Authors' reply to the Discussion of 'Automatic Change-Point Detection in Time Series via Deep Learning' at the Discussion Meeting on 'Probabilistic and statistical aspects of machine learning'

Jie Li†
Department of Statistics, London School of Economics and Political Science, London, UK
E-mail: j.li196@lse.ac.uk
Paul Fearnhead
Department of Mathematics and Statistics, Lancaster University, Lancaster, UK
E-mail: p.fearnhead@lancaster.ac.uk
Piotr Fryzlewicz
Department of Statistics, London School of Economics and Political Science, London, UK
E-mail: p.fryzlewicz@lse.ac.uk
Tengyao Wang
Department of Statistics, London School of Economics and Political Science, London, UK
E-mail: t.wang59@lse.ac.uk

We would like to thank the proposer, seconder and all discussants for their time in reading our paper and their thought-provoking comments. We are glad to find a broad consensus that neural-network-based approach offers a flexible framework for automatic change-point analysis. There are a number of common themes to the comments, and we have therefore structured our response around the topics of the theory, training, the importance of standardisation and possible extensions, before addressing some of the remaining individual comments.

Theory Both Wilkinson and Zhang compare the theoretical bound on the generalisation error in Theorem 4.2 with our empirical results in Figure 2 of main text, Figures S2 and S3 of supplement. Our experience has been that the empirical generalisation error has been substantially lower than the theoretical bounds suggest — with good performance of our fitted neural network with training sample sizes that are orders of magnitude smaller than one may expect given the number of parameters within the neural network. Related to this is that how the bound on the generalisation error depends on e.g. the number of layers, is not particularly informative about how these factors affect the error in practice. We agree with both Wilkinson and Zhang that there is a need for a new theoretical framework for bounding the generalisation error of neural networks that is more meaningful in practice.

We thank Zhang for pointing out how our theoretical analysis can be extended to more general data generating mechanisms, including heavier-than-Gaussian noise distributions and data with weak temporal correlation (a concern of Hong et al. (2024)). Indeed, as Zhang comments, the same procedure still works in such settings and the current proof will go through, with minor modifications to the choice of λ in Corollary 4.1. As λ is adaptively chosen by the neural network (which is one of the main attractions of our procedure), the results in Theorems 4.2 and 4.3 will

†Addresses for correspondence: Jie Li, Department of Statistics, London School of Economics and Political Science, Columbia House, Houghton Street, London, WC2A 2AE. **Email**: j.li196@lse.ac.uk

be essentially unchanged. In a similar vein, Schmidt-Hieber mentions that our theory could be modified to prove local change-point detection error rates using convolutional neural networks (CNN). Indeed, by combining existing results on MOSUM (Eichinger and Kirch, 2018) together with VC dimension results of CNN, we could arrive at a similar result to Theorems 4.2 and 4.3 in the paper.

Gavioli-Akilagun points out that the current architecture cannot directly exactly represent the likelihood-ratio test statistic for detection of piecewise affine changes. We agree with this observation. However, by including squared observations as inputs, or if we include squaring in the set of non-linearities permitted by the neural network, we can directly express the likelihood ratio test as a function in the neural network class (VC-dimension results concerning neural networks with piecewise polynomial activation functions are given in Theorem 7 of Bartlett et al. (2019)). Moreover, by including multiple layers within our architecture, the neural network can accurately approximate the square function. This is related to the nice results of Gavioli-Akilagun that show test statistics based on linear functions, which can be represented simply by a neural network, can have statistical performance comparably to those based on the likelihood-ratio test.

Regarding Zhang's observation that the first term in Theorem 4.3 does not decrease as training sample size N increases, our intuition is that the first term represents the Bayes risk of the classification task on the test set, which is achieved by the CUSUM-based classifier. Therefore, it will not be affected by increasing training sample size and only depends on the test sample.

Neural network architecture Both Wilkinson and Nemeth raise the question of how to choose a suitable neural network architecture for different settings. This is a well-studied problem in machine learning. A few possible Neural Architecture Search approaches are mentioned in Paaß and Giesselbach (2023, Section 2.4.3) (also discussed in Section 6 of the main text). Nemeth also asks the question of whether it is easier to choose the neural network architecture than to choose stochastic models to represent the data. Given these Neural Architecture Search methods mentioned above, we believe that the former is at least a more structured problem that can be solved algorithmically.

Schmidt-Hieber discusses the effect of the depth of the network on its ability to detect various structures in the signal. Indeed, we agree that in general the less we know about the data generating mechanism, the more layers we need in the architecture.

Training Cribben and Anastasiou raise a number of important practical considerations with training. First, as our approach requires labelled data, there is the challenge of obtaining such data. We agree that for manually labelled data, the labelling of changes is subjective, and this means that our method will only aim to replicate the manual classification. If we use simulation, then we can avoid this, but at the expense of needing to model what changes would look like. They also point out that often changes are rare — so training data may be imbalanced, and we would also want to account for this imbalance when detecting new changes. If we believe the frequency of changes will be different in the training than in the test data, or if we wish to account differently for different types of errors (false detection versus missing a true change), this is possible by including different weights for each error when training.

Both Bodenham and Adams, and Hong et al., ask how our approach could be applied to a single long time-series, for example from finance. First, our method requires labelled training data, so we would require part of the time-series to have labelled changes, or known to have no changes. In this case we can divide such historical data into time-series of a fixed length, leading to a set of labelled training data. One can then fit a neural network classifier to this data, and use the fitted neural network to classify windows of new data using the same moving window idea as in Section 6.

When not enough training data is available, we proposed to simulate artificial data to train our neural network. Wilkinson points out that this has a close link to the simulation-based inference. We agree that it is worth investigating the links with this literature further.

Standardisation We agree with Chen and Chen that the simple neural network classifier is not automatically invariant to the shifting and scaling because it may not have learned an exactly invariant statistic.

Differences in the results for standardised versus non-standardised data show that the algorithm (not unexpectedly) learns to perform classification for the problem at hand, whether or not it aligns with the analyst's perceived best way of distinguishing the two classes. More specifically, the algorithm's task is to solve a binary classification problem, distinguishing between two groups of sequences. While the analyst may suspect that it is the presence of the change-point that is the main feature separating the two classes, the learning algorithm may take a different view given the training data. For example, in the non-standardised case, if all the input data starts with $\mu_{\mathrm{L}} = 0$, what the analyst regards as a change-point problem the algorithm may construe as a testing problem of departure from a zero mean.

Furthermore, as pointed out by Wilkinson, the min-max scaling used in our algorithm may not be appropriate for very heavy-tailed data. In such contexts, scaling by empirical quantiles other than 0 and 1 would be more appropriate.

In many applications, one would also like the test to be invariant to the reversal of the time direction. Hence, Chen and Chen's proposal of adding reversed sequence $X_n, \ldots, X_1$ into the training data, which has the additional benefit of enlarging the sample size, makes sense.

Extensions As Bodenham and Adams point out, often one is interested in localising changes, rather than just detecting them. As a first work in the area of using neural networks to automatically construct change-point detectors, we deliberately focused on the problem of detection rather than localisation. However frameworks similar to that of MOSUM (Eichinger and Kirch, 2018), where we apply a detector to different windows of data, can use a change-point detector to also estimate the location of any changes. We used this idea, i.e. Algorithm 1, in the analysis of the HASC data in Section 6. However, we believe that there may be more attractive ways of extending our idea to the problem of change-point localisation.

A number of discussants (Anastasiou and Cribben, Schoenberg and Wong, Bodenham and Adams) ask whether our method can be applied in an online setting. This is possible provided that we have a pre-trained neural network that can classify data of a given window length, say h. When we receive a new observation, we can apply the neural network classifier to the most recent h time-points. Computationally this is possible in an online setting, as once trained, the cost of running the classifier is fixed. There are challenges in terms of how to tune the classifier so that the resulting online change-point detector has an appropriate average run length. Also, how to extend this idea so that we also update the classifier on-line as we get new data, is an interesting open question.

Schoenberg and Wong, Anastasiou and Cribben, Mateu, van Lieshout and Lu ask whether our ideas could be extended to multivariate data, and in particular spatial data or point process data. We are interested to hear about the recent developments in this area and look forward to seeing future works in this direction. One challenge of dealing with point process data is that the number of points is random and cannot be easily interpreted as input of a neural network. As a first order approximation, we could bin the data into a (multivariate) time-series of counts and a similar method to the one proposed in our paper could then be applied.

Other comments We are interested to read about other research at the interface between neural networks and change-point and related areas in statistics. This includes the using change-point detection to improve the fitting of deep neural network models to time-series data (Jungbluth and Lederer, 2023) and the possibility of using ideas from our paper to improve existing change-point detection methods as suggested by Ombao. Schmidt-Hieber also points out the possibility of using a convolutional neural network-based approach for detecting local change-points or two-dimensional edge detection. Finally, we agree with MacKenzie that one disadvantage with automated procedures like the one in our paper is that the final test statistic for a change is hard to interpret. We welcome work on improving interpretability of AI, and believe ideas in this area will be important as AI methods are increasingly using within statistics.

References

Bartlett, P. L., N. Harvey, C. Liaw, and A. Mehrabian (2019). Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *J. Mach. Learn. Res. 20*(63), 1–17.

Eichinger, B. and C. Kirch (2018). A MOSUM procedure for the estimation of multiple random change points. *Bernoulli 24*(1), 526–564.

Hong, Y., O. Linton, B. McCabe, J. Sun, and S. Wang (2024). Kolmogorov–Smirnov type testing for structural breaks: A new adjusted-range based self-normalization approach. *Journal of Econometrics 238*(2), 105603.

Jungbluth, A. and J. Lederer (2023). The DeepCAR method: Forecasting time-series data that have change points. *arXiv preprint arXiv:2302.11241.*

Paaß, G. and S. Giesselbach (2023). *Foundation Models for Natural Language Processing: Pre-trained Language Models Integrating Media.* Artificial Intelligence: Foundations, Theory, and Algorithms. Springer International Publishing.