We would like to thank the reviewers for their comments and suggestions. Below, we address the points raised by the reviewers; we will make sure to update our manuscript accordingly.

1 Comparison with Other Unsupervised Methods (R1)

The main paper **does already include** (lines 218-222) a comparison of our method with the two most recent and significant unsupervised methods in the literature, TimeNet [1] and RWS [2], which we consistently outperform.

In particular, TimeNet is a seq2seq method relying on an antoencoding loss and using LSTMs as encoder and decoder. While classically used in NLP, such methods did not receive much attention in the time series community apart from TimeNet, and notably do not scale to long time series (as explained on lines 144-157), unlike ours.

2 Additional Experiments (R1, R2, R3)

Comparison with different losses (R1, R2, R3) As we focus on scalability, we did not train our encoder with an autoencoding loss. However, we did perform experiments on some datasets with different loss variants. Replacing the word2vec loss with a triplet margin loss and choosing positive samples close to the anchor instead of being included in the anchor segment performs similarly to our triplet loss, but would require more hyperparameter tuning. In the end, we chose to present the loss which is simplest to tune, which is desirable in the context of unsupervised learning.

Comparison with different encoders (R3) The encoder architecture can indeed play an important role in the performance of the unsupervised training. As we aimed at achieving scalability of the encoder, we did not provide a comparison between different types of encoders. However, preliminary experiments indicate that using an LSTM to replace the causal CNN resulted in significantly worse results, besides increasing the computational cost of the method. Despite the poor computational scalability of LSTMs to long sequences, we will include additional comparisons between LSTMs and CNNs to shed more light on the importance of the encoder architecture.

SVMs vs kNNs (R3) Preliminary tests done by training kNNs on the learned embeddings suggest that they can achieve performance close to the one of SVMs, outperforming DTW. Thus, distances between our learnt embeddings are indeed meaningful. We will add insights on this matter to the paper.

Clustering (R1) We will include 2D visualizations of the computed embeddings for some datasets obtained using a dimensionality-reduction technique such as t-SNE.

3 Multivariate Time Series (R1)

We **do test** our method on multivariate time series (lines 253-265) using the recently released UEA archive, that is aimed at being used as a reference multivariate archive similarly to the UCR one. To this end, the encoder network is only modified by setting the number of input channels of its first convolution to the number of dimensions of the time series in the dataset. We will clarify this point.

4 Hyperparameters (R2)

We provide all hyperparameters values in the supplementary material, and will release all our code. The scaling factor in the code has always been set to one and can thus be discarded in the description of the paper.

Choosing a good set of hyperparameters for an unsupervised method is often challenging since the plurality of downstream tasks are usually supervised. Thus, similarly to what was done in other unsupervised work [2], we chose to keep all hyperparameters constant for each entire dataset archive (i.e., a single set of hyperparameter values for all UCR datasets, and another set for all UEA datasets), without any specific tuning for any particular downstream task. We will clarify these points.

5 Writing (R1)

We will proofread the manuscript for language clarity.

References

[1] P. Malhotra et al. TimeNet: Pre-trained deep recurrent neural network for time series classification. *arXiv:1706.08838*, 2017.

[2] L. Wu et al. Random Warping Series: A random features method for time-series embedding. In *AISTATS*, 2018.