Kernel Change-point Detection with Auxiliary Deep Generative Models

Wei-Cheng Chang **Chun-Liang Li** **Yiming Yang** **Barnabás Póczos**
Carnegie Mellon University
{wchang2,chunlial,bapoczos,yiming}@cs.cmu.edu

Abstract

Detecting the emergence of abrupt property changes in time series is a challenging problem. Kernel two-sample test has been studied for this task making fewer assumptions on the distributions than traditional parametric approaches. However, selecting kernels is non-trivial in practice. Although kernel selection for two-sample test has been studied, the insufficient samples in change point detection (CPD) problem hinders the success of those developed kernel selection algorithms. In this paper, we propose **KL-CPD**, a novel kernel learning framework that optimizes a lower bound of test power via an auxiliary generative model. With deep kernel parameterization, **KL-CPD** endows kernel two-sample test with the data-driven kernel to detect different types of change-points in real-world applications. The proposed approach significantly outperformed other state-of-the-art methods in our comparative evaluation of benchmark datasets.

1 Introduction

Detecting changes in the temporal evolution of a system in time series analysis has attracted considerable attention in machine learning decades [4, 7]. In this work, we study the retrospective change-point detection (CPD) problem [29, 21], which allows a flexible time window to react on the change-points. Retrospective CPD not only enjoys robust detection [9] but embraces many real-world applications [26, 31, 36]. Albeit being developed for many years [16], many works are parametric with strong assumptions on the distributions [4, 15], including auto-regressive models [35] and state-space models [18] for tracking changes in various statistics.

On the other hand, kernel two-sample test has been applied to time series CPD that makes fewer assumptions on the distributions (e.g. [17, 21]). The performance of kernel methods, nevertheless, relies heavily on the choice of kernels. [12, 13] conducted kernel selection for RBF kernel bandwidths via median heuristic. While certainly straightforward, it has no statistical guarantees regarding to the test power of hypothesis testing. [14] show explicitly optimizing the test power leads to better kernel choice for hypothesis testing under mild conditions. Kernel selection by optimizing the test power, however, is not directly applicable for time series CPD due to insufficient samples.

In this paper, we propose **KL-CPD**, a kernel learning framework for time series CPD, highlighting three contributions: In Section 2, we discuss the inaptness of existing kernel learning approaches in a simulated example. We then propose to optimize a lower bound of the test power via an auxiliary generative model, serving as a surrogate of the abnormal events. In Section 3, we present a deep kernel parametrization of our framework, which endows a data-driven kernel for the kernel two-sample test. **KL-CPD** induces composition kernels by combining RNNs and RBF kernels that are suitable for the time series applications. In Section 4, we conduct extensive benchmark evaluation showing the outstanding performance of **KL-CPD** in real-world CPD applications.

Workshop on Modeling and Decision-Making in the Spatiotemporal Domain, 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

2 Optimizing Test Power for Change-Point Detection

Maximum mean Discrepancy (MMD) is a nonparametric probabilistic distance commonly used in two-sample-test [12, 13]. Given a kernel k, the MMD distance between two distributions $\mathbb{P}$ and $\mathbb{Q}$ is $M_k(\mathbb{P}, \mathbb{Q}) := \mathbb{E}_{\mathbb{P}}[k(x, x')] - 2\mathbb{E}_{\mathbb{P},\mathbb{Q}}[k(x, y)] + \mathbb{E}_{\mathbb{Q}}[k(y, y')]$. In practice, with finite samples $X = \{x_1, \ldots, x_m\} \sim \mathbb{P}$ and $Y = \{y_1, \ldots, y_m\} \sim \mathbb{Q}$, we estimate $M_k(\mathbb{P}, \mathbb{Q})$ with an unbiased estimator $\hat{M}_k(X, Y) := \frac{1}{\binom{m}{2}} \sum_{i \neq i'} k(x_i, x_{i'}) - \frac{2}{m^2} \sum_{i,j} k(x_i, y_j) + \frac{1}{\binom{m}{2}} \sum_{j \neq j'} k(y_j, y_{j'})$. For any characteristic kernel k, $M_k(\mathbb{P}, \mathbb{Q}) = 0$ *iff* $\mathbb{P} = \mathbb{Q}$. However, the estimator $\hat{M}_k(X, X')$ may not be 0 even though $X, X' \sim \mathbb{P}$ due to finite sample size. Hypothesis test instead offers thorough statistical guarantees of whether two finite sample sets are the same distribution. Following [13], the hypothesis test is defined by the null hypothesis $H_0 : \mathbb{P} = \mathbb{Q}$ and alternative $H_1 : \mathbb{P} \neq \mathbb{Q}$, using test statistic $m\hat{M}_k(X, Y)$. For a given allowable false rejection probability α (i.e., Type I error), we choose a test threshold c_α and reject H_0 if $m\hat{M}_k(X, Y) > c_\alpha$.

The kernel selection objective to maximize the test power [14, 28] is presented as follows. Under the alternative $H_1 : \mathbb{P} \neq \mathbb{Q}$, $\hat{M}_k$ is asymptotically normal with $V_m(\mathbb{P}, \mathbb{Q})$ denoting the asymptotic variance. The test power is then

$$\Pr\left(m\hat{M}_k(X, Y) > c_\alpha\right) \longrightarrow \Phi\left(\frac{M_k(\mathbb{P}, \mathbb{Q})}{\sqrt{V_m(\mathbb{P}, \mathbb{Q})}} - \frac{c_\alpha}{m\sqrt{V_m(\mathbb{P}, \mathbb{Q})}}\right) \qquad (1)$$

where Φ is the CDF of the standard normal distribution. Given a set of kernels $\mathcal{K}$, We aim to choose a kernel $k \in \mathcal{K}$ to maximize the test power, which is equivalent to maximizing the argument of Φ.

In time series CPD, we denote $\mathbb{P}$ as the distribution of usual events and $\mathbb{Q}$ as the distribution for the event when change-points happen. The difficulty of choosing kernels via optimizing test power in Eq. (1) is that we have very limited samples from the abnormal distribution $\mathbb{Q}$. Kernel learning in this case may easily overfit, leading to sub-optimal performance in time series CPD.

2.1 Difficulties of Optimizing Kernels for CPD

To demonstrate how limited samples of $\mathbb{Q}$ would affect optimizing test power, we consider kernel selection for Gaussian RBF kernels on the Blobs dataset [14, 28], which is considered hard for kernel two-sample test. $\mathbb{P}$ is a 5×5 grid of two-dimensional standard normals. $\mathbb{Q}$ is laid out identically, but with covariance $\frac{\epsilon_q - 1}{\epsilon_q + 1}$ between the coordinates. Right figure shows $X \sim \mathbb{P}$ (red samples), $Y \sim \mathbb{Q}$ (blue dense samples), $\tilde{Y} \sim \mathbb{Q}$ (blue sparse samples) with $\epsilon_q = 6$. Note that when $\epsilon_q = 1$, $\mathbb{P} = \mathbb{Q}$.

For $\epsilon_q \in \{4, 6, 8, 10, 12, 14\}$, we take 10K samples for X, Y and 200 samples for $\tilde{Y}$. We choose kernels by: 1) *median heuristic*; 2) *max-ratio* $\eta_{k^*}(X, Y) = \arg\max_k \hat{M}_k(X, Y)/\sqrt{V_m(X, Y)}$; among 20 kernel bandwidths. We repeat this process 1000 times and report the test power under false rejection rate $\alpha = 0.05$. As shown in Fig. 1, optimizing kernels using limited samples $\tilde{Y}$ significantly decreases the test power compared to Y (blue curve down to the cyan curve). This result not only verifies our claim on the inaptness of existing kernel learning objectives for CPD task, but stimulates us with the following question, *How to optimize kernels with very limited samples from $\mathbb{Q}$, even none in an extreme?*

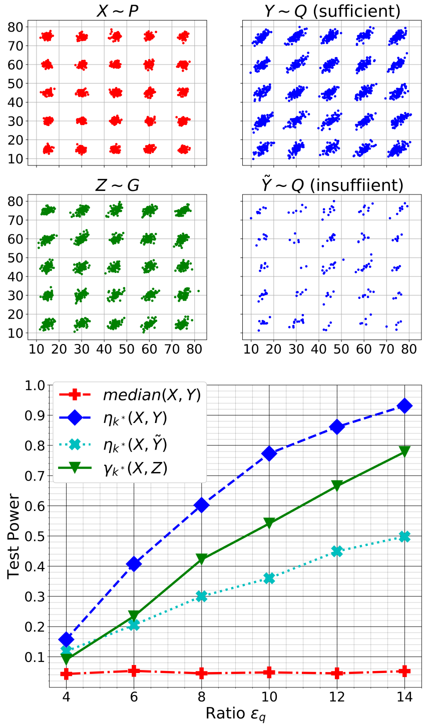

Figure 1: Test power versus ϵ_q

2.2 A Practical Lower Bound on Optimizing Test Power

We first assume there exist a surrogate distribution $\mathbb{G}$ that we can easily draw samples from ($Z \sim \mathbb{G}$, $|Z| \gg |\tilde{Y}|$), and also satisfies the following property:

$$M_k(\mathbb{P}, \mathbb{P}) < M_k(\mathbb{P}, \mathbb{G}) < M_k(\mathbb{P}, \mathbb{Q}), \forall k \in \mathcal{K}, \qquad (2)$$

Besides, we assume dealing with non trivial case of $\mathbb{P}$ and $\mathbb{Q}$ where a lower bound $\frac{1}{m}v_l \leq V_{m,k}(\mathbb{P},\mathbb{Q}), \forall k$ exists. Since $M_k(\mathbb{P},\mathbb{Q})$ is bounded, there exists an upper bound v_u. With bounded variance $\frac{v_l}{m} \leq V_{m,k}(\mathbb{P},\mathbb{Q}) \leq \frac{v_u}{m}$ condition, we derive an lower bound $\gamma_{k*}(\mathbb{P},\mathbb{G})$ of the test power

$$\max_{k\in\mathcal{K}} \frac{M_k(\mathbb{P},\mathbb{Q})}{\sqrt{V_m(\mathbb{P},\mathbb{Q})}} - \frac{c_\alpha/m}{\sqrt{V_m(\mathbb{P},\mathbb{Q})}} \geq \max_{k\in\mathcal{K}} \frac{M_k(\mathbb{P},\mathbb{Q})}{\sqrt{v_u/m}} - \frac{c_\alpha}{\sqrt{mv_l}} \geq \max_{k\in\mathcal{K}} \frac{M_k(\mathbb{P},\mathbb{G})}{\sqrt{v_u/m}} - \frac{c_\alpha}{\sqrt{mv_l}} = \gamma_{k*}(\mathbb{P},\mathbb{G}).$$

Just for now in the blob toy experiment, we artifact this distribution $\mathbb{G}$ by mimicking $\mathbb{Q}$ with the covariance $\epsilon_g = \epsilon_q - 2$. We defer the discussion on how to find $\mathbb{G}$ in the later subsection 2.3. Choosing kernels via $\gamma_{k^*}(X,Z)$ using surrogate samples $Z \sim \mathbb{G}$, as represented by the green curve in Fig. 1, substantially boosts the test power compared to $\eta_{k*}(X,\tilde{Y})$ with sparse samples $\tilde{Y} \sim \mathbb{Q}$.

Test Threshold Approximation Under $H_0 : \mathbb{P} = \mathbb{Q}$, $m\hat{M}_k(X,Y)$ converges asymptotically to an unknown distribution depending on $\mathbb{P}$ [13, Theorem 12], yielding a non closed form test threshold c_α. Even estimating c_α with permutation test or some kernel approximated distributions, it is difficult to optimize c_α because it is a function of k and $\mathbb{P}$. Alternatively, since c_α is a function of $\hat{M}_k(X,X')$ that controls the Type I error, bounding $\hat{M}_k(X,X')$ could be an approximation of bounding c_α. Therefore, we propose the following objective that maximizing a lower bound of test power

$$\operatorname*{argmax}_{k\in\mathcal{K}} M_k(\mathbb{P},\mathbb{G}) - \lambda \hat{M}_k(X,X'), \tag{3}$$

where λ is a hyper-parameter to control the trade-off between Type-I and Type-II errors. Note that the optimization of Eq. (3) is solved using the unbiased estimator of $M_k(\mathbb{P},\mathbb{G})$ with empirical samples.

2.3 Surrogate Distributions using Generative Models

The remaining question is how to construct the surrogate distribution $\mathbb{G}$. As no prior knowledge nor empirical samples of $\mathbb{Q}$, to ensure (2) holds for any possible $\mathbb{Q}$ (e.g. $\mathbb{Q} \neq \mathbb{P}$ but $\mathbb{Q} \approx \mathbb{P}$), intuitively, we have to make $\mathbb{G}$ as closed to $\mathbb{P}$ as possible. We propose to learn an *auxiliary generative model* $\mathbb{G}_\theta$ parameterized by θ such that $\hat{M}_k(X,X') < \min_\theta M_k(\mathbb{P},\mathbb{G}_\theta) < M_k(\mathbb{P},\mathbb{Q}), \forall k \in \mathcal{K}$. To ensure the first inequality hold, we set early stopping criterion when solving $\mathbb{G}_\theta$ in practice. Moreover, the limited capacity of $\mathbb{G}_\theta$ (e.g. small neural networks) [3] and finite samples of $\mathbb{P}$ hinder us to fully recover $\mathbb{P}$. Thus, we result in a min-max formulation to consider all possible $k \in \mathcal{K}$ when we learn $\mathbb{G}$,

$$\min_\theta \max_{k\in\mathcal{K}} \quad M_k(\mathbb{P},\mathbb{G}_\theta) - \lambda \hat{M}_k(X,X'), \tag{4}$$

and solve the kernel for the hypothesis test in the mean time. Lastly, we remark that although the resulted objective (4) is similar to [20], *the motivation and explanation are different.* One major difference is we aim to find k with highest test power while their goal is finding $\mathbb{G}_\theta$ to approximate $\mathbb{P}$.

3 KLCPD: Realization for Time Series Applications

To have a more expressive kernel for complex time series, we consider compositional kernels $K = \left\{\tilde{k} \mid \tilde{k}(x,x') = \exp(-\|f_\phi(x) - f_\phi(x)'\|^2)\right\}$. The resulted kernel $\tilde{k}$ is still characteristic if f is an injective function and k is characteristic [13]. Inspired by the recent success of combining deep neural networks into kernels [32, 1, 20], we parameterize f_ϕ by RNNs to capture the temporal dynamics of time series. For an injective function f, there exists a function F such that $F(f(x)) = x, \forall x \in \mathcal{X}$. A practical realization of f is a RNN encoder parametrized by ϕ while the function F is a RNN decoder parametrized by ψ trained to minimize the reconstruction loss. Thus, our final objective is

$$\min_\theta \max_\phi \quad M_{f_\phi}\big(\mathbb{P},\mathbb{G}_\theta\big) - \lambda \cdot \hat{M}_{f_\phi}\big(X,X'\big) - \beta \cdot \mathbb{E}_{\nu\in\mathbb{P}\cup\mathbb{G}_\theta}\|\nu - F_\psi\big(f_\phi(\nu)\big)\|_2^2. \tag{5}$$

Practical Implementation We consider two consecutive windows in mini-batch to estimate $\hat{M}_{f_\phi}\big(X,X'\big)$ in an online fashion for efficiency. The sample $X \sim \mathbb{P}$ is divided into the left window segment $X^{(l)} = \{x_{t-w},\dots,x_{t-1}\}$ and the right window segment $X^{(r)} = \{x_t,\dots,x_{t+w-1}\}$ such that $X = \{X^{(l)},X^{(r)}\}$. We present an realization of **KL-CPD** in Algorithm 1 with the weight-clipping technique, where the generator g_θ is also a Seq2Seq model aims at conditional generation. The stopping condition is based on a maximum number of epochs or the detecting power of $M_{f_\phi}\big(\mathbb{P},\mathbb{G}_\theta\big) \leq \epsilon$. This ensure the surrogate $\mathbb{G}_\theta$ is not too close to $\mathbb{P}$, as motivated in Sec. 2.2.

Algorithm 1: KL-CPD, our proposed algorithm.

```
input :learning rate $\alpha$, clipping range $c$, window size $w$ , $n_c$ kernel learning update per iter
while $M_{k \circ f_\phi}(\mathbb{P}, \mathbb{G}_\theta) > \epsilon$ do
    for $t = 1, \dots, n_c$ do
        Sample a minibatch $X_t \sim \mathbb{P}$, denote $X_t = \{X_t^{(l)}, X_t^{(r)}\}$, and $\omega \sim \mathbb{P}(\Omega)$
        gradient($\phi$) $\leftarrow \nabla_\phi M_{k \circ f_\phi}(\mathbb{P}, \mathbb{G}_\theta) - \lambda \hat{M}_{k \circ f_\phi}(X_t^{(l)}, X_t^{(r)}) - \beta \mathbb{E}_{\nu \sim \mathbb{P} \cup \mathbb{G}_\theta} \|\nu - F_\psi(f_\phi(\nu))\|_2^2$
        $\phi \leftarrow \phi + \alpha \cdot$ RMSProp($\phi$, gradient($\phi$))
        $\phi \leftarrow$ clip($\phi, -c, c$)
    Sample a minibatch $X_{t'} \sim \mathbb{P}$, denote $X_{t'} = \{X_{t'}^{(l)}, X_{t'}^{(r)}\}$, and $\omega \sim \mathbb{P}(\Omega)$
    gradient($\theta$) $\leftarrow \nabla_\theta M_{k \circ f_\phi}(\mathbb{P}, \mathbb{G}_\theta)$
    $\theta \leftarrow \theta - \alpha \cdot$ Adam($\theta$, gradient($\theta$))
```

4 Experiment Results

We compare the proposed **KL-CPD** with seven representative baselines on benchmark datasets from real-world applications of CPD, including **Bee-Dance** [30], **Fishkiller** [30], **HASC** [23] and **Yahoo** [34]. Detailed data description are available in Appendix B.1. Following [19, 27, 23], the datasets are split into training/validation/test set by $60\%, 20\%, 20\%$ ratio in chronological order. Note that training is fully unsupervised for all methods while labels in the validation set are used for hyperparameters tuning. We consider AUC under the ROC curves as the evaluation metric, which is commonly used in CPD literature [21, 23, 33].

We compare **KL-CPD** with real-time CPD methods (**ARMA**, **ARGP**, **RNN**,**LSTNet**) and retrospective CPD methods (**ARGP-BOCPD**, **RDR-KCPD**, **Mstats-KCPD**). Details are in Appendix B.2. Note that **OPT-MMD** is a deep kernel learning baseline which optimizes MMD by treating past samples as $\mathbb{P}$ and the current window as $\mathbb{Q}$ (insufficient samples).

Method	**Bee-Dance**	**Fishkiller**	**HASC**	**Yahoo**
ARMA [6]	0.5368	0.8794	0.5863	0.8615
ARGP [8]	0.5833	0.8813	0.6448	**0.9318**
RNN [10]	0.5827	0.8872	0.6128	0.8508
LSTNet [19]	0.6168	0.9127	0.5077	0.8863
ARGP-BOCPD [27]	0.5089	0.8333	0.6421	0.9130
RDR-KCPD [23]	0.5197	0.4942	0.4217	0.6029
Mstats-KCPD [21]	0.5616	0.6392	0.5199	0.6961
OPT-MMD	0.5262	0.7517	0.6176	0.8193
KL-CPD (Proposed method)	**0.6767**	**0.9596**	**0.6490**	0.9146

Table 1: AUC on four real-world datasets. **KL-CPD** has the best AUC on three out of four datasets.

KL-CPD shows significant gain over the other methods mostly, except being in a second place on the **Yahoo** dataset, with 2% lower AUC compared to the leading **ARGP**. This confirms the importance of data-driven kernel selection and effectiveness of our kernel learning framework. Notice that **OPT-MMD** performs not so good compared to **KL-CPD**, which again verifies our simulated example in Sec. 2 that directly applying existing kernel learning approaches with insufficient samples may not be suitable for real-world CPD task.

5 Conclusion

We propose **KL-CPD**, a new kernel learning framework for two-sample test by optimizing a lower bound of test power with a auxiliary generator, to resolve the issue of insufficient samples in change-points detection. The deep kernel parametrization of **KL-CPD** combines RNNs with RBF kernels that effectively detect a variety of change-points from different real-world applications. Extensive evaluation of our new approach along with strong baseline methods on benchmark datasets shows the outstanding performance of the proposed method in retrospective CPD.

References

[1] Maruan Al-Shedivat, Andrew Gordon Wilson, Yunus Saatchi, Zhiting Hu, and Eric P Xing. Learning scalable deep kernels with recurrent structure. *JMLR*, 2017.

[2] Michael Arbel, Dougal J Sutherland, Mikołaj Bińkowski, and Arthur Gretton. On gradient regularizers for mmd gans. In *NIPS*, 2018.

[3] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*, 2017.

[4] Michèle Basseville, Igor V Nikiforov, et al. *Detection of abrupt changes: theory and application*. Prentice Hall Englewood Cliffs, 1993.

[5] Mikołaj Bińkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *ICLR*, 2018.

[6] George Box. Box and jenkins: time series analysis, forecasting and control. *A Very British Affair, ser. Palgrave Advanced Texts in Econometrics. Palgrave Macmillan UK*, 2013.

[7] E Brodsky and Boris S Darkhovsky. *Nonparametric methods in change point problems*. Springer Science & Business Media, 2013.

[8] Joaquin Quinonero Candela, Agathe Girard, Jan Larsen, and Carl Edward Rasmussen. Propagation of uncertainty in bayesian kernel models-application to multiple-step ahead forecasting. In *ICASSP*. IEEE, 2003.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 2009.

[10] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014. URL `http://aclweb.org/anthology/D/D14/D14-1179.pdf`.

[11] Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *UAI*, 2015.

[12] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *NIPS*, 2007.

[13] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *JMLR*, 2012.

[14] Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *NIPS*, 2012.

[15] Fredrik Gustafsson. The marginalized likelihood ratio test for detecting abrupt changes. *IEEE Transactions on automatic control*, 1996.

[16] Fredrik Gustafsson and Fredrik Gustafsson. *Adaptive filtering and change detection*. Citeseer, 2000.

[17] Zaid Harchaoui, Eric Moulines, and Francis R Bach. Kernel change-point analysis. In *NIPS*, 2009.

[18] Yoshinobu Kawahara, Takehisa Yairi, and Kazuo Machida. Change-point detection in time-series data based on subspace identification. In *ICDM*. IEEE, 2007.

[19] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *SIGIR*, 2018.

[20] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *NIPS*, 2017.

[21] Shuang Li, Yao Xie, Hanjun Dai, and Le Song. M-statistic for kernel change-point detection. In *NIPS*, 2015.

[22] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *ICML*, pages 1718–1727, 2015.

[23] Song Liu, Makoto Yamada, Nigel Collier, and Masashi Sugiyama. Change-point detection in time-series data by relative density-ratio estimation. *Neural Networks*, 2013.

[24] Youssef Mroueh, Chun-Liang Li, Tom Sercu, Anant Raj, and Yu Cheng. Sobolev gan. In *ICLR*, 2018.

[25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[26] Jaxk Reeves, Jien Chen, Xiaolan L Wang, Robert Lund, and Qi Qi Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, 2007.

[27] Yunus Saatçi, Ryan Turner, and Carl Edward Rasmussen. Gaussian process change point models. In *ICML*, June 2010.

[28] Dougal J Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alex Smola, and Arthur Gretton. Generative models and model criticism via optimized maximum mean discrepancy. In *ICLR*, 2017.

[29] Jun-ichi Takeuchi and Kenji Yamanishi. A unifying framework for detecting outliers and change points from time series. *IEEE transactions on Knowledge and Data Engineering*, 2006.

[30] Ryan Turner. *Gaussian Processes for State Space Models and Change Point Detection*. PhD thesis, University of Cambridge, Cambridge, UK, July 2011.

[31] Yao Wang, Chunguo Wu, Zhaohua Ji, Binghong Wang, and Yanchun Liang. Non-parametric change-point method for differential gene expression detection. *PloS one*, 2011.

[32] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *AISTATS*, 2016.

[33] Zhao Xu, Kristian Kersting, and Lorenzo von Ritter. Stochastic online anomaly analysis for streaming time series. In *IJCAI*, 2017.

[34] Yahoo. Yahoo anomaly dataset. `https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70`, 2018. Accessed: 2018-10-20.

[35] Kenji Yamanishi and Jun-ichi Takeuchi. A unifying framework for detecting outliers and change points from non-stationary time series data. In *SIGKDD*. ACM, 2002.

[36] Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams, and Peter Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 2004.