# Divide and Contrast: Learning Robust Temporal Features without Augmentation


NorwAI Norwegian Research Center for AI Innovation · NTNU

Abdul-Kazeem Shamba  Kerstin Bach  Gavin Taylor


Project Page

ICML International Conference On Machine Learning

## Motivation

**Self-supervised learning (SSL) for time series**

- Learns representations from raw temporal signals **without labels** to reduce reliance on annotation while improving downstream performance
- Contrastive learning is the dominant paradigm in SSL for time series that involves constructing positive/negative pairs via **augmentation** or temporal assumptions

**Contrastive learning in time series in challenging**

- Handcrafted data augmentation or masking strategies distort time series
- High computational cost from sequence length $T$ dependent operations
- Multiple encoder passes for each view and large similarity matrices scale with sequence length $T$

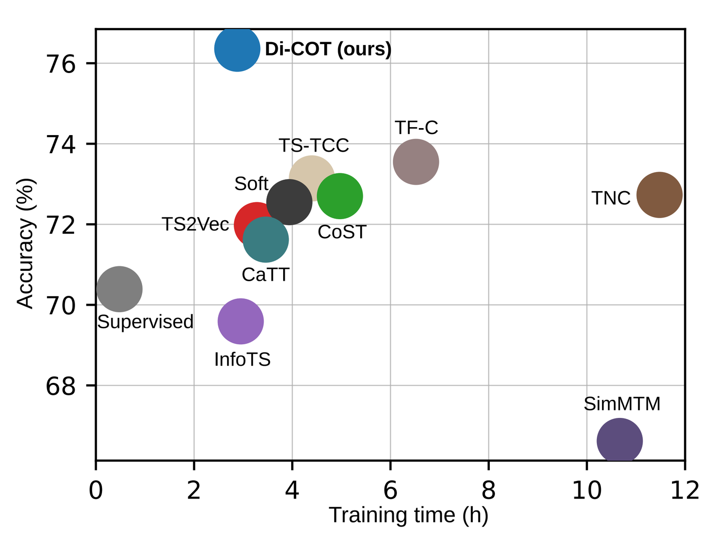
Handcrafted augmentation reduces scalability and introduce bias in representation learning

**Why not contrast sub-structures within a sequence instead of handcrafted augmentated views?**

## Linear Evaluation with Frozen Backbone

5 seeds · 6 datasets ($>20$k instances) · 1% linear probe evaluation

- Supervision derived purely from **internal structure** of a sequence
- Sub-structure contrast reduces dependency on sequence length $T$
- Supervised method struggles to learn meaningful representation under **limited labels**



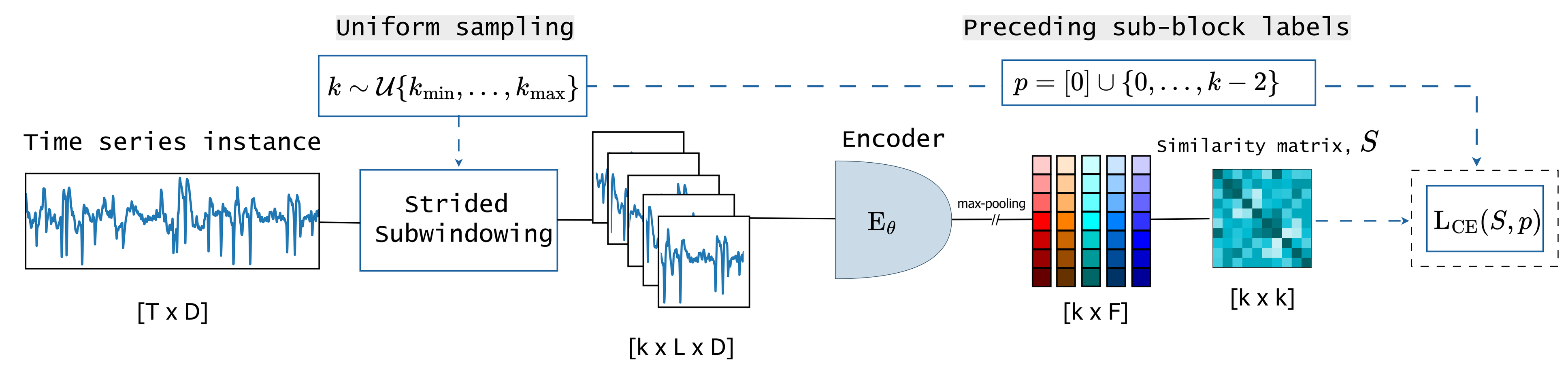

## Dynamic sub-block contrasting

**Di-COT** learns generalizable and transferable features by performing dynamic within-instance sub-block contrasting, which

- Eliminates the need for augmentation, masking, and multiple encoder passes
- Provides **dense supervision** within a single instance through rolling adjacent contrast
- Reduces computational overhead while **preserving temporal semantics**

## Divide and Contrast (Di-COT) Framework

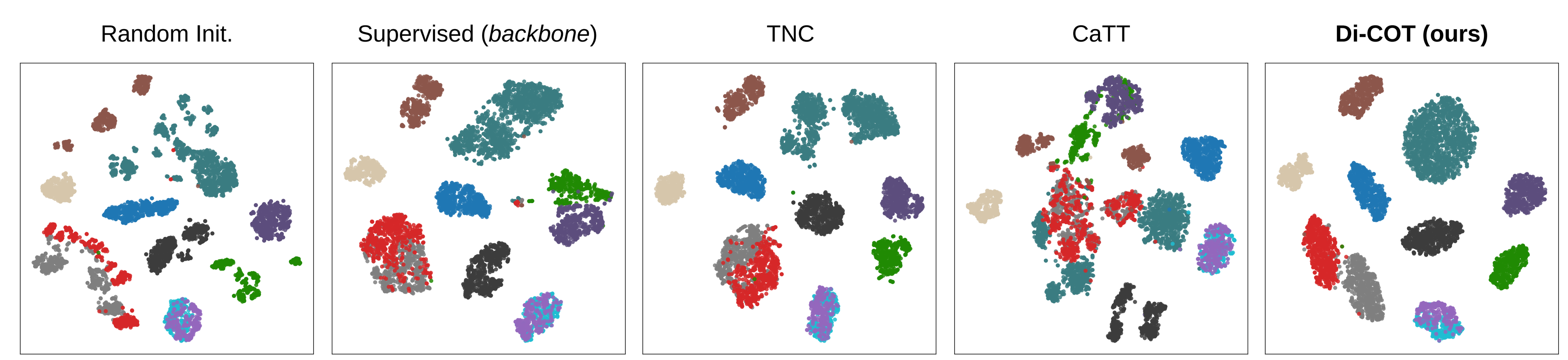
Each time series instance is stochastically divided into $k$ overlapping sub-blocks, which are encoded and contrasted via a preceding-sub-block predictive objective based on their pairwise similarities




1. Stochastic $k$-sampling exposes the model to diverse temporal contexts and granularities
2. Rolling adjacent contrast encourages locally consistent representations
3. Overlapping sub-blocks and stochastic partitioning provide **diverse shifted views** of the same signal

## Do Similar Samples Cluster? (t-SNE)

t-SNE visualizations of the learned embeddings on the SKODA test dataset across self-supervised methods, supervised training, and random initialization



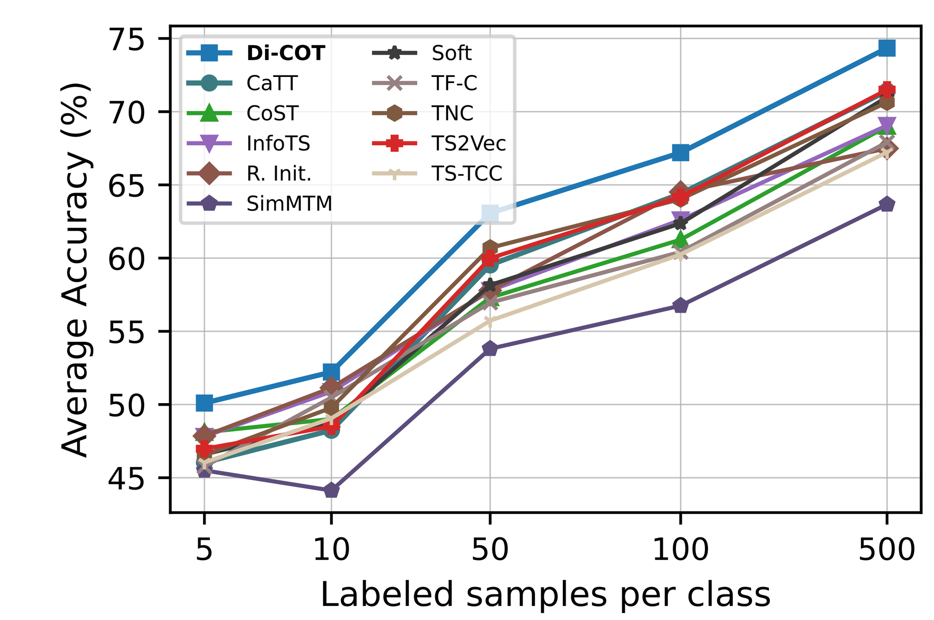

- **Di-COT** cleanly separates the challenging `red` and `gray` classes, which remain entangled in competing representations
- Even a **supervised backbone** trained on the full labeled dataset fails to achieve similar class separation, suggesting limited generalization

| Metric | Di-COT | Sup_B | CoST | TNC | InfoTS | Soft | R. Init | CaTT | TF-C | TS2Vec | TS-TCC | SimMTM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NMI | **0.508** | 0.493 | 0.486 | 0.491 | 0.496 | 0.475 | 0.470 | 0.408 | 0.459 | 0.466 | 0.458 | 0.448 |
| ARI | **0.406** | 0.376 | 0.379 | 0.375 | 0.375 | 0.359 | 0.367 | 0.262 | 0.361 | 0.338 | 0.346 | 0.359 |
| AvgRank | **2.92** | 4.33 | 4.58 | 5.42 | 7.83 | 6.83 | 7.08 | 7.67 | 7.67 | 7.58 | 7.83 | 8.25 |

Clustering performance (NMI, ARI) averaged across five random seeds and six large-scale publicly available datasets

## Can Neighbors Predict Labels?

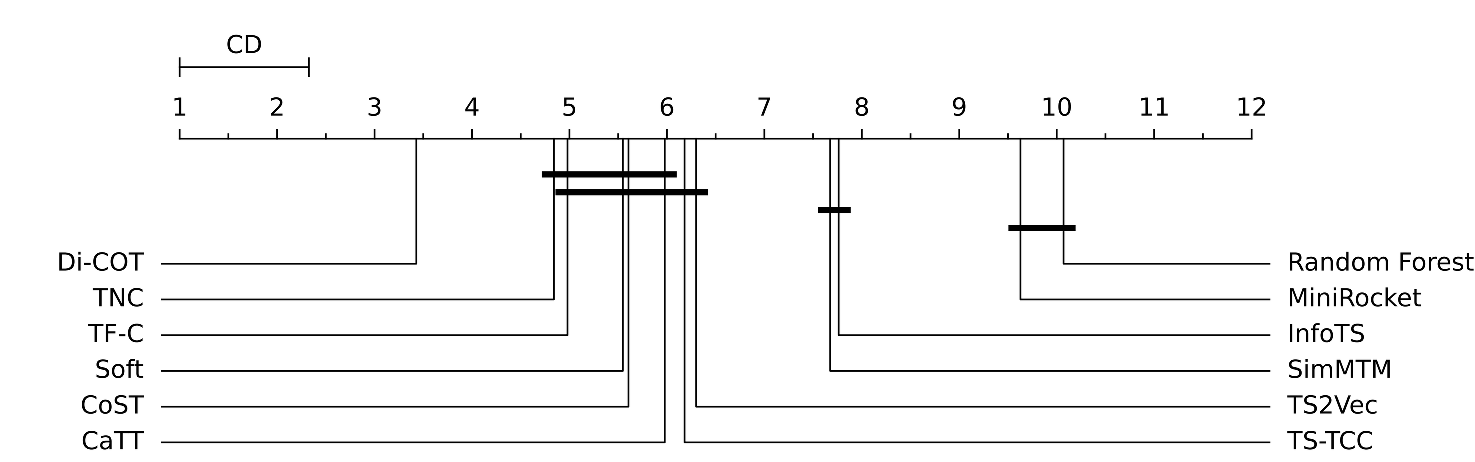
**Non-parametric $k$NN Evaluation**



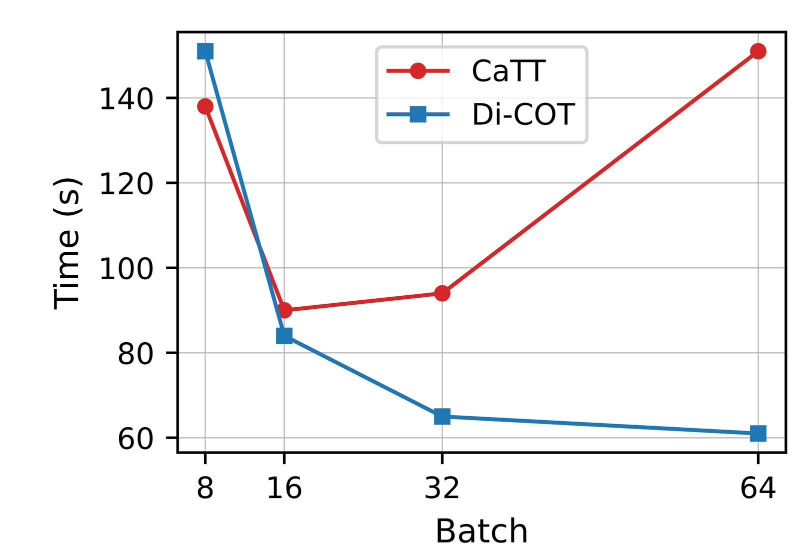

### Findings

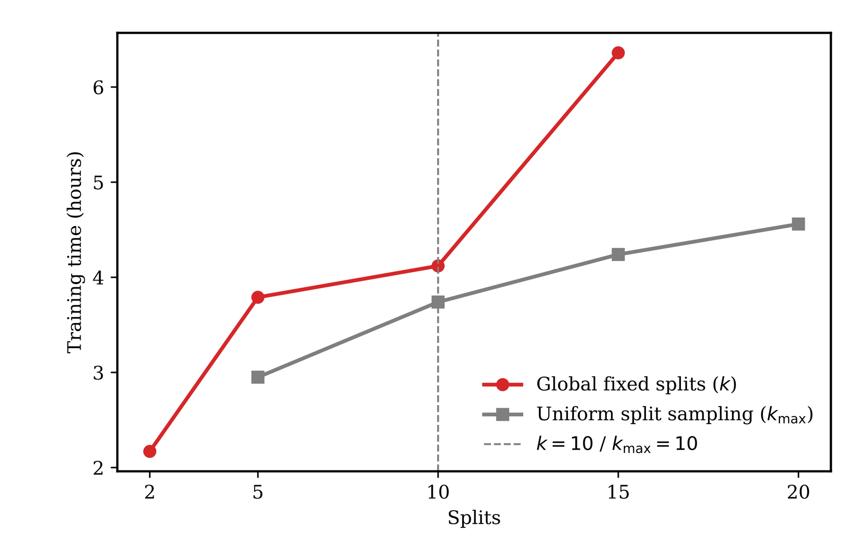
- **Best overall** with the highest average accuracy
- **Consistent gains** across label regimes
- **Robust** and strong performance even with very small labeled sets
- **Stable** results across 5 random seeds and 6 large-scale datasets

**Critical Difference (CD) diagram of SSL methods across all dataset categories with a confidence level of 95%**




## Algorithm and Computational Complexity




(**Left**) Di-COT improves efficiency by operating on sub-blocks instead of full sequences, avoiding timestep-level **quadratic cost** with increasing batch size (**Right**) Introducing variability in the number of splits keeps training efficient