Unsupervised Scalable Representation Learning for Multivariate Time Series

Jean-Yves Franceschi, Aymeric Dieuleveut, Martin Jaggi

Unsupervised Scalable Representation Learning for Multivariate Time Series

Jean-Yves Franceschi,[1] Aymeric Dieuleveut,[2,3] Martin Jaggi[2]

[1]Sorbonne Université, CNRS, LIP6, F-75005 Paris, France (previously at ENS de Lyon and EPFL)
[2]MLO, EPFL, Lausanne CH-1015, Switzerland
[3]CMAP, Ecole Polytechnique, Palaiseau, France

Motivation

- Time series are:
  - mostly unlabeled
  - potentially long
  - of unequal length in the same dataset
- Previous work does not tackle these issues simultaneously:
  - most of the time supervised (Bagnall et al., 2017)
  - not scalable (Malhotra et al., 2017)
  - tested on too few datasets with no code available (Malhotra et al., 2017; Wu et al., 2018)
- Objectives of this work:
  - learn **unsupervised** time series representations,
  - in a **scalable** way,
  - for time series of potentially **unequal lengths**,
  - suitable to and extensively tested on **various tasks**

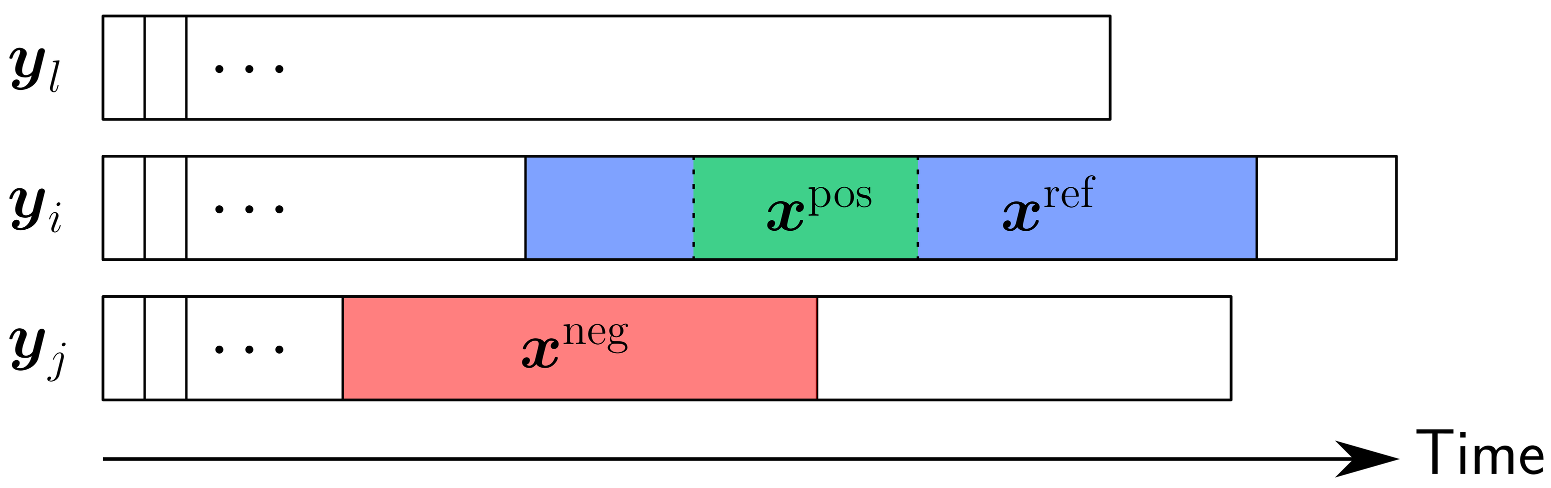


Figure: Choices of $\boldsymbol{x}^{\mathrm{ref}}$, $\boldsymbol{x}^{\mathrm{pos}}$ and $\boldsymbol{x}^{\mathrm{neg}}$.

Unsupervised training

- Encoder network $\boldsymbol{f}$ taking as input time series of arbitrary length
- Training with a triplet loss:
  - challenge: selecting similar and dissimilar inputs without supervision
  - problem: no unsupervised triplet loss has been proposed for time series yet
  - proposed solution: time-based triplet loss
  - inspired by CBOW and word2vec models
- Procedure and *analogies*:
  - choose $\boldsymbol{x}^{\mathrm{pos}}$ in some $\boldsymbol{y}_i$: *word*
  - choose $\boldsymbol{x}^{\mathrm{ref}}$ in $\boldsymbol{y}_i$ containing $\boldsymbol{x}^{\mathrm{pos}}$: *context*
  - choose $\boldsymbol{x}_k^{\mathrm{neg}}$ in some $\boldsymbol{y}_{j_k}$: *random word*
  - optimize the loss:

$$-\log\left(\sigma\left(\boldsymbol{f}\left(\boldsymbol{x}^{\mathrm{ref}}, \boldsymbol{\theta}\right)^\top \boldsymbol{f}\left(\boldsymbol{x}^{\mathrm{pos}}, \boldsymbol{\theta}\right)\right)\right) - \sum_{k=1}^{K} \log\left(\sigma\left(-\boldsymbol{f}\left(\boldsymbol{x}^{\mathrm{ref}}, \boldsymbol{\theta}\right)^\top \boldsymbol{f}\left(\boldsymbol{x}_k^{\mathrm{neg}}, \boldsymbol{\theta}\right)\right)\right)$$

- Desirable properties:
  - simple and efficient:
    - does not require a decoder
    - the cost of an iteration is linear in the cost of evaluating and backpropagating through $\boldsymbol{f}$
    - if $\boldsymbol{x}^{\mathrm{pos}}$ and $\boldsymbol{x}^{\mathrm{neg}}$ are chosen of the same length, their representations can be computed in parallel
    - memory can be optimized by performing backpropagatation per term
  - acts on time series of arbitrary length

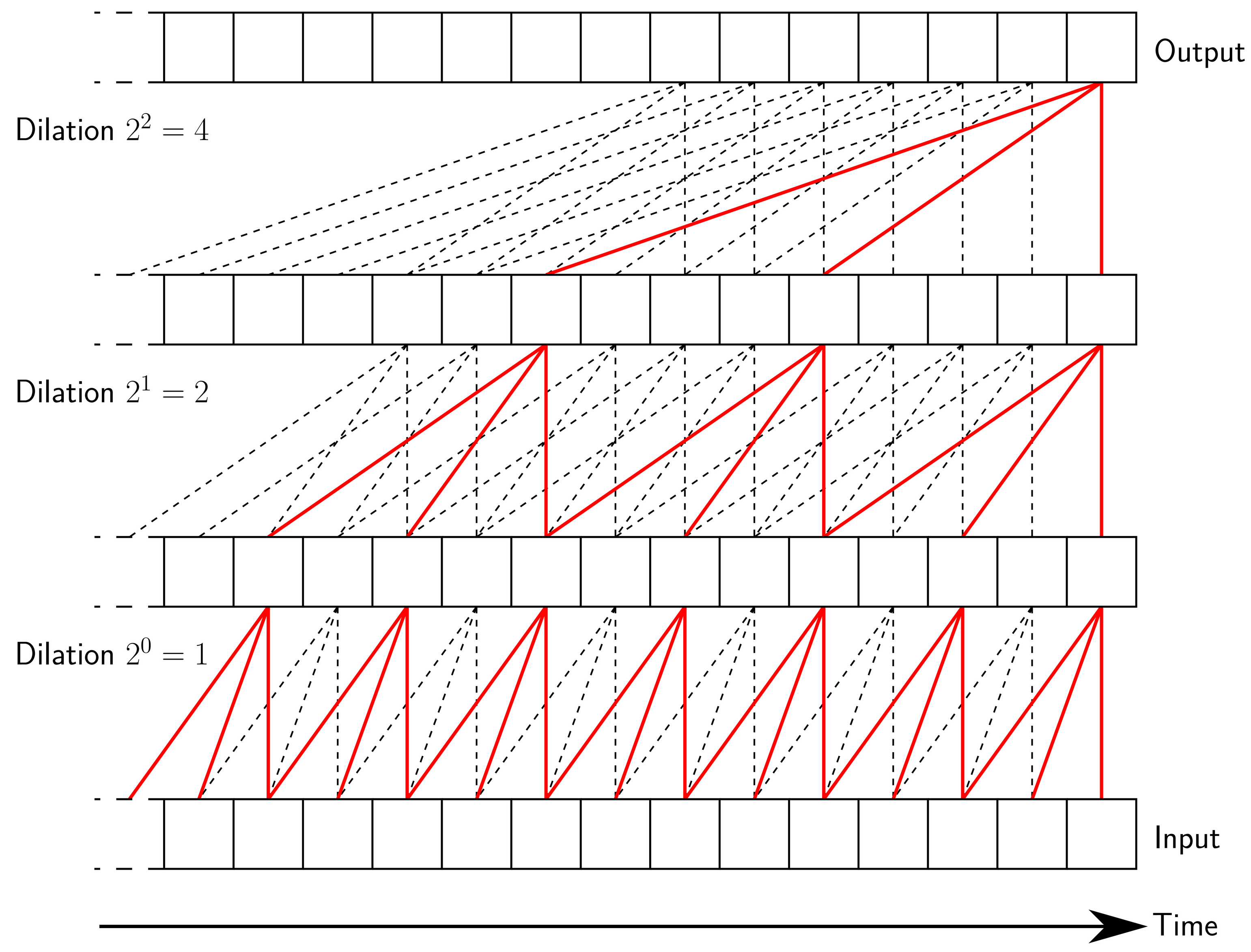


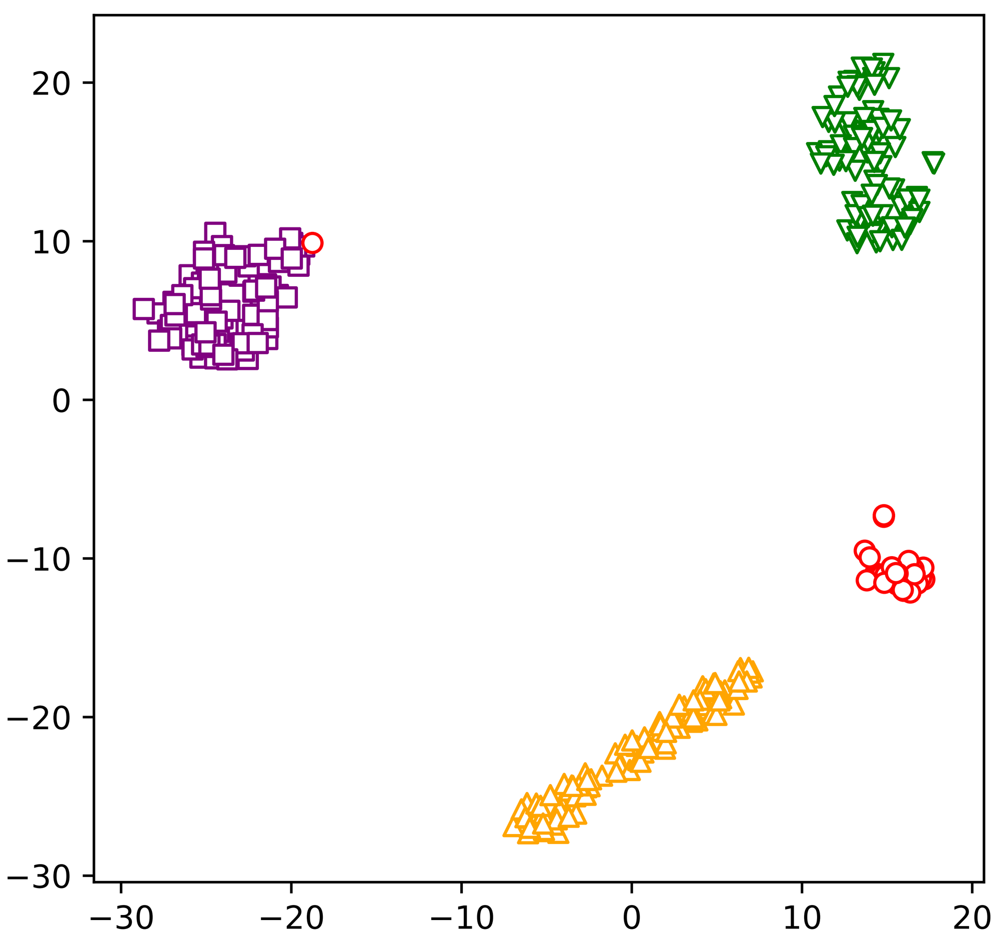
Figure: Principle of causal CNNs with exponentially dilated convolutions.

Encoder Architecture

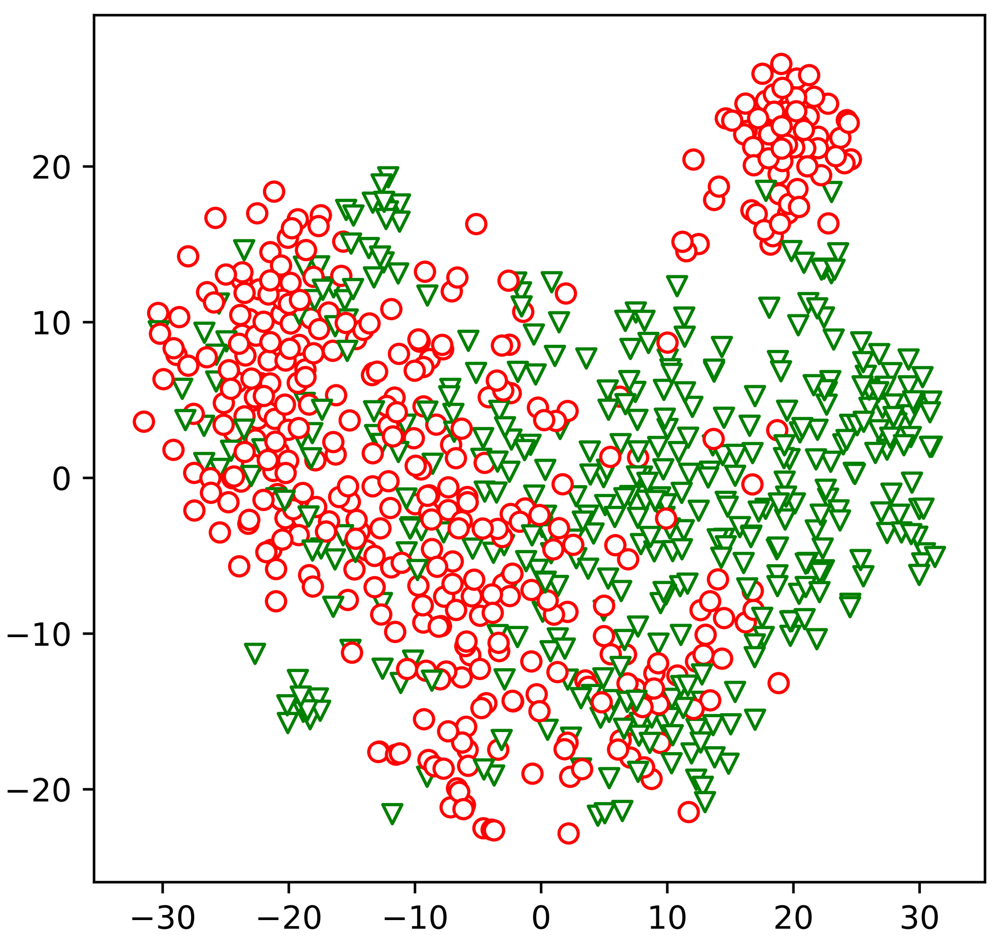
- We use a neural network based on *exponentially dilated convolutions* rather than a recurrent network:
  - more efficient and parallelizable on modern hardware
  - exponentially increasing dilation allows to increase the receptive field at constant depth
  - good performance on time series for other tasks (Bai et al., 2018; Ismail Fawaz et al., 2019)
  - experimentally performs better in our experiments
- We make the network *causal*:
  - maps a sequence to a sequence of the same length
  - each output element only depends on input values with lower time indices
  - can help to save computation time when adding an element to a time series
- The global architecture is sequentially shaped by:
  - a causal network formed with exponentially dilated convolutions associated with:
    - weight normalization
    - leaky ReLU
    - residual connections
  - a global max pooling layer squeezing the temporal dimension and aggregating temporal information in a fixed-size vector
  - a final linear transformation

Training

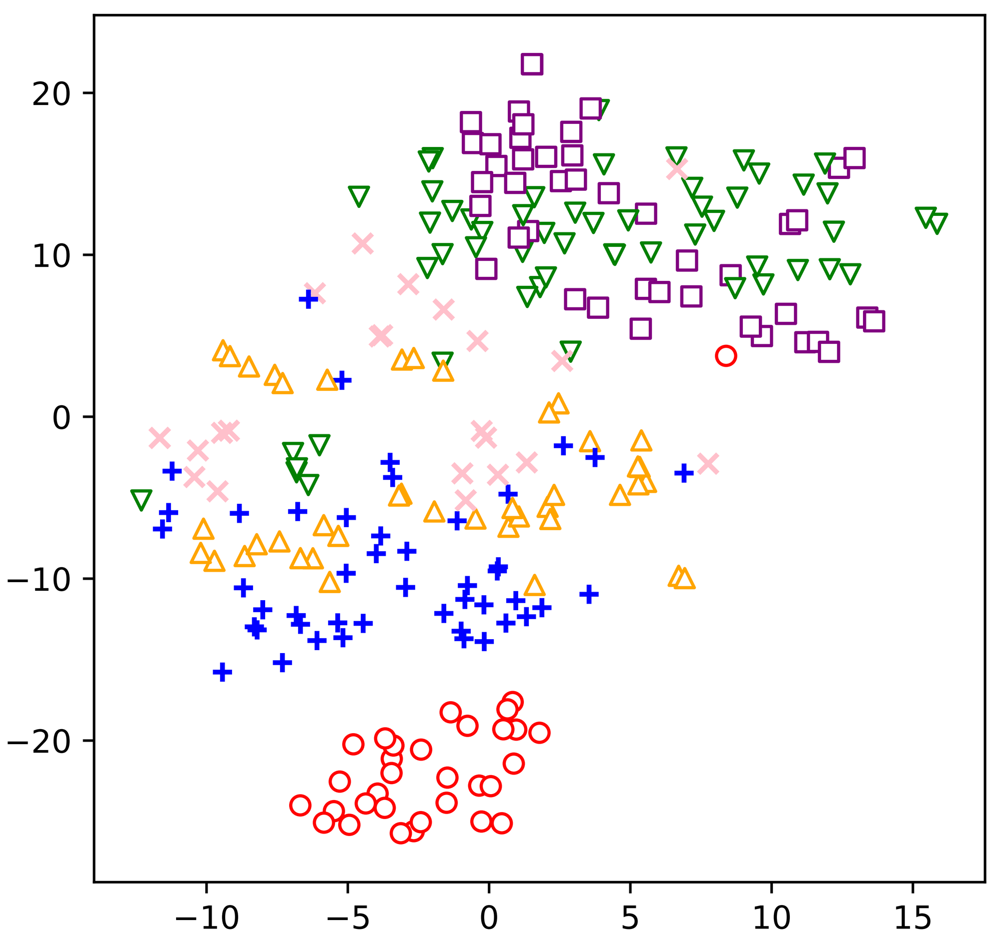
- Encoder training and testing performed on a single GPU
- No labels used during encoder training
- No hyperparameter optimization
- Open-source code, pretrained models and hyperparameters available
- Examples of dimensionality reduction plots using t-SNE:



(a) DiatomSizeReduction.



(b) FordB.



(c) OSULeaf.

Unsupervised Scalable Representation Learning for Multivariate Time Series

Jean-Yves Franceschi,[1] Aymeric Dieuleveut,[2,3] Martin Jaggi[2]

[1]Sorbonne Université, CNRS, LIP6, F-75005 Paris, France (previously at ENS de Lyon and EPFL)
[2]MLO, EPFL, Lausanne CH-1015, Switzerland
[3]CMAP, Ecole Polytechnique, Palaiseau, France

Classification

- Protocol:
  - unsupervised training of the encoder on the train dataset
  - training of an SVM with RBF kernel on top of the learned features with the train labels
- Results on the full UCR archive (Dau et al., 2018):
  - we outperform previous unsupervised state-of-the-art methods by a large margin on the few datasets they were tested on
  - we achieve close to state-of-the-art performance when comparing to supervised methods
- Tests were also performed on multivariate time series

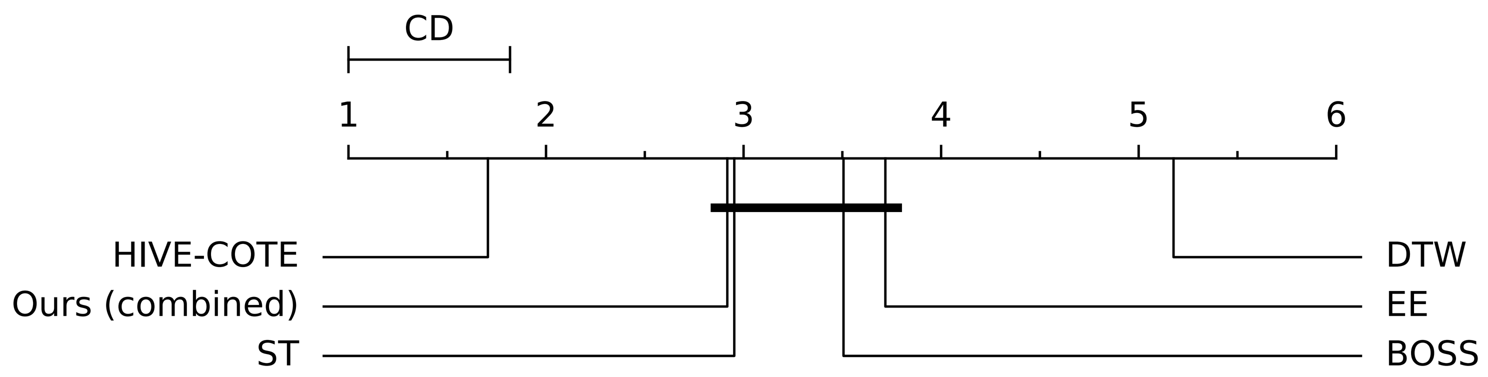


Figure: Mean ranks of compared methods.

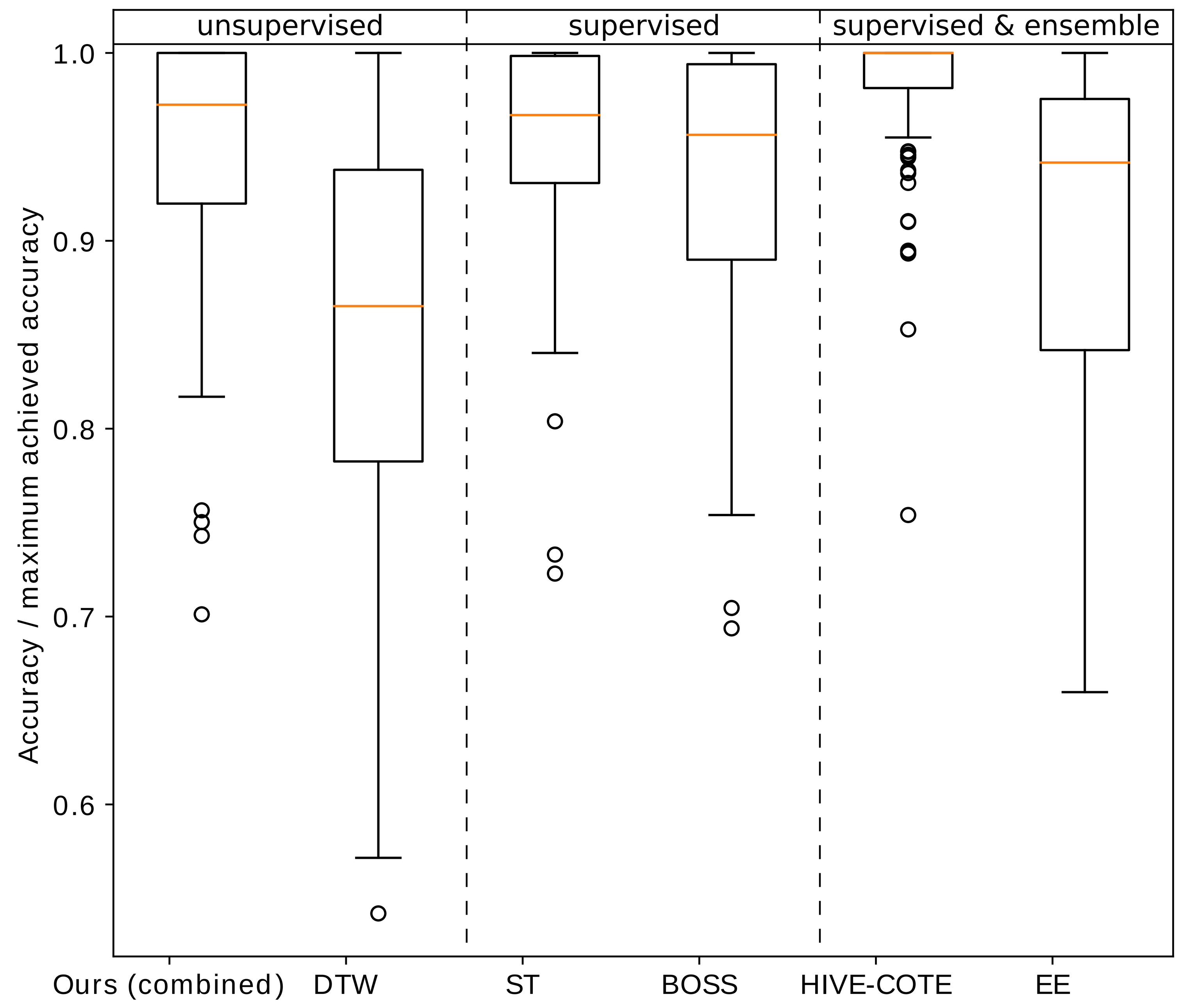


Figure: Boxplot of the ratio of the accuracy versus maximum achieved accuracy.

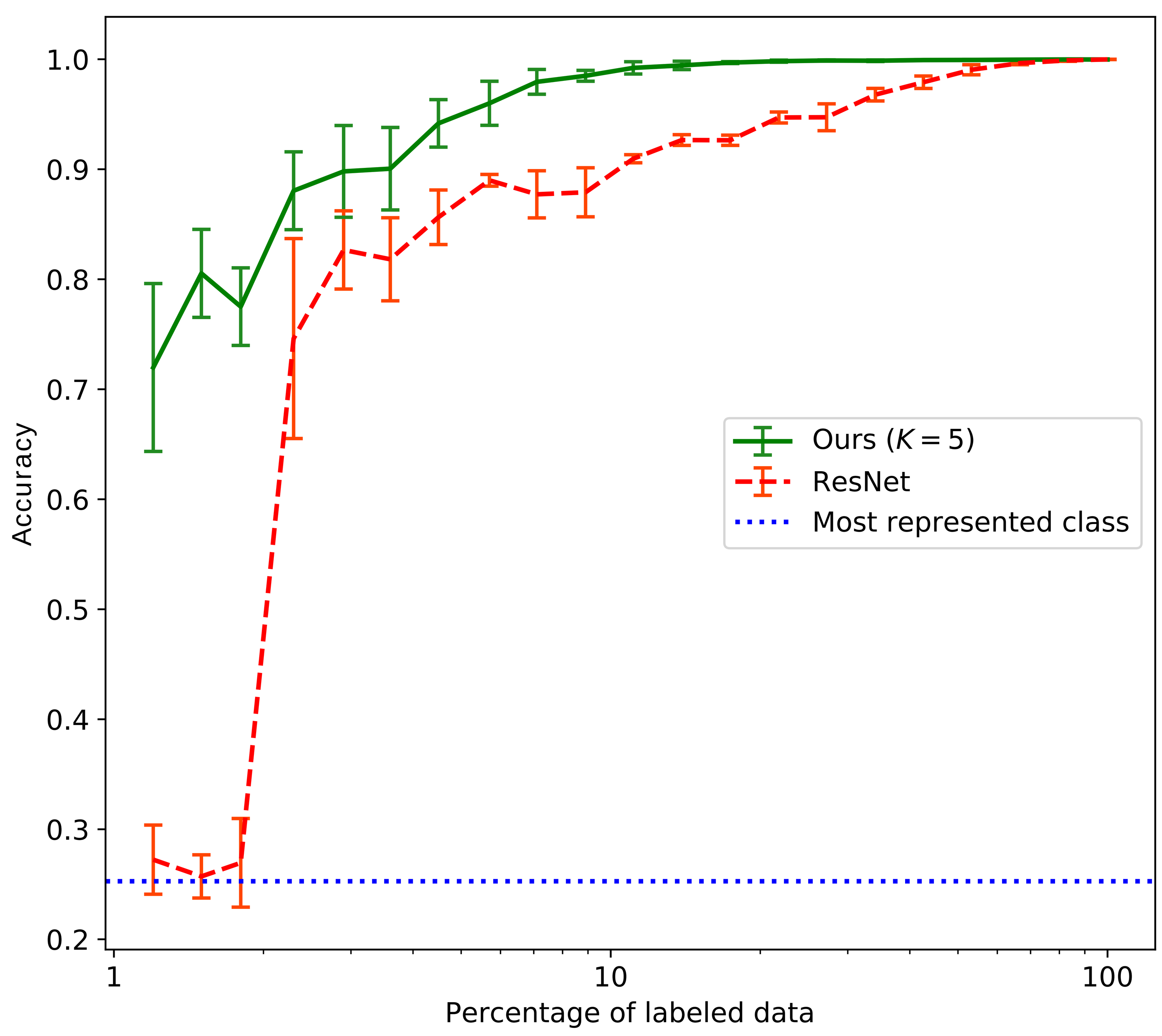


Figure: Accuracy of ResNet and our method with respect to the ratio of labeled data on TwoPatterns.

Additional Features

- Our unsupervised method can be applied in a sparse labeling setting, where it outperforms state-of-the-art deep neural networks
- Learning a one-nearest-neighbor classifier allows to outperform DTW which uses the same classifier on raw data
- The learned representations are transferable across datasets

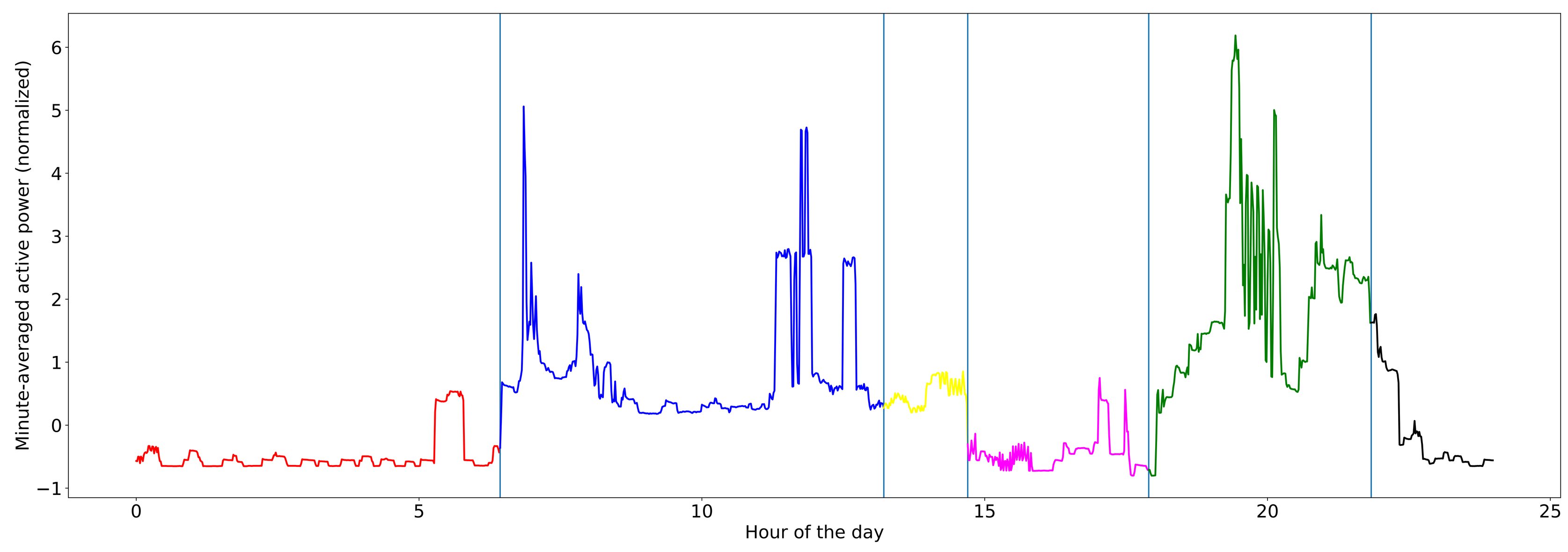


Figure: Subseries of the IHEPC dataset, with clustering induced by learned representations.

Moving Average Prediction

- IHEPC dataset:
  - minute-averaged electricity consumption of a single household for four years
  - single unlabeled time series of length $\approx 2\,000\,000$
- Encoder on such a long time series is trained in a few hours
- Linear regressors on raw data versus learned representations for moving average prediction:
  - task: predict next day / quarter average from the previous day / quarter data
  - regressors on raw data show slightly better results
  - regressors on learned representations are much more efficient
- The learned representations can be leveraged at different time scales

Table: Results obtained on the IHEPC dataset.

Task	Metric	Representations	Raw values
Day	Test MSE	$\mathbf{8.92 \times 10^{-2}}$	$\mathbf{8.92 \times 10^{-2}}$
	Wall time	**12s**	3min 1s
Quarter	Test MSE	7.26×10^{-2}	$\mathbf{6.26 \times 10^{-2}}$
	Wall time	**9s**	1h 40min 15s

References

Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31 (3):606–660, May 2017.

Bai, S., Kolter, J. Z., and Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Dau, H. A., Keogh, E., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., Ratanamahatana, C. A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., and Batista, G. The UCR time series classification archive, October 2018.

Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, March 2019.

Malhotra, P., TV, V., Vig, L., Agarwal, P., and Shroff, G. TimeNet: Pre-trained deep recurrent neural network for time series classification. *arXiv preprint arXiv:1706.08838*, 2017.

Wu, L., Yen, I. E.-H., Yi, J., Xu, F., Lei, Q., and Witbrock, M. Random Warping Series: A random features method for time-series embedding. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 793–802. PMLR, April 2018.

The Thirty-third Conference on Neural Information Processing Systems – NeurIPS 2019, Vancouver, Canada