A Novel Loss for Change Point Detection Models with Time-invariant Representations

Zhenxiang Cao, Nick Seeuws, Maarten De Vos, and Alexander Bertrand

***Abstract*—Change point detection (CPD) refers to the problem of detecting changes in the statistics of pseudo-stationary signals or time series. A recent trend in CPD research is to replace the traditional statistical tests with distribution-free autoencoder-based algorithms, which can automatically learn complex patterns in time series data. In particular, the so-called time-invariant representation (TIRE) models have gained traction, as these separately encode time-variant and time-invariant subfeatures, as opposed to traditional autoencoders. However, optimizing the trade-off between two loss terms, i.e., the reconstruction loss and the time-invariant loss, is challenging. To address this issue, we propose a novel loss function that elegantly combines both losses without the need for manually tuning a trade-off hyperparameter. We demonstrate that this new hyperparameter-free loss, in combination with a relatively simple convolutional neural network (CNN), consistently achieves superior or comparable performance compared to the manually-tuned baseline TIRE models across diverse benchmark datasets, both simulated and real-life. In addition, we present a representation analysis, demonstrating that the distribution of the time-invariant features extracted by our model is more concentrated within the same segment (more so than with previous TIRE models), which implies that these features can potentially be used for other applications, such as classification and clustering.**

***Index Terms*—Autoencoder, change point detection, time-invariant representation (TIRE), unsupervised learning**

I. Introduction

CHANGE point detection (CPD) is a prevalent challenge in many disciplines that deal with time series data, e.g., signal processing [1], [2], finance [3], [4], biology [5], [6], and climate science [7], [8]. The purpose of CPD is to identify the time points at which the statistical properties of a signal or time series change abruptly. These changes may be attributed to modifications in underlying physical processes, regime transitions, or the emergence of anomalies.

Advances in machine learning and deep learning, particularly autoencoders, have lead to powerful CPD methods that can handle complicated data distributions. However, existing autoencoder-based approaches such as Adaptive Change Detection (ACD) [9] and Autoencoder-based Breakpoints Detection (ABD) [10] only focused on the reconstruction ability of the autoencoder. To address this, the time-invariant representation (TIRE) [11] model introduced a time-invariant loss in addition to the reconstruction loss. This model learns time-invariant features by minimizing the distance between encoded sub-features of consecutive windows. Change points (CPs) are then extracted by detecting significant changes in these time-invariant features.

While the TIRE model is superior to other CPD methods [11], it has the disadvantage that it introduces an additional hyperparameter to control the trade-off between the reconstruction loss and the time-invariant loss. Tweaking this hyperparameter can be challenging, particularly in an unsupervised setting where no ground truth is available regarding CPs. A similar trade-off appears in the choice of the (relative) size of the time-invariant (TI) versus time-variant (TV) subfeature vectors.

In this paper, we present a novel loss function for unsupervised CPD using TIRE-based approaches to overcome the aforementioned concerns, which consistently achieves superior or comparable results across diverse simulated and real-life benchmark datasets. Our main contributions can be summarized as follows.

- We introduce a new hyperparameter-free loss function that considers simultaneously the reconstruction power of the autoencoder and the informativeness of the encoded representations.
- We also show that the new loss reduces the leakage of TV information in the TI subfeatures and vice versa, making the model less sensitive to the proper selection of the dimensions of both subfeature vectors. As a result, the extracted TV representations are more informative for CPD while maintaining clear boundaries between segments.

II. Brief Introduction of the TIRE Model

The original TIRE model [11] is composed of a simple autoencoder with one fully-connected layer in both the encoder and decoder and takes inputs in the form of 50% overlapping windows containing N time samples. In the latent space, the TIRE model disentangles each representation into two distinct features: a TI feature ($\mathbf{f}^{ti}$), designed to encapsulate information shared across consecutive time windows (assuming no Change Point exists between them), and a TV feature ($\mathbf{f}^{tv}$), meant to reflect information specific to individual windows. CPs are identified by considering only the dissimilarity between the TI features. To ensure a compact distribution of TI features within the same segment, a time-invariant loss term was introduced in addition to the reconstruction loss, resulting in the following combined loss

$$\mathcal{L} = \mathcal{L}^{rec} + \lambda \mathcal{L}^{ti} = \sum_{t} (||\mathbf{w}_t - \mathbf{w}'_t||_2^2 + \lambda (||\mathbf{f}_t^{ti} - \mathbf{f}_{t-1}^{ti}||_2^2), \quad (1)$$

This research received funding from the Flemish Government (AI Research Program) and from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No.802895).

All authors are with the STADIUS Center Department of Electrical Engineering, KU Leuven, Belgium (e-mails: zhenxiang.cao@kuleuven.be). All authors are affiliated to Leuven.AI - KU Leuven institute for AI, KU Leuven, Belgium. Maarten De Vos is also with Department of Development and Regeneration, KU Leuven, Belgium.

where $\mathbf{w}_t$ and $\mathbf{w}'_t$ denote the input time window at time t and the reconstruction result, respectively. λ represents the trade-off hyperparameter to balance the two losses.

The time-invariant loss term effectively compresses the distribution of the TI features $\mathbf{f}^{ti}$ within the same segment [1] by minimizing the distance between consecutive TI features. It also enforces the features to be close to each other between segments. This can lead to blurred boundaries and negatively impact the final detection performance. In contrast to the constraints placed on $\mathbf{f}^{ti}$, there are no explicit constraints on $\mathbf{f}^{tv}$ to ensure that they carry all window-dependent information. This presents a challenge in determining the appropriate dimension of $\mathbf{f}^{tv}$. If the length is too short, there is a risk of window-dependent information leaking into $\mathbf{f}^{ti}$ to improve the reconstruction loss. In contrast, if the length is too long, the autoencoder may encode all information into $\mathbf{f}^{tv}$, causing $\mathbf{f}^{ti}$ to lose its intended function. These limitations of the TIRE model will be visualized and further discussed in Section IV-E.

In summary, the TIRE loss function is difficult to tune, both in terms of the hyperparameter λ, as well as the dimension of its two feature vectors $\mathbf{f}^{ti}$ and $\mathbf{f}^{tv}$ (in particular with respect to their relative sizes).

III. Proposed Method

We adopt the same pre- and post-processing strategies as in [11] to split the input time series into windows and identify CPs based on the extracted TI features. The window size N is user-defined and specifies the time-resolution with which change points can be detected. A rule of thumb is to set N based on the expected minimal time between consecutive change points.

A. Diamond loss

Based on the aforementioned limitations of the TIRE model, we propose a new loss function without trade-off parameter, while also removing the potential danger of TV information leaking into the TI features (and vice versa) in case the feature dimension of $\mathbf{f}^{tv}$ and $\mathbf{f}^{ti}$ are not properly tuned.

Fig.1 illustrates the concept of TI and TV features. The core idea behind the new loss function is the following. If $\mathbf{f}^{ti}$ indeed encodes TI information, then it should be possible to reconstruct the window $\mathbf{w}_t$ at time t using the TI feature $\mathbf{f}^{ti}_{t-1}$ of the *previous* window at time $t-1$, in combination with its own TV features $\mathbf{f}^{tv}_t$. Based on this, feeding the combination of $\mathbf{f}^{ti}_t$ and $\mathbf{f}^{tv}_{t-1}$ to the decoder should produce a reconstruction of time window $\mathbf{w}_{t-1}$ as the output, denoted as $\mathbf{w}''_{t-1}$. Similarly, we can obtain $\mathbf{w}''_t$ by combining $\mathbf{f}^{ti}_{t-1}$ and $\mathbf{f}^{tv}_t$.

In summary, we begin by mapping pairs of consecutive windows, denoted as $(\mathbf{w}_{t-1}, \mathbf{w}_t)$, to the latent feature space using the encoder. This latent feature space comprises both the TI and TV feature spaces. Subsequently, we recombine the encoded TI and TV features before passing them through the decoder. The decoder then produces the reconstructed results, denoted as $(\mathbf{w}''_{t-1}, \mathbf{w}''_t)$. Finally, we define the new loss function as:

$$\mathcal{L}^{dia} = \sum_t (||\mathbf{w}_t - \mathbf{w}''_t||_2^2 + ||\mathbf{w}_{t-1} - \mathbf{w}''_{t-1}||_2^2). \quad (2)$$

We refer to this newly proposed loss as the diamond loss due to the diamond-shape of the diagram in Fig.1 relating to this new loss.

While the idea is simple, the impact of this new diamond loss is significant, as it offers several advantages over the loss function in the TIRE model [11], which will be illustrated in Section IV: 1) The diamond loss can handle both optimization targets, $\mathcal{L}^{rec}$ and $\mathcal{L}^{ti}$, simultaneously, thereby avoiding the need to balance the weights between the two targets. 2) The diamond loss provides a less stringent regularization of the TI features, which can theoretically lead to more explicit segment boundaries. 3) The diamond loss and the way the TI/TV features are used in the construction (see Fig.1) implicitly adds additional constraints to both types of features. The TI-features are *only* used across the window boundaries, whereas the TV features are *only* used within their respective windows. This reduces the risk of leakage of TI/TV information into the TV/TI features, respectively.

B. CNN-based TIRE autoencoder

Another potential disadvantage of the TIRE model [11] is that it employs a multilayer perceptron (MLP) network with numerous trainable parameters to build the autoencoder. As the model is trained in an unsupervised and transductive setting, and the length of training time series data is often limited, the TIRE model's performance can be sensitive to parameter initialization. To overcome this limitation, we introduce a new CNN-based autoencoder that is more robust to initialization. We detail the structure of this CNN-based autoencoder in Section IV-C.

IV. Experiment

A. Benchmark datasets

We evaluate the detection performance of our proposed model and other baseline methods using four simulated datasets and three real-life datasets.

1) Simulated datasets: The simulated datasets are generated using the following auto-regressive model [11]: $s(t) = a_1 s(t-1) + a_2 s(t-2) + \epsilon_t$, where the error term follows a Gaussian distribution $\epsilon_t \sim \mathcal{N}(\mu_t, \sigma_t^2)$, and where $s(1) = s(2) = 0$, $a_1 = 0.6$, $a_2 = -0.5$, $\mu_t = 0$, and $\sigma_t = 1.5$, unless otherwise stated.

- **Jumping Mean (JM)**: The Jumping Mean dataset is constructed by shifting the μ_t at each CP.
- **Scaling Variance (SV)**: The Scaling Variance dataset is generated by altering the σ_t at each CP.
- **Changing Coefficients (CC)**: While keeping $a_2 = 0$, a_1 in this dataset is alternatively sampled from two independent uniform distributions if a CP is crossed.
- **Gaussian Mixture (GM)**: Unlike the previous three simulated datasets, the Gaussian Mixture dataset is created by alternatively sampling from two separate Gaussian mixtures.

[1] A *segment* is defined as the time samples between two consecutive CPs.

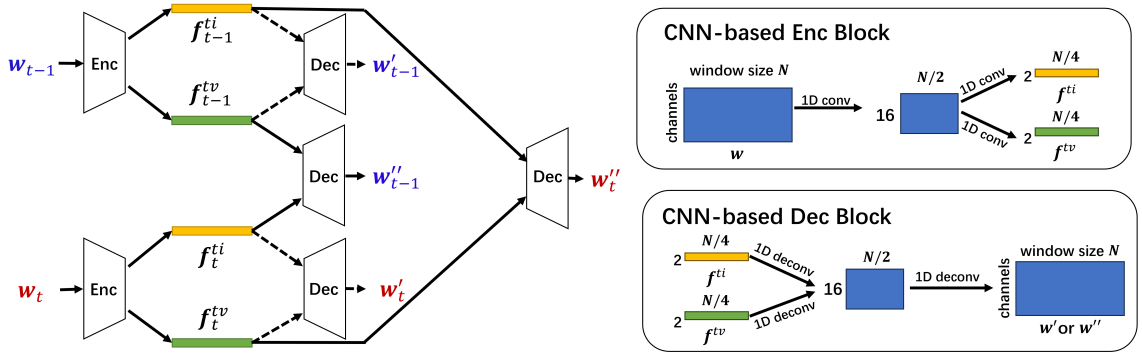

Fig. 1: The pipeline to extract TI and TV features from input window pairs and to compute the proposed diamond loss. In the left subfigure, the dashed and solid lines represent the data flows in the original TIRE loss and the diamond loss, respectively. In the right subfigure, we demonstrate the proposed CNN-based TIRE encoder and decoder structures.

2) Real-life datasets:

• **HASC-2011 [12]**: This dataset comprises human activity information collected by a three-axis accelerometer. It includes different segments that correspond to six behaviors: staying, walking, jogging, skipping, walking downstairs, and climbing stairs. Following [11], we select the data from subject 671 and use the magnitude of the acceleration as input.

• **Well log [13]**: CPs in the Well log dataset reflect transitions in the properties of the rock layers encountered during the drilling process.

• **Honeybee Dance [14]**: The Honeybee Dance dataset comprises six sequences, each featuring a bee performing a three-stage waggle dance. Each sequence is a three-dimensional time series representing the bee's position in 2D coordinates, along with its angle differences.

B. Evaluation metrics and baseline models

We use the criteria presented in [11] in our experiment to determine if a detected alarm is a real positive CP. Similar to [15], [16], [17], the f1-score metric is employed to compare the detection effectiveness of our suggested model to that of other baseline CPD techniques. The f1-score metric is defined as: f1-score $= 2 \cdot \text{precision} \cdot \text{recall}/(\text{precision} + \text{recall})$ with precision $= N_{TP}/(N_{TP} + N_{FP})$ and recall $= N_{TP}/(N_{TP} + N_{FN})$, where $N_{(\cdot)}$ denotes the number of samples, and TP, FP, and FN represent true-positive, false-positive, and false-negative detections, respectively. The rules to define TPs, FPs and FNs are the same as in [11], which also requires the definition of the tolerated maximal distance τ between a detected alarm and a corresponding ground-truth CP.

To demonstrate the benefits of the diamond loss and the proposed CNN structure, we compare our suggested model against the original TIRE model [11], and the combination of the TIRE network and the diamond loss.

C. Experiment settings

In our implementation, the window size N is kept the same for all baseline algorithms but selected separately for each dataset, such that for each dataset the median f1-score is maximal when taking the median across all (versions of) algorithms in the comparison. Specifically, we set $N = 40$ for all simulated datasets and $N = 280$, $N = 100$, and $N = 16$ for the HASC-2011, Well Log, and Honeybee Dance datasets, respectively. Furthermore, we set the tolerance value τ used to judge whether a detected alarm is a true-positive sample equal to the window size N in each dataset.

For the MLP networks (corresponding to the original TIRE model in [11]), we set the length of $\mathbf{f}^{ti}$ to 2 and the length of $\mathbf{f}^{tv}$ to 1, as suggested in [11]. The CNN-based autoencoder consists of four layers, where the input time windows are first mapped to a 16-dimensional intermediate space. Two sub-layers then map the intermediate features into TI and TV spaces, respectively, each with an output size of $2 \cdot N/4$. The TI and TV features from consecutive windows are recombined as shown in Fig.1 and transformed back to 16 dimensions in a deconvolutional layer, before the output layer ensures that the reconstructed results have the same shape as the input windows. We adopt the *Tanh* activation function in the feature-extracting and output layers to limit the value range of features and reconstructed windows. The *Leaky ReLU* function is utilized in other layers. We set the kernel size as 9 and the stride as 2 for all layers in the CNN structure. All models are optimized using an Adam optimizer with a fixed learning rate of 0.001. To reduce the effects of randomness caused by initialization and shuffling of input mini-batches, we report the mean and standard deviation across 10 different runs. The full implementation can be found in [18].

D. Results

Similar to [11], we conduct an evaluation of all models in three distinct settings. The first setting involves detecting CPs exclusively in the time domain (TD), while the second setting involves detecting CPs solely in the frequency domain (FD). Finally, we evaluate the models by combining information from both domains to detect CPs ('Both'). The results are summarized in TABLE I.

As illustrated in TABLE I, our proposed model (CNN+diamond loss) demonstrates significant improvements across the majority of our evaluation datasets in comparison to the original TIRE baseline model of [11] (MLP+original

TABLE I: The f1-scores and standard deviations across realizations and repetitions achieved by different combinations of network structures and loss functions. For each dataset, we highlight the best-performing baseline in bold.

Model	Domain	Simulated datasets				Real-life datasets		
		JM	SV	CC	GM	HASC-2011	Well log	Honeybee dance
MLP+original loss	TD	**0.957±0.021**	0.751±0.074	0.693±0.079	0.765±0.108	0.489±0.042	0.471±0.042	0.657±0.164
	FD	0.935±0.045	0.864±0.070	0.873±0.071	**0.985±0.021**	0.488±0.024	0.419±0.042	0.748±0.122
	Both	0.952±0.067	0.862±0.067	0.821±0.074	0.977±0.022	0.488±0.019	0.406±0.093	0.740±0.149
MLP+diamond loss	TD	0.946±0.023	0.786±0.088	0.797±0.075	0.782±0.104	0.497±0.036	0.467±0.052	0.672±0.121
	FD	0.934±0.045	0.932±0.036	0.911±0.073	0.981±0.021	0.540±0.017	0.481±0.053	0.752±0.117
	Both	0.945±0.026	0.930±0.036	0.899±0.067	0.977±0.022	0.554±0.018	0.488±0.058	0.751±0.103
CNN+diamond loss	TD	0.946±0.017	0.736±0.046	0.646±0.042	0.966±0.020	0.505±0.013	0.504±0.031	0.713±0.064
	FD	0.952±0.021	**0.942±0.021**	**0.966±0.022**	0.972±0.028	0.558±0.012	0.506±0.030	**0.776±0.092**
	Both	0.946±0.018	0.937±0.022	0.954±0.023	0.975±0.018	**0.571±0.011**	**0.507±0.048**	0.768±0.098

loss). The only exceptions are the Jumping mean and Gaussian Mixture datasets, where detecting CPs is not a particularly challenging task, and therefore all models under comparison exhibit similar efficacy.

Furthermore, even when using an MLP network to construct the autoencoder, the diamond loss still leads to substantially higher f1-scores on the other two more difficult simulated datasets and all real-life datasets when compared to the original TIRE loss. This observation verifies the superiority of the diamond loss, which aligns with our original design purpose.

Introducing the CNN-based autoencoder leads to further improvement in the detection accuracy and smaller deviations, implying that the CNN model is more robust to the initialization of parameters. While the transition from the MLP to the CNN structure may result in a drop in f1-score in the time domain, the performance remains superior or comparable in the **Both** setting, which is the default setting if no domain knowledge is available on whether CPs are most pronounced in the time or frequency domain [11].

E. Representation Analysis

To ensure that the diamond loss can achieve our design purpose, we investigate the extracted TI features further.

In order to provide a visual representation of the performance of our proposed framework to extract TI features, we compute the distance matrices between the TI features across different time points. The entry at position (i, j) is equal to the distance between $\mathbf{f}_{t_i}^{ti}$ and $\mathbf{f}_{t_j}^{ti}$. If there is no CP between t_i and t_j, this distance should ideally be 0. The heatmaps in Fig.2 illustrate the ground truth, the TI features extracted by our CNN+diamond loss model, and the TI features extracted by the MLP+original loss model, from left to right in each row for two of the datasets. Note that the checkerboard pattern appears due to the fact that the ground truth statistics alternate between CPs, hence it is expected that the TI features are similar between segments containing the same ground truth statistics, even if they are far away in time.

The heatmaps clearly visualize the differences between the TI features extracted by our proposed framework and the MLP+original loss model. Specifically, the heatmaps obtained by the MLP+original loss model show that the contrast between dark and (neighboring) light blocks is much lower than for CNN+diamond loss model. This observation suggests that window-dependent information has leaked into the TI features. In contrast, the distance matrices of the TI features extracted by the CNN+diamond loss model show a higher contrast between neighboring blocks that are separated by a CP boundary. In addition, the clear dark blocks verify the high degree of similarity of the TI features within the same segments after discarding the instantaneous information. This result aligns with the design of the diamond loss, which explicitly encourages TV features to capture window-dependent information while minimizing the leakage of TV information into the TI features.

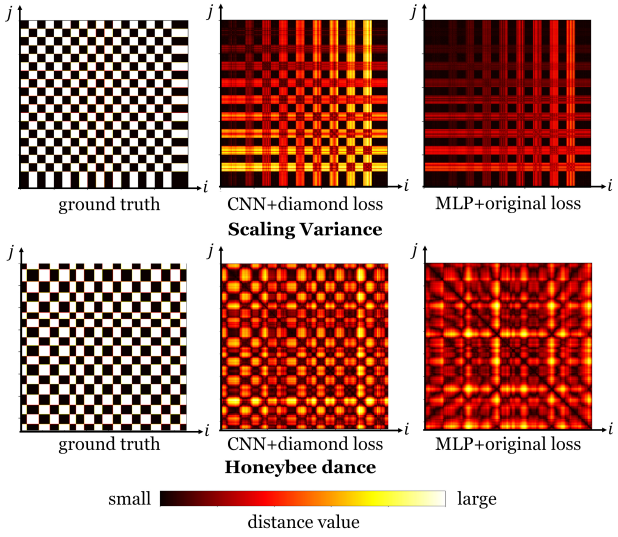

Fig. 2: Visualization of distance matrices of TI features in the form of heatmaps. The entry at position (i, j) represents the distance between TI features $\mathbf{f}_{t_i}^{ti}$ and $\mathbf{f}_{t_j}^{ti}$ at time steps t_i and t_j. The locations of real CPs correspond to the boundaries of the dark blocks along the diagonal in the ground truth sub-figures.

V. Conclusion

We have proposed the diamond loss to achieve better CPD accuracy by incorporating the TI and TV constraints within the reconstruction loss itself, rather than adding it as a separate penalty loss. Compared to existing approaches, the new loss function reduces the manual effort to tune hyperparameters and achieves superior detection performance on both simulated and real-life datasets.

REFERENCES

[1] A. Owrang, M. Malek-Mohammadi, A. Proutiere, and M. Jansson, "Consistent change point detection for piecewise constant signals with normalized fused lasso," *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 799–803, 2017.

[2] Y.-W. Liu and H. Chen, "A fast and efficient change-point detection framework based on approximate k-nearest neighbor graphs," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1976–1986, 2022.

[3] J. Bai and P. Perron, "Estimating and testing linear models with multiple structural changes," *Econometrica*, pp. 47–78, 1998.

[4] M. Lavielle and G. Teyssiere, "Adaptive detection of multiple change-points in asset price volatility," *Long memory in economics*, pp. 129–156, 2007.

[5] M. Bosc, F. Heitz, J.-P. Armspach, I. Namer, D. Gounot, and L. Rumbach, "Automatic change detection in multimodal serial mri: application to multiple sclerosis lesion evolution," *NeuroImage*, vol. 20, no. 2, pp. 643–656, 2003.

[6] R. Malladi, G. P. Kalamangalam, and B. Aazhang, "Online bayesian change point detection algorithms for segmentation of epileptic activity," in *2013 Asilomar Conference on Signals, Systems and Computers*. IEEE, 2013, pp. 1833–1837.

[7] J.-F. Ducré-Robitaille, L. A. Vincent, and G. Boulet, "Comparison of techniques for detection of discontinuities in temperature series," *International Journal of Climatology: A Journal of the Royal Meteorological Society*, vol. 23, no. 9, pp. 1087–1101, 2003.

[8] J. Reeves, J. Chen, X. L. Wang, R. Lund, and Q. Q. Lu, "A review and comparison of changepoint detection techniques for climate data," *Journal of applied meteorology and climatology*, vol. 46, no. 6, pp. 900–915, 2007.

[9] M. Gupta, R. Wadhvani, and A. Rasool, "Real-time change-point detection: A deep neural network-based adaptive approach for detecting changes in multivariate time series data," *Expert Systems with Applications*, vol. 209, p. 118260, 2022.

[10] W.-H. Lee, J. Ortiz, B. Ko, and R. Lee, "Time series segmentation through automatic feature learning," 2018. [Online]. Available: https://arxiv.org/abs/1801.05394

[11] T. De Ryck, M. De Vos, and A. Bertrand, "Change point detection in time series data using autoencoders with a time-invariant representation," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3513–3524, 2021.

[12] N. Kawaguchi, Y. Yang, T. Yang, N. Ogawa, Y. Iwasaki, K. Kaji, T. Terada, K. Murao, S. Inoue, Y. Kawahara *et al.*, "Hasc2011corpus: towards the common ground of human activity recognition," in *Proceedings of the 13th international conference on Ubiquitous computing*, 2011, pp. 571–572.

[13] J. O. Ruanaidh, W. J. Fitzgerald, and K. J. Pope, "Recursive bayesian location of a discontinuity in time series," in *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4. IEEE, 1994, pp. IV–513.

[14] S. Oh, J. Rehg, T. Balch, and F. Dellaert, "Learning and inferring motion patterns using parametric segmental switching linear dynamic systems," *International Journal of Computer Vision*, vol. 77, 05 2008.

[15] A. De Brabandere, Z. Cao, M. De Vos, A. Bertrand, and J. Davis, "Semi-supervised change point detection using active learning," in *Discovery Science: 25th International Conference, DS 2022, Montpellier, France, October 10–12, 2022, Proceedings*. Springer, 2022, pp. 74–88.

[16] S. Deldari, D. V. Smith, H. Xue, and F. D. Salim, "Time series change point detection with self-supervised contrastive predictive coding," in *Proceedings of the Web Conference 2021*, 2021, pp. 3124–3135.

[17] G. J. J. v. d. Burg and C. K. I. Williams, "An evaluation of change point detection algorithms," 2020. [Online]. Available: https://arxiv.org/abs/2003.06222

[18] "GitHub repository for diamond loss," https://github.com/caozhenxiang/diamond-loss.