is trained purely through a stable classification signal derived from the temporal structure within each instance.

Difficulty Control via K. The number of candidates K directly controls the difficulty of the pretext task. At $K = 2$, the model must distinguish the true neighbour from a single distractor which renders an easy binary identification task that provides a weak but stable learning signal. As K increases, the task becomes progressively harder, requiring finer-grained temporal discrimination. We treat K as a hyperparameter and study its effect in our ablation analysis. Importantly, K also controls computational cost: since the encoder processes $K + 1$ patches per instance (one reference and K candidates), larger K increases compute per update but provides a richer training signal. At $K = 2$, the encoder processes only $3L$ time steps per update, compared to T for full-sequence methods, a substantial efficiency gain for long sequences where $L \ll T$.

Inference. The prediction head is discarded at evaluation time. The encoder f_θ processes the full input sequence of length T to produce a single representation vector $z \in \mathbb{R}^d$, used directly for downstream tasks via linear probing. The encoder therefore benefits from the full temporal context at inference, having been trained on diverse patch-level temporal relationships.

4 Experiments

4.1 Evaluation Protocol

Datasets. We evaluate C-Pred on six publicly available large-scale real-world time series datasets: PAMAP2 (Reiss & Stricker, 2012), WISDM2 (Weiss & Lockhart, 2012), HARTH (Logacjov et al., 2021), SLEEP (Goldberger et al., 2000), ECG (Moody & Mark, 1983), and SKODA (Zappi et al., 2012), spanning heterogeneous temporal dynamics from $T = 40$ to $T = 3000$, and input channels ranging from $C = 1$ to $C = 64$. To further assess generalisation across diverse temporal domains and dataset scales, we additionally evaluate on 28 UEA (Bagnall et al., 2018) multivariate archive. Full dataset statistics are provided in the Appendix.

Evaluation Protocols. To evaluate the quality of learned representations, we follow the widely adopted linear evaluation protocol (Chen et al., 2020; Yue et al., 2022; Shamba et al., 2025), where a linear classifier is trained on top of the frozen encoder and test accuracy serves as a proxy for representation quality.

Baselines. We compare C-Pred against the following time series self-supervised learning methods: TNC (Tonekaboni et al., 2021) , TS-TCC (Eldele et al., 2021), TF-C (Zhang et al., 2022), TS2Vec (Yue et al., 2022), SimMTM (Dong et al., 2023), SoftCLT (Lee et al., 2024), TimeDRL (Chang et al., 2024), and CaTT (Shamba et al., 2025). All baselines are reproduced using their official implementations under a unified evaluation protocol to ensure fair comparison.

Default Settings. All methods share the same InceptionTime backbone (Ismail Fawaz et al., 2020) and a unified data-loading pipeline with aligned hyperparameters wherever applicable. For C-Pred, the encoder maps each patch of length L to a d-dimensional representation with $d = 256$. We use $P = 10$ patches, $K = 2$ candidates for the large-scale experiments and $K = 5$ for the UEA benchmark. The prediction head is a nonlinear 2-layer MLP and is discarded at evaluation time. For the six large-scale datasets, models are trained for 1500 iterations with batch size 128. Due to memory constraints from long sequence lengths, we set the batch size to 16 for the SLEEP and ECG datasets. For the UEA benchmark, we follow the training protocol of Yue et al. (2022). All experiments are conducted on NVIDIA V100 GPUs.

4.2 Comparison with State-of-the-Art Time Series SSL methods

We assess the quality of learned representations by freezing the pretrained encoder and training a logistic regression classifier on the extracted embeddings. This linear evaluation protocol provides a direct measure of representation quality, independent of any fine-tuning. Table 1 reports results across all six large-scale datasets.

Large-Scale Benchmarks. Table 1 reports results across all six large-scale datasets. C-Pred achieves the highest average accuracy of $\mathbf{81.37}\%$, while requiring only 1.52 hours of training, the lowest among all methods and $7.5\times$ faster than the second best method TNC (11.48 h). On SKODA and PAMAP2, Random Initialisation achieves the highest accuracy suggesting that these datasets are

Table 1: Classification performance across six datasets. Best results among the standard methods are in **bold**, second-best in *italic*. CI denotes the channel-independent variant.

Model	ECG	HARTH	PAMAP2	SKODA	SLEEP	WISDM2	Time (h) ↓	Avg Acc
w/o Pretrain	70.41$_{\pm 6.09}$	89.78$_{\pm 1.32}$	**71.73**$_{\pm 1.75}$	**99.11**$_{\pm 0.12}$	82.22$_{\pm 0.66}$	59.97$_{\pm 1.42}$	0.00	78.87
w/o Pretrain (CI)	68.18$_{\pm 4.38}$	86.85$_{\pm 4.81}$	72.99$_{\pm 0.59}$	99.80$_{\pm 0.05}$	82.86$_{\pm 0.45}$	65.68$_{\pm 1.19}$	0.00	79.39
TNC	77.93$_{\pm 5.04}$	91.70$_{\pm 1.81}$	*71.59*$_{\pm 2.90}$	99.00$_{\pm 0.12}$	85.14$_{\pm 0.30}$	61.16$_{\pm 1.87}$	11.48	*81.09*
TS-TCC	76.66$_{\pm 2.02}$	89.29$_{\pm 2.17}$	70.42$_{\pm 1.40}$	98.01$_{\pm 0.29}$	85.15$_{\pm 0.34}$	61.70$_{\pm 2.58}$	4.40	80.21
TF-C	76.02$_{\pm 2.81}$	92.24$_{\pm 1.10}$	71.30$_{\pm 1.77}$	98.23$_{\pm 0.21}$	85.18$_{\pm 0.29}$	62.54$_{\pm 3.99}$	6.52	80.92
TS2Vec	76.35$_{\pm 2.07}$	90.49$_{\pm 2.67}$	70.74$_{\pm 0.80}$	98.93$_{\pm 0.16}$	85.05$_{\pm 0.17}$	59.95$_{\pm 2.20}$	3.02	80.25
SimMTM	76.97$_{\pm 2.44}$	89.99$_{\pm 2.00}$	70.15$_{\pm 0.50}$	98.19$_{\pm 0.10}$	**85.22**$_{\pm 0.21}$	60.64$_{\pm 3.66}$	10.23	80.19
SoftCLT	76.90$_{\pm 4.59}$	91.58$_{\pm 0.91}$	70.67$_{\pm 0.39}$	99.02$_{\pm 0.19}$	85.05$_{\pm 0.16}$	61.73$_{\pm 3.17}$	3.62	80.83
TimeDRL	*78.39*$_{\pm 5.12}$	87.54$_{\pm 5.25}$	70.61$_{\pm 1.10}$	98.69$_{\pm 0.14}$	84.22$_{\pm 0.57}$	**65.20**$_{\pm 2.05}$	*2.83*	80.78
CaTT	**78.81**$_{\pm 9.30}$	*93.13*$_{\pm 0.53}$	69.86$_{\pm 0.33}$	94.87$_{\pm 0.69}$	*85.17*$_{\pm 0.33}$	63.25$_{\pm 3.36}$	3.47	80.85
C-Pred	76.15$_{\pm 2.73}$	**93.43**$_{\pm 1.37}$	70.54$_{\pm 0.55}$	*99.08*$_{\pm 0.12}$	85.15$_{\pm 0.17}$	*63.87*$_{\pm 4.21}$	**1.52**	**81.37**
C-Pred (CI)	79.88$_{\pm 2.63}$	94.45$_{\pm 0.78}$	80.95$_{\pm 0.50}$	99.62$_{\pm 0.06}$	85.16$_{\pm 0.31}$	66.09$_{\pm 1.69}$	10.76	84.36
TNC (CI)	72.36$_{\pm 6.31}$	OOM	OOM	OOM	85.14$_{\pm 0.25}$	65.95$_{\pm 1.04}$	-	-

Note: OOM denotes out-of-memory during training. TNC with channel-independent (CI) encoding exceeded the available GPU memory on HARTH, PAMAP2, and SKODA; results are therefore not reported for these datasets.

sufficiently structured that even unlearned features provide strong signal, leaving limited room for SSL methods to improve.

A striking finding emerges from an additional ablation reported in the same table: applying C-Pred under *channel independence* (CI), where each channel is encoded separately by the same univariate encoder and representations are aggregated, yields an average accuracy of $\mathbf{84.36}\%$, a gain of 2.99 percentage points (pp) over channel mixing and surpassing all baselines by a substantial margin across every dataset. Although, the training time under channel independence increases to 10.76 hours, reflecting the cost of processing each channel independently, nevertheless this remains comparable to TNC (11.48 h) while achieving 3.27 percentage points higher average accuracy.

Table 2: Summary of classification results on the 28 UEA multivariate time series datasets. Best results in **bold**, second-best in *italic*. CI denotes the channel-independent variant.

Method	Avg. Acc. ↑	Avg. Rank ↓	Ranks 1st ↑	Avg. Diff. (%) ↓	W/T/L	Wilcoxon p-value ↓	Time (h) ↓
w/o Pretrain	67.28	7.41	0	4.78	17/6/5	0.00	0.00
w/o Pretrain (CI)	66.04	7.86	1	6.02	19/4/5	0.00	0.00
TNC	70.38	5.91	*3*	1.68	13/3/12	0.15	0.61
TS-TCC	69.49	6.89	0	2.56	16/4/8	0.00	0.47
TF-C	*70.23*	*5.71*	1	*1.83*	16/2/10	*0.03*	0.53
TS2Vec	68.55	6.71	1	3.50	18/3/7	0.00	0.37
SimMTM	65.82	8.46	1	6.23	21/1/6	0.00	1.24
SoftCLT	69.61	6.25	1	2.44	15/6/7	0.01	0.25
TimeDRL	68.76	6.36	2	3.29	15/3/10	0.02	0.32
CaTT	69.07	6.68	2	2.99	15/2/11	0.05	**0.18**
C-Pred	**72.05**	**5.05**	**4**	–	–	–	*0.19*
C-Pred (CI)	70.67	4.70	8	1.39	10/5/13	0.46	0.63

Note: We also evaluated a channel-independent (CI) variant of TNC, the second-best standard method. However, the CI implementation resulted in out-of-memory (OOM) errors on several UEA datasets, preventing evaluation across the full benchmark.

UEA Multivariate Archive. Table 2 reports summary statistics across 28 UEA datasets. C-Pred under channel mixing achieves the best average accuracy ($\mathbf{72.05}\%$) and the most first-place wins ($\mathbf{4}$), confirming strong generalisation across diverse multivariate temporal domains. Channel independence variant of C-Pred achieves 70.67% average accuracy and a better average rank ($\mathbf{4.70}$ vs 5.05) with eight first-place wins, more than any other method, but a lower overall average due to underperformance on a subset of small datasets, such as `StandWalkJump` and `UWaveGestureLibrary`, where cross-variable dependencies are informative. Wilcoxon signed-rank tests confirm that C-Pred significantly outperforms all methods except TNC ($p = 0.15$), which is the only baseline whose performance is statistically indistinguishable from C-Pred, while requiring a fraction of the training time of TNC. Specifically, TNC trains for 0.61 hours on UEA, more than $3\times$ longer than C-Pred (0.19 h), and achieves a higher average rank (5.91 vs 5.05), a worse W/T/L record, and a higher average accuracy gap. These results collectively demonstrate that C-Pred learns representations that generalise consistently across diverse multivariate temporal domains and achieves the best accuracy-efficiency trade-off among all methods evaluated.

4.3 COMPUTATIONAL EFFICIENCY

A key design property of C-Pred is its computational efficiency, which arises directly from the patch-based pretext task formulation. Figure 5, Table 1 and Table 2 report total training time across the six large-scale datasets and the UEA benchmark. C-Pred achieves the lowest total training time overall ($1.52 + 0.19$ h). This efficiency gain comes from the design of the method. At each training step, C-Pred's encoder processes only $(K + 1)$ patches of length $L = T/P$, a total of $(K + 1)T/P$ time steps, compared to T or $2T$ for single-view and dual-view full-sequence methods. For the large scale datasets, at the default settings of $P = 10$ and $K = 2$, the encoder sees only $3T/10 = 0.3T$ time steps per update, a 70% reduction in encoder compute relative to full-sequence methods. This reduction scales proportionally with sequence length, making C-Pred particularly efficient for long sequences such as ECG ($T = 2500$) and SLEEP ($T = 3000$).

5 ANALYSIS

Table 3 reports a comprehensive ablation study of C-Pred across six design dimensions on the large-scale benchmarks and the 28 UEA datasets. Default settings are highlighted in gray.

Pretext Task Formulation. We compare our predictive K-way cross-entropy objective against a contrastive alternative that uses the same patch pairs but optimises NT-Xent instead. The contrastive variant achieves higher average accuracy on the large-scale benchmarks but lower accuracy on UEA, suggesting that the contrastive objective overfits to the characteristics of large datasets while the predictive objective generalises more consistently across diverse dataset scales. We adopt the predictive formulation as the default for its more stable cross-entropy objective and superior UEA performance.

Table 3: Ablation study of C-Pred across different settings on large-scale datasets and the 28 UEA benchmark.

Method	ECG	HARTH	PAMAP2	SKODA	WISDM2	28 UEA Datasets	Total Time (h) ↓
Classification Logits							
Contrastive	$81.07_{\pm 6.19}$	$93.47_{\pm 0.87}$	$70.70_{\pm 0.29}$	$99.51_{\pm 0.11}$	$65.02_{\pm 3.23}$	69.56	1.52
Predictive	$76.15_{\pm 2.73}$	$93.43_{\pm 1.37}$	$70.54_{\pm 0.55}$	$99.08_{\pm 0.12}$	$63.87_{\pm 4.21}$	72.05	1.55
Prediction Head							
Linear	$76.67_{\pm 7.66}$	$92.51_{\pm 1.04}$	$70.31_{\pm 0.62}$	$99.01_{\pm 0.15}$	$64.10_{\pm 2.26}$	69.54	1.51
Nonlinear	$76.15_{\pm 2.73}$	$93.43_{\pm 1.37}$	$70.54_{\pm 0.55}$	$99.08_{\pm 0.12}$	$63.87_{\pm 4.21}$	72.05	1.55
Encoding method							
Channel Independence	$79.88_{\pm 2.63}$	$94.45_{\pm 0.78}$	$80.95_{\pm 0.50}$	$99.62_{\pm 0.06}$	$66.09_{\pm 1.69}$	70.67	11.14
Channel Mixing	$76.15_{\pm 2.73}$	$93.43_{\pm 1.37}$	$70.54_{\pm 0.55}$	$99.08_{\pm 0.12}$	$63.87_{\pm 4.21}$	72.05	1.55
Neighbor Identification							
Previous Patch	$73.77_{\pm 3.17}$	$92.95_{\pm 1.50}$	$70.44_{\pm 0.50}$	$99.00_{\pm 0.11}$	$62.04_{\pm 2.83}$	70.70	1.55
Next Patch	$76.15_{\pm 2.73}$	$93.43_{\pm 1.37}$	$70.54_{\pm 0.55}$	$99.08_{\pm 0.12}$	$63.87_{\pm 4.21}$	72.05	1.55
Effect of K							
$K = 2$	$76.15_{\pm 2.73}$	$93.43_{\pm 1.37}$	$70.54_{\pm 0.55}$	$99.08_{\pm 0.12}$	$63.87_{\pm 4.21}$	70.58	1.53
$K = 5$	$77.50_{\pm 4.53}$	$92.98_{\pm 2.00}$	$69.99_{\pm 0.63}$	$98.99_{\pm 0.21}$	$61.88_{\pm 2.88}$	72.05	2.79
Effect of P							
$P = 6$	$75.50_{\pm 0.76}$	$91.86_{\pm 1.55}$	$70.13_{\pm 0.36}$	$98.98_{\pm 0.09}$	$63.02_{\pm 3.93}$	70.93	1.76
$P = 10$	$76.15_{\pm 2.73}$	$93.43_{\pm 1.37}$	$70.54_{\pm 0.55}$	$99.08_{\pm 0.12}$	$63.87_{\pm 4.21}$	72.05	1.55
Encoder							
– FCN	$71.46_{\pm 3.88}$	$92.56_{\pm 1.28}$	$71.63_{\pm 0.62}$	$98.37_{\pm 0.14}$	$62.15_{\pm 1.95}$	70.24	0.58
– ResNet	$72.58_{\pm 1.98}$	$91.80_{\pm 0.96}$	$70.48_{\pm 1.04}$	$99.21_{\pm 0.12}$	$59.31_{\pm 3.32}$	71.10	0.92
– InceptionTime	$76.15_{\pm 2.73}$	$93.43_{\pm 1.37}$	$70.54_{\pm 0.55}$	$99.08_{\pm 0.12}$	$63.87_{\pm 4.21}$	72.05	1.55

Prediction Head. A nonlinear head marginally outperforms a linear head on UEA (72.05% vs 69.54%) with negligible difference on the large-scale benchmarks. This is consistent with findings in visual SSL (Chen et al., 2020) that a nonlinear projection absorbs task-specific invariances and leaves the encoder output more transferable. We adopt the nonlinear head as default.

Neighbour Identification Direction. Predicting the next patch consistently outperforms predicting the previous patch across all settings, with comparable training time. We attribute this to the causal structure of physical and biological time series, where the current state is more predictive of the future than the past is of the present. We adopt next-patch identification as the default.

Effect of K and P. At $K = 2$ vs $K = 5$, we observe different trends across evaluation settings. On the large-scale datasets, $K = 2$ consistently performs as well as or better than $K = 5$ while requiring substantially lower training time (1.53 vs. 2.79 hours). In contrast, on the 28 UEA datasets, $K = 5$ achieves the best overall accuracy. Based on these observations, we use $K = 2$ as the default setting for large-scale experiments and $K = 5$ for the UEA benchmark. For patch count, $P = 10$ outperforms $P = 6$ consistently across all datasets, with comparable training time (1.55 h vs 1.76 h). Fewer patches produce longer patches that reduce the diversity of temporal contexts seen during training. We adopt $P = 10$ as default. **Encoder Architecture.** InceptionTime achieves the best performance on most datasets and the highest UEA average (72.05%), while being marginally slower than FCN (1.55 h vs 0.58 h). FCN achieves competitive performance on PAMAP2 but underperforms InceptionTime on all other datasets. ResNet underperforms both on HARTH (91.80%), WISDM2 (59.31%), and UEA (71.10%), indicating that residual connections alone do not confer an advantage in this setting. All subsequent experiments use InceptionTime as default. **Batch Size.** Unlike contrastive methods whose training signal depends critically on batch composition, C-Pred's intra-instance objective is entirely independent of what other instances appear in the batch. Figure 4 confirms this: C-Pred performance is stable across batch sizes $\{32, 64, 128, 256, 512\}$. This batch-size independence is a practical advantage as C-Pred can be trained with small batch sizes without sacrificing representation quality, making it applicable in memory-constrained settings.

Channel Independence vs Channel Mixing. The most significant finding in our ablation concerns the channel encoding strategy. Under *channel mixing*, the default assumption of every SSL baseline in Table 1, all channels are processed jointly by the encoder. Under *channel independence*, each channel is processed separately by the same univariate encoder and representations are aggregated before the prediction head.

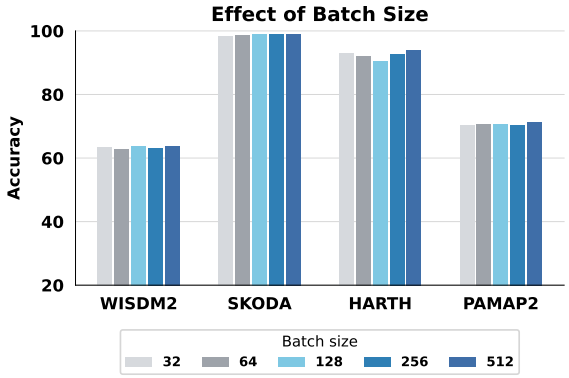

Figure 4: Effect of batch size on linear probing accuracy (%) across four representative large-scale datasets.

Channel independence produces substantial improvement on the large-scale datasets, at the cost of increased training time (10.76 h vs 1.52 h). Despite this overhead, the channel independence variant remains faster than TNC (11.48 h), the second best channel mixing method at 81.09%, while exceeding it by 3.27 pp. On the UEA benchmark, channel independence improves performance on several individual datasets compared to channel mixing (9 wins vs. 4 wins). However, despite these strong dataset-specific gains, the channel mixing variant achieves a higher overall average accuracy across all 28 datasets (72.05% vs. 70.67%), while also being substantially more computationally efficient (0.19 h vs. 0.63 h). The large increase in training time for channel independence arises from per-channel processing, which significantly increases the effective batch size and memory consumption on datasets with high channel dimensionality.

6 Conclusion

We present *C-Pred*, a self-supervised framework for multivariate time series classification based on context prediction. Given a reference patch, the encoder predicts its immediate temporal neighbour from K candidates within the same instance, forming a K-way classification task with a single correct answer and requiring no augmentation, cross-instance negative sampling, or reconstruction. C-Pred achieves competitive results across six large-scale benchmarks and 28 UEA datasets while maintaining the lowest training time among all methods. Our analysis surfaces two findings of broader significance. First, channel independence substantially outperforms channel mixing, despite most existing SSL method for time series classification using channel mixing. Beyond accuracy, channel independence enables a single univariate backbone to handle any channel dimensionality, directly addressing the varying channel count problem that limits scalable time series SSL and foundation model development. Second, C-Pred avoids the overfitting-to-pretext failure that caused proxy objectives to be abandoned in vision using hyperparameter K to provides a principled difficulty control mechanism. Together, these results suggest that the time series SSL community's focus on contrastive learning and masked modelling, while productive, has overlooked a simpler and more efficient family of objectives. A carefully reformulated context prediction task offers a principled and practical alternative that achieves strong empirical performance without the design complexity of existing approaches. We hope C-Pred serves as both a strong practical baseline and a conceptual argument for revisiting the full spectrum of self-supervised objectives in the time series domain.

Limitations. C-Pred does not completely avoid the challenges of periodic signals. In contrastive learning, periodic signals can lead to false negatives when patches that are far apart in time are actually very similar, since the objective treats them as negatives and encourages them to be separated from the anchor. C-Pred avoids this explicit positive-negative separation by focusing on predicting the next patch. However, periodicity can still create a similar issue. Among the K candidates, only the immediate temporal neighbor is treated as the correct answer. If a distant candidate happens to look very similar to the true next patch because of the periodic structure, the cross-entropy loss still penalizes the model for assigning probability to that candidate. Thus, C-Pred handles periodicity by defining the learning task around temporal order instead of pairwise similarity, with the immediate next patch always providing a single, well-defined target, even when other candidates may contain similar patterns.

REFERENCES

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018, 2018. URL https://arxiv.org/abs/1811.00075.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Ching Chang, Chiao-Tung Chan, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. Timedrl: Disentangled representation learning for multivariate time-series. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 625–638. IEEE, 2024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36:29996–30025, 2023.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 2352–2359, 2021.

Navid Mohammadi Foumani, Chang Wei Tan, Geoffrey I Webb, Hamid Rezatofighi, and Mahsa Salehi. Series2vec: similarity-based self-supervised representation learning for time series classification. *Data Mining and Knowledge Discovery*, 38(4):2520–2544, 2024.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1v4N2l0-.

Ary Goldberger, Luís Amaral, Leon Glass, Jeffrey Hausdorff, Plamen Ivanov, Roger Mark, Joseph Mietus, George Moody, Chung-Kang Peng, and H. Stanley. Physiobank, physiotoolkit, and physionet : Components of a new research resource for complex physiologic signals. *Circulation*, 101:E215–20, 07 2000. doi: 10.1161/01.CIR.101.23.e215.

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*, pp. 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 recognition*, pp. 16000–16009, 2022.

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

Seunghan Lee, Taeyoung Park, and Kibok Lee. Soft contrastive learning for time series. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=pAsQSWlDUf.

Aleksej Logacjov, Kerstin Bach, Atle Kongsvold, Hilde Bremseth Bårdstu, and Paul Jarle Mork. Harth: A human activity recognition dataset for machine learning. *Sensors (Basel, Switzerland)*, 21, 2021. URL https://api.semanticscholar.org/CorpusID:244808329.

George B Moody and Roger G Mark. A new method for detecting atrial fibrillation using r-r intervals. In *Computers in Cardiology*, pp. 227–230, 1983.

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.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016. doi: 10.1109/CVPR.2016.278.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*, pp. 108–109. IEEE, 2012.

Abdul-Kazeem Shamba, Kerstin Bach, and Gavin Taylor. Contrast all the time: Learning time series representation from temporal consistency. In *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pp. 2810–2817, 2025. URL https://doi.org/10.3233/FAIA251137.

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=8qDwejCuCN.

Gary M Weiss and Jeffrey W Lockhart. The impact of personalization on smartphone-based activity recognition. In *AAAI workshop on activity context representation: techniques and languages*, pp. 98–104. Toronto., 2012.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8980–8987, 2022.

Piero Zappi, Daniel Roggen, Elisabetta Farella, Gerhard Tröster, and Luca Benini. Network-level power-performance trade-off in wearable activity recognition: A dynamic sensor selection approach. *ACM Transactions on Embedded Computing Systems (TECS)*, 11(3):1–30, 2012.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2114–2124, 2021.

Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35:3988–4003, 2022.

Code Availability – To support reproducibility, the complete codebase and scripts for C-Pred are provided in the supplementary material and will be released publicly upon acceptance.

A Reproducibility and Computational Environment

To ensure reproducibility, we report all experimental settings and computational details used in our study. Experiments on the large-scale benchmarks are conducted over five random seeds $\{1, 2, 3, 4, 5\}$, and results are reported as averages across seeds. For the UEA benchmarks, we follow prior work and use a fixed random seed of 1. C-Pred uses $K = 2$ for the large scale datasets and $K = 5$ for the UEA benchmark datasets, with $P = 10$ throughout.

All experiments are implemented in Python 3.12.3 using PyTorch 2.5.1 with CUDA 12.4. Experiments are executed on NVIDIA Tesla V100S GPUs with 32 GB memory. The system uses NVIDIA driver version 570.172.08.

B Channel-Independent Encoding and Variable-Length Inference

For the channel-independent (CI) variant of C-Pred, each channel is processed independently using a shared univariate encoder. Given a multivariate time series $\mathbf{x} \in \mathbb{R}^{T\times C}$, we first transpose the input to obtain C univariate sequences,

$$\mathbf{x}^{(c)} \in \mathbb{R}^{T\times 1}, \qquad c = 1, \ldots, C.$$

The same encoder f_θ is then applied to every channel,

$$\mathbf{z}^{(c)} = f_\theta\left(\mathbf{x}^{(c)}\right), \qquad \mathbf{z}^{(c)} \in \mathbb{R}^d,$$

where the encoder parameters θ are shared across all channels. Thus, the number of encoder parameters does not depend on the number of input channels.

The channel representations are subsequently concatenated to form the final multivariate representation,

$$\mathbf{z} = \left[\mathbf{z}^{(1)}; \mathbf{z}^{(2)}; \ldots; \mathbf{z}^{(C)}\right] \in \mathbb{R}^{Cd}.$$

This aggregation preserves channel-specific information rather than mixing the channels before encoding. Consequently, unlike the standard C-Pred encoder, whose representation has dimension d, the CI representation has dimension Cd. For example, with $C = 52$ channels and an encoder output dimension of $d = 256$, the resulting representation has $52 \times 256 = 13{,}312$ dimensions.

Variable-length encoding. C-Pred is trained using patches of length L, where the reference and candidate patches are independently encoded by f_θ. At inference time, however, the pretrained encoder is applied directly to the complete input sequence of length T. This is possible because the temporal dimension is aggregated within the encoder using global average pooling (GAP). Specifically, if the final temporal feature map is

$$\mathbf{H} \in \mathbb{R}^{T'\times d},$$

GAP computes

$$\mathbf{z} = \frac{1}{T'}\sum_{t=1}^{T'} \mathbf{H}_{t,:} \in \mathbb{R}^d.$$

The resulting representation therefore has a fixed dimensionality d, independent of the input sequence length. Hence, the same encoder can process training patches of length L and full sequences of length

T without requiring padding, truncation, or a separate inference-time architecture. In the CI setting, this operation is performed independently for each channel before the resulting C representations are concatenated.

C BASELINE DESCRIPTIONS.

This appendix provides implementation details and reproduction settings for all baseline methods compared in this work. To ensure a controlled and fair comparison, we standardize the backbone architecture and data-loading pipeline across all methods. Specifically, all models are trained using a shared 1D InceptionTime backbone Ismail Fawaz et al. (2020), ensuring that performance differences reflect contrastive learning objectives rather than architectural choices. All models are optimized using AdamW with weight decay of 3×10^{-4} and momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.99$, under a cosine learning-rate schedule with linear warm-up during the first 10% of training iterations.

Learning rates are selected via a hyperparameter search over $\{1 \times 10^{-3}, 3 \times 10^{-4}, 1 \times 10^{-4}\}$ on the six large-scale datasets. Specifically, for C-Pred, we use learning rates 1×10^{-3} and 3×10^{-4} for the six large-scale and 28 UEA benchmarks respectively. For all methods, we use the official implementation with default method-specific hyperparameters as described in the original work.

D EXTENDED RELATED WORKS.

Self-Supervised Learning. Self-supervised learning (SSL) learns meaningful representations from unlabelled data by defining pretext tasks that exploit the inherent structure of the data. Early approaches formulated these as proxy objectives such as predicting image rotations (Gidaris et al., 2018), solving jigsaw puzzles (Noroozi & Favaro, 2016), or inferring spatial context (Doersch et al., 2015), encouraging models to capture semantic structure without manual annotations. However, these methods often suffer from a mismatch between the pretext objective and downstream tasks, leading to representations that overfit to the proxy task. Contrastive learning addressed this limitation by replacing fixed proxy tasks with a flexible objective that directly encodes desired invariances into the representation space. SimCLR (Chen et al., 2020) demonstrated that strong data augmentation combined with a normalised temperature-scaled cross-entropy loss is sufficient to learn highly transferable visual representations. MoCo (He et al., 2020) introduced a momentum encoder and memory bank to decouple batch size from the number of negatives, improving scalability. DINO (Caron et al., 2021) extended this to a self-distillation setting without explicit negatives, using a teacher-student architecture to prevent representation collapse. Masked image modelling subsequently emerged as a complementary paradigm, MAE (He et al., 2022) demonstrated that reconstructing masked patches of an image from visible context is sufficient to learn rich and transferable visual representations. These works collectively established that the choice of SSL objective and the structure of the supervisory signal critically determine representation quality, a finding that directly motivates our work.

Relation to CPC and TNC. CPC (Oord et al., 2018) learns representations by summarising past observations into a context vector with an autoregressive aggregator and predicting future latent codes at multiple horizons through an InfoNCE objective, with negatives drawn from the batch. TNC (Tonekaboni et al., 2021) exploits temporal neighbourhood structure by treating signals from nearby time windows as positives and distant windows as negatives. C-Pred shares with CPC and TNC the premise that temporal proximity provides a free supervisory signal; we therefore state the differences explicitly. CPC predicts *future* latent codes from an autoregressive summary of the past, requiring a context aggregator over the observed prefix, a separate prediction matrix per forecast horizon, and negatives drawn predominantly from other instances in the batch. C-Pred has neither an aggregator nor horizon-specific parameters: the reference is a single patch, and all candidates, the true neighbour and the distractors, are drawn from the same instance, so instance-level cues such as amplitude, baseline drift, or subject identity carry no discriminative information and the encoder must rely on local temporal structure. TNC defines an explicit neighbourhood around a reference window whose width is set by a stationarity test, samples positives inside and negatives outside it, and scores each pair independently with a binary discriminator; because a distant window may belong to the same underlying state, TNC must correct the resulting false negatives with Positive-Unlabelled learning. C-Pred requires neither neighbourhood-width estimation nor a PU correction, because its supervision is *relative* rather than absolute. Finally, K is an explicit difficulty parameter decoupled

from batch size, whereas the number of negatives is tied to the batch in CPC and to the sampled neighbourhood in TNC.

E Detailed Dataset Information.

We use publicly available time series datasets previously employed in self-supervised representation learning research. The WISDM2, PAMAP2, SLEEP and SKODA datasets were directly obtained from the repository linked by the Series2Vec project (Foumani et al., 2024), without applying additional preprocessing. For the ECG dataset, we generate random windows for self-supervised learning following Tonekaboni et al. (2021). Each subject's continuous 12-hour recording was split into 10-minute segments to reduce redundancy, and during training, a random window of 2,500 time steps was sampled from each segment. The first 5 subjects were held out for testing, with the remaining subjects used for training. This strategy increases variability in the training data while maintaining consistent input dimensions across subjects. For linear probing and kNN evaluation, we extract a deterministic window of 2,500 time steps from the middle of each segment from the train set to ensure reproducible evaluation. For the HARTH dataset, we performed subject-wise split with 22 subjects in total. The first 18 subjects were used for training and the remaining 4 for testing. Each subject's multivariate time series was split into overlapping windows of 500 time steps with a stride of 250, and the majority class within each window was assigned as the label.

Table 4 summarizes the key characteristics of each dataset. All datasets are split using subject-wise splitting, the SKODA dataset, collected from a single subject, however, does not have subject-wise splits.

Table 4: Summary of datasets used in the large-scale experiments.

Dataset	Train set	Test set	Subject	Sequence Length	Channel	Class
ECG	59,922	16,645	23	2,500	2	4
HARTH	23,025	2,785	22	500	6	18
PAMAP2	51,192	11,590	9	100	52	18
SKODA	22,587	5,646	1	50	64	11
SLEEP*	25,612	8,910	20	3,000	1	5
WISDM2	134,614	14,421	29	40	3	6

* SLEEP is the only univariate dataset in our experiments. To accommodate methods designed exclusively for multivariate time series, the single-channel signal is duplicated to form a two-channel input.

F Full UEA Benchmarks Results.

Table 5 present the full 28 UEA benchmark results. The UEA archive Bagnall et al. (2018) originally contains 30 datasets. In our experiments, we report results on 28 datasets, excluding *InsectWingbeat* and *EigenWorms*. These two datasets exhibit atypical input shapes and extremely large memory requirements, which lead to runtime or memory errors for several baseline methods despite consistent preprocessing and hyperparameter settings. We note that this behavior is not unique to our evaluation protocol. Prior work such as TS2Vec Yue et al. (2022) similarly omits a comparable subset of datasets, reporting results on 29 UEA datasets, due to analogous issues related to input dimensionality, memory constraints, or training instability.

Table 5: Classification accuracy (%) across UEA datasets.

Dataset	CaTT	w/o Pretrain	SimMTM	Soft	TF_C	TNC	TS2Vec	TS_TCC	TimeDRL	C-Pred	C-Pred (CI)	w/o Pretrain (CI)
ArticularyWordRecognition	98.33	96.33	91.33	97.33	97.33	97.33	97.00	97.33	95.00	97.33	**98.67**	92.67
AtrialFibrillation	26.67	26.67	20.00	33.33	33.33	26.67	20.00	26.67	26.67	**40.00**	33.33	26.67
BasicMotions	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
CharacterTrajectories	99.23	97.98	97.77	**99.51**	**99.51**	99.37	99.30	99.30	99.30	99.30	98.82	95.54
Cricket	86.11	98.61	88.89	97.22	95.83	**100.00**	91.67	97.22	98.61	98.61	98.61	97.22
DuckDuckGeese	46.00	60.00	54.00	52.00	**62.00**	54.00	54.00	52.00	50.00	60.00	60.00	54.00
ERing	86.30	80.37	89.26	92.22	90.37	88.52	89.63	85.56	**92.96**	92.22	92.22	66.30
Epilepsy	91.30	93.48	94.93	93.48	92.03	95.65	92.03	92.75	94.20	93.48	**97.10**	96.38
EthanolConcentration	27.38	30.80	27.38	28.14	26.24	25.86	30.42	26.24	30.04	25.48	**35.36**	34.22
FaceDetection	52.44	52.67	52.01	52.33	54.60	52.38	52.38	52.89	54.28	52.41	**56.84**	53.77
FingerMovements	60.00	56.00	49.00	56.00	52.00	**63.00**	47.00	58.00	47.00	56.00	61.00	49.00
HandMovementDirection	25.68	**35.14**	32.43	27.03	32.43	28.38	**35.14**	32.43	28.38	33.78	33.78	33.78
Handwriting	53.29	38.71	21.65	53.41	49.65	53.65	47.18	50.71	**58.35**	56.00	50.24	30.94
Heartbeat	75.12	74.15	73.66	69.27	74.63	73.17	75.12	70.73	74.15	**77.07**	72.68	73.17
JapaneseVowels	98.11	96.49	94.32	95.95	96.76	97.57	95.68	97.03	97.57	97.03	**98.65**	97.03
LSST	50.53	50.49	51.62	50.57	51.34	49.92	50.12	49.76	48.58	49.27	57.34	**61.72**
Libras	**88.33**	60.56	70.00	80.00	79.44	86.67	77.22	77.78	75.00	81.11	87.78	70.00
MotorImagery	56.00	54.00	55.00	44.00	53.00	**57.00**	54.00	53.00	50.00	56.00	49.00	53.00
NATOPS	92.78	83.89	81.11	**95.00**	88.33	91.11	87.22	85.56	89.44	90.56	93.33	83.33
PEMS-SF	75.72	77.46	63.58	79.19	73.99	78.03	**80.92**	76.30	75.72	**80.92**	72.25	74.57
PenDigits	96.88	97.03	**97.88**	97.66	97.06	97.03	97.77	97.43	97.74	97.03	97.00	97.00
PhonemeSpectra	24.34	19.33	22.85	23.83	26.33	23.50	25.35	24.10	23.59	23.83	**26.45**	21.06
RacketSports	81.58	83.55	79.61	83.55	83.55	79.61	**86.18**	81.58	76.97	78.29	82.24	84.87
SelfRegulationSCP1	78.16	86.01	82.25	82.94	86.69	86.01	80.89	87.03	86.69	86.35	**87.37**	80.89
SelfRegulationSCP2	**55.00**	48.89	47.22	50.00	52.22	48.89	50.00	46.67	54.44	51.67	45.00	51.67
SpokenArabicDigits	97.09	93.36	97.54	97.32	97.04	97.27	97.45	97.00	97.77	97.23	**99.00**	95.41
StandWalkJump	40.00	33.33	33.33	40.00	40.00	40.00	26.67	46.67	20.00	**60.00**	13.33	26.67
UWaveGestureLibrary	71.56	58.44	74.38	77.81	80.62	80.00	79.06	84.06	82.81	**86.56**	81.25	48.12
Average	69.07	67.28	65.82	69.61	70.23	70.38	68.55	69.49	68.76	**72.05**	70.67	66.04

G TSNE PLOTS

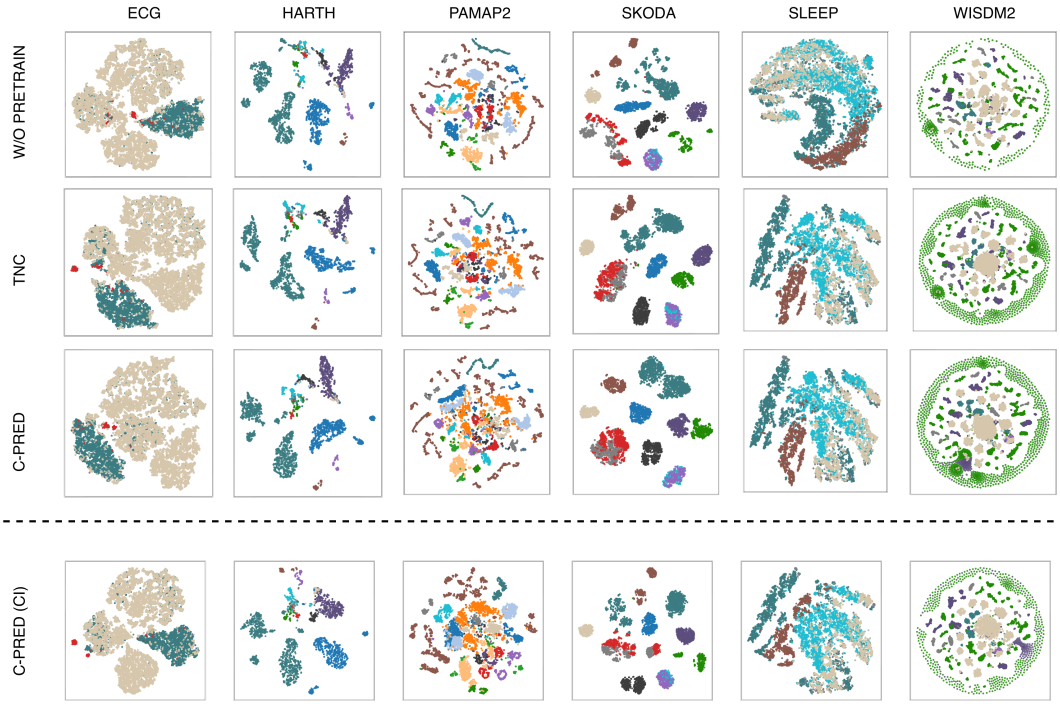

Figure 5: t-SNE visualizations of learned representations on the six large datasets