Continuous Meta-Learning without Tasks

James Harrison, Apoorva Sharma, Chelsea Finn, Marco Pavone
{jharrison, apoorva, cbfinn, pavone}@stanford.edu
Stanford University, Stanford, CA

1 Introduction

Meta-learning methods have recently shown promise as an effective strategy for enabling efficient few-shot learning in complex domains from image classification to nonlinear regression (Finn et al., 2017; Snell et al., 2017). These methods leverage an offline "meta-training" phase, in which they use data from a distribution of tasks to optimize learning performance on new tasks. These algorithms have focused on settings with *task segmentation*, where the learning agent knows when tasks change. At meta-train time, these algorithms assume access to a meta-dataset of datasets from individual tasks, and at meta-test time, the learner is evaluated on a single task. However, there are many applications where task segmentation is unavailable, which have thus far been under-addressed in the meta-learning literature.

In this work, we present MOCA, an approach to enable meta-learning in task-unsegmented settings. MOCA operates directly on time series in which the latent task undergoes discrete, unobserved switches, rather than requiring a pre-segmented meta-dataset. MOCA integrate a Bayesian changepoint estimation scheme with existing meta-learning approaches, allowing the algorithm to reason about whether or not the task has changed in a time series. Thus, we enable a standard meta-learning algorithm, which is designed for the task segmented setting, to be both trained and tested directly on time series data without the need for task segmentation.

Problem statement. Our goal is to apply meta-learning tools to the problem of task-unsegmented continual learning, in which an agent is presented sequentially with input $\boldsymbol{x}_t$, asked to make a (probabilistic) prediction $p(\hat{\boldsymbol{y}}_t \mid \boldsymbol{x}_t)$, and is then given the true label $\boldsymbol{y}_t$, and can thus ideally improve its predictions by learning from the labeled examples. Following the terminology of meta-learning, we assume that these data are drawn from a distribution according to some latent *task* $\mathcal{T}_t$, $p(\boldsymbol{x}_t, \boldsymbol{y}_t \mid \mathcal{T}_t) = p(\boldsymbol{x}_t \mid \mathcal{T}_t)p(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \mathcal{T}_t)$. We will write $\boldsymbol{x}, \boldsymbol{y} \sim \mathcal{T}_t$ as shorthand for $\boldsymbol{x}, \boldsymbol{y} \sim p(\boldsymbol{x}, \boldsymbol{y} \mid \mathcal{T}_t)$. We assume a distribution over tasks, which we write $p(\mathcal{T})$, and that the initial task $\mathcal{T}_1 \sim p(\mathcal{T})$. At each timestep, the task is either re-sampled from $p(\mathcal{T})$ with some probability λ (which we refer to as the hazard rate), or remains the same.

Our goal is to optimize a learning agent to perform well in this setting. Let $p_{\boldsymbol{\theta}}(\hat{\boldsymbol{y}}_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1})$ by the agent's prediction for $\boldsymbol{y}_t$ given input $\boldsymbol{x}_t$ and the past labeled examples. We will evaluate the learner's performance through a negative log likelihood loss, and our objective is as follows:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \quad \mathbb{E}\left[\sum_{t=1}^{\infty} -\log p_{\boldsymbol{\theta}}(\boldsymbol{y}_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1})\right] \\ &\text{subject to} \quad \boldsymbol{x}_t, \boldsymbol{y}_t \sim \mathcal{T}_t, \quad \mathcal{T}_t = \begin{cases} \mathcal{T}_{t-1} & \text{w.p. } 1-\lambda \\ \mathcal{T}_{t,\text{new}} & \text{w.p. } \lambda \end{cases} \quad \mathcal{T}_1 \sim p(\mathcal{T}), \quad \mathcal{T}_{t,\text{new}} \sim p(\mathcal{T}) \end{aligned} \tag{1}$$

We assume that we have access to a representative time series generated in the same manner from the same distribution of tasks, and use this time series to optimize $\boldsymbol{\theta}$ in an offline, meta-training phase.

2 MOCA: Meta-Learning via Online Changepoint Analysis

MOCA uses Bayesian changepoint detection to enable the application of meta-learning algorithms to settings without task segmentation, both at train and test time. We extend Bayesian online changepoint detection (BOCPD) framework (Adams & MacKay, 2007) to derive a recursive Bayesian filtering algorithm for run length in the conditional and joint density estimation setting, and leverage a base meta-learning algorithm with parameters $\boldsymbol{\theta}$ to provide an underlying predictive model when conditioned on a run length. Specifically, we define the run length, r_t, as the number of timesteps since the current (at time t) task was sampled. We write $\boldsymbol{\eta}_t[r]$ to denote the posterior statistics at time t associated with a run length of r. More specifically, a meta-learning algorithm updates some

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

initialization based on the previous r samples, and $\boldsymbol{\eta}_t[r]$ refers to these updated parameters. A review of BOCPD and a unifying perspective on meta-learning in terms of posterior statistics is presented in the appendix. In the following subsections, we first derive MOCA's Bayesian filtering updates, and then outline how the full framework can be used to both train and evaluate meta-learning models on time series without task segmentation.

2.1 Bayesian Run-length Filtering

As in BOCPD, MOCA maintains a belief over possible run lengths r_t. Throughout this paper, we use b_t to refer to the updated belief before observing data at that timestep, $(\boldsymbol{x}_t, \boldsymbol{y}_t)$. Note that b_t is a discrete distribution with support over $r_t \in \{0, ..., t-1\}$.

At time t, the agent first observes the input $\boldsymbol{x}_t$, then makes a prediction $p(\hat{\boldsymbol{y}}_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1})$, and subsequently observes $\boldsymbol{y}_t$. Generally, the latent task can influence both the marginal distribution of the input, $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t-1})$ as well as the conditional distribution $p(\boldsymbol{y}_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1})$. Thus, the agent can update its belief over run lengths once after observing the input $\boldsymbol{x}_t$, and again after observing the label $\boldsymbol{y}_t$. We will use $b_t(r_t \mid \boldsymbol{x}_t) = p(r_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1})$ to represent the updated belief over run length after observing only $\boldsymbol{x}_t$, and $b_t(r_t \mid \boldsymbol{x}_t, \boldsymbol{y}_t) = p(r_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t})$ to represent the fully updated belief over r_t after observing $\boldsymbol{y}_t$. Finally, we will propagate this forward in time according to our assumptions on task dynamics to compute $b_{t+1}(r_{t+1})$, which is used in the subsequent timestep.

To derive the Bayesian update rules, we start by noting that the updated posterior is proportional to the joint density,

$$\begin{aligned} b_t(r_t \mid \boldsymbol{x}_t) &= p(r_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1}) = Z^{-1} p(r_t, \boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t-1}) \\ &= Z^{-1} p(\boldsymbol{x}_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t-1}, r_t) p(r_t \mid \boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t-1}) \\ &= Z^{-1} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{\eta}_{t-1}[r_t]) b_t(r_t) \end{aligned} \tag{2}$$

where the normalization constant Z can be computed by summing over the finite support of $b_{t-1}(r_t)$. Importantly, this update requires $p_{\boldsymbol{\theta}}(\boldsymbol{x}_t \mid \boldsymbol{\eta}_{t-1}[r_t])$, the base meta-learning algorithm's posterior predictive density over the inputs. Within classification, this density is available for generative models, and thus a generative approach is favorable to a discriminative approach within MOCA. In regression, it is uncommon to estimate the distribution of the independent variable. We take the same approach in this work and assume that $\boldsymbol{x}_t$ is independent of the task for regression problems, in which case $b_t(r_t \mid \boldsymbol{x}_t) = b_t(r_t)$. We discuss the specific choice of underlying meta-learning models in the regression and classification settings in the appendix.

Next, upon observing $\boldsymbol{y}_t$, we can similarly factor the belief over run lengths for the next timestep,

$$b_t(r_t \mid \boldsymbol{x}_t, \boldsymbol{y}_t) = Z^{-1} p_{\boldsymbol{\theta}}(\boldsymbol{y}_t \mid \boldsymbol{x}_t, \boldsymbol{\eta}_{t-1}[r_t]) b_t(r_t \mid \boldsymbol{x}_t). \tag{3}$$

Again, the normalization constant can be computed via a sum over the support of r_t.

Finally, we must propagate this belief forward in time to obtain $b_{t+1}(r_{t+1})$:

$$\begin{aligned} b_{t+1}(r_{t+1}) &= p(r_{t+1} \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t}) = \sum_{r_t} p(r_{t+1}, r_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t}) \\ &= \sum_{r_t} p(r_{t+1} \mid r_t, \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t}) p(r_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t}) = \sum_{r_t} p(r_{t+1} \mid r_t) b_t(r_t \mid \boldsymbol{x}_t, \boldsymbol{y}_t). \end{aligned}$$

where we have exploited the assumption that the changes in task, and hence the evolution of run length r_t, happen independently of the data generation process. The conditional run-length distribution $p(r_{t+1} \mid r_t)$ is defined by our model of task evolution.

Recall that we assume that the task switches with fixed probability λ, the hazard rate. Thus, for all r_t, $p(r_{t+1} = 0 \mid r_t) = \lambda$, implying

$$b_{t+1}(r_{t+1} = 0) = \sum_{r_t} \lambda b_t(r_t \mid \boldsymbol{x}_t, \boldsymbol{y}_t) = \lambda. \tag{4}$$

Conditioned on the task remaining the same, $r_{t+1} = k > 0$ and $r_t = k - 1$. Thus, $p(r_{t+1} = k \mid r_t) = (1 - \lambda)\mathbb{1}\{r_t = k - 1\}$ implying

$$b_{t+1}(r_{t+1} = k) = (1 - \lambda) b_t(r_t = k - 1 \mid \boldsymbol{x}_t, \boldsymbol{y}_t). \tag{5}$$

Equations (4) and (5) together define b_{t+1} over its support $r_{t+1} \in \{0, \ldots, t\}$

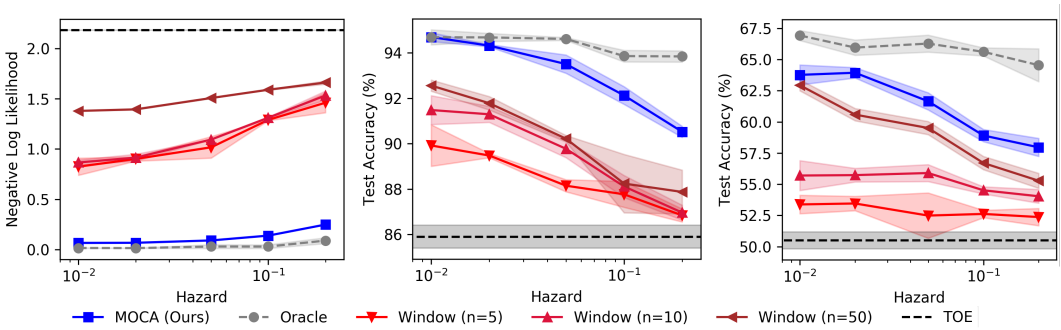

Figure 1: Performance of MOCA versus baselines in sinusoid regression (**left**; lower is better), Rainbow MNIST (**center**; higher is better), and miniImageNet (**right**; higher is better), versus hazard rate. Note that for both problems, MOCA always outperforms the baselines and the performance degrades only slightly from the performance of the oracle. In contrast, sliding window methods result in severely degraded performance.

2.2 Meta Learning without Task Segmentation

By taking a Bayesian filtering approach to changepoint detection, we avoid hard assignments of changepoints and instead perform a soft selection over run lengths. In this way, MOCA is able to backpropagate through the changepoint detection and directly optimize the underlying predictive model, which may be any meta-learning model that admits a probabilistic interpretation.

MOCA processes a time series sequentially. We initialize $b_1(r_1 = 0) = 1$, and initialize the posterior statistics for $\boldsymbol{\eta}_0[r_1 = 0]$ as specified by the parameters $\boldsymbol{\theta}$ of the meta-learning algorithm. Then, at timestep t, we first observe inputs $\boldsymbol{x}_t$ and update our belief over run length accordingly, computing $b_t(r_t \mid \boldsymbol{x}_t)$ according to (2). Next, we marginalize over this belief to make a probabilistic prediction for the label $\boldsymbol{y}_t$,

$$p_{\boldsymbol{\theta}}(\hat{\boldsymbol{y}}_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1}) = \sum_{r_t=0}^{t-1} b_t(r_t \mid \boldsymbol{x}_t) p_{\boldsymbol{\theta}}(\hat{\boldsymbol{y}}_t \mid \boldsymbol{x}_t, \boldsymbol{\eta}_{t-1}[r_t]) \tag{6}$$

We then observe the true label $\boldsymbol{y}_t$ and incur the corresponding negative log likelihood loss. We can then use this observation to update both the belief over run length, computing $b_t(r_t \mid \boldsymbol{x}_t, \boldsymbol{y}_t)$ according to (3), as well as update the posterior statistics for all the run lengths using the labeled example. A recursive update rule for $\boldsymbol{\eta}$ allows these parameters to be computed efficiently using the past values of $\boldsymbol{\eta}$

$$\boldsymbol{\eta}_t[r] = h(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{\eta}_{t-1}[r-1]) \quad \forall\, r = 1, \ldots, t. \tag{7}$$

While MOCA could be used with an algorithm which didn't admit such a recursive update rule, this would require storing data online and running the non-recursive posterior computation (8) on $\mathcal{D}_{-r_t}$ for every r_t, which involves t operations using datasets of sizes from 0 to t, and thus can be an $O(t^2)$ operation. In contrast, the recursive updates involve t operations involving just the latest datapoint, yielding $O(t)$ complexity. Finally, we propagate the belief over run length forward in time according to (4) and (5) to obtain $b_t(r_{t+1})$ to be ready to process the next data point.

Since all these operations are differentiable, given a training time series in which there are task switches $\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}$, we can run this procedure, sum the NLL losses incurred at each step, and use backpropagation within a standard deep learning framework to optimize the parameters of the base learning algorithm $\boldsymbol{\theta}$. Algorithm 1 outlines this training procedure. In practice, we sample shorter time-series of length T from the training data to ease computational requirements during training; we discuss implications of this in the appendix. If available, a user can input various levels of knowledge on task segmentation by manually updating $b(r_t)$ at any time; further details on this task semi-segmented use case are provided in the appendix.

3 Experimental Results and Conclusions

We investigate the performance of MOCA in three problem settings: one in regression and two in classification. Our primary goal is to characterize the impact on performance of using MOCA to move from the standard *task-segmented* meta-learning setting to the task-unsegmented case. To this end, we investigate the performance of MOCA versus an "oracle" model that uses the same base meta-learning algorithm, but has access to exact task segmentation at train and test time. We additionally compare against baseline sliding window models of various window lengths, which again use the same meta-learning algorithm, but always condition on the last n data points. These baselines are a competitive approach to learning in time-varying data streams (Gama et al., 2014)

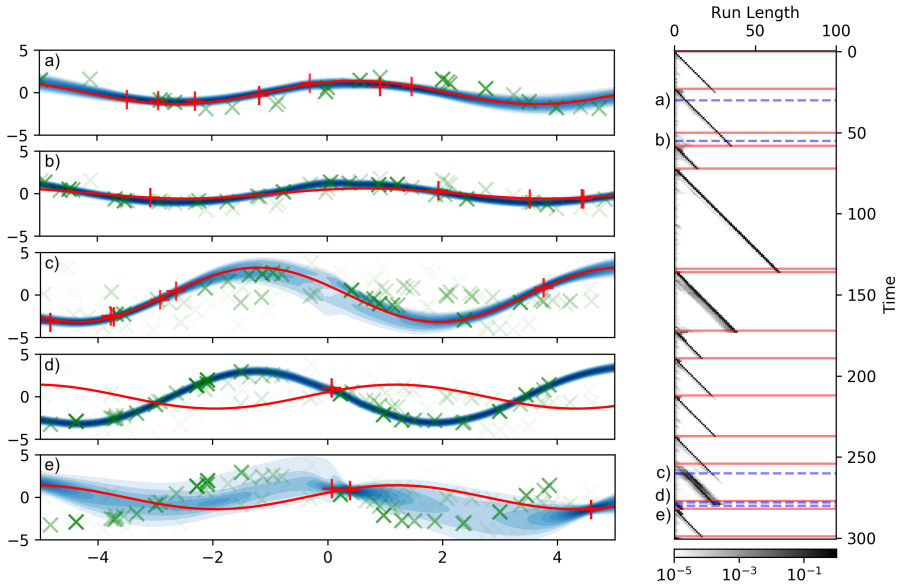

Figure 2: The performance of MOCA on the sinusoid regression problem. **Right:** The belief over run length versus time. The intensity of each point in the plot corresponds to the belief in run length at the associated time. The red lines show the true changepoints. **Left:** Visualizations of the posterior predictive density corresponding to the blue dotted lines in the figure on the right. The red line denotes the current function (task), and red points denote samples from that function. Green points denote data from previous tasks, where more faint points are older. **a)** A visualization of the posterior at an arbitrary time. **b)** The visualization of the posterior for a case in which MOCA did not successfully detect the changepoint. In this case, it is because the pre- and post-change function (corresponding to figure a and b) are highly similar. **c)** An instance of a multimodal posterior. **d)** The changepoint is initially missed due to the data generated from the post-change function being highly likely under the previous posterior. **e)** After an unlikely data point, the model increases its uncertainty as the changepoint is detected.

and have been used effectively for meta-learning in time-varying settings (see e.g. Nagabandi et al. (2019a)). Finally, we compare to a "train on everything" model, which only learns a prior and does not adapt online, corresponding to a standard supervised learning approach. Performance of MOCA versus baselines is presented in Fig. 1 for all problem domains. In addition, we investigate in isolation the effects of task-segmentation information when provided at train-time and at test-time and under partial task supervision. Due to space constraints, we defer this to the appendix.

Regression. To characterize MOCA in the regression setting, we investigate the performance on a switching sinusoid problem adapted from (Finn et al., 2017), in which a task change corresponds to a re-sampled sinusoid phase and amplitude. Qualitative results are visualized for the sinusoid in Fig. 2, as well as a visualization of the belief over run length at each time. Qualitatively, MOCA is capable of accurate and calibrated posterior inference with only a handful of data points, and is capable of identifying task change extremely rapidly. Typically, it identifies task change in one timestep, if the generated data does not happen to have high likelihood under the previous task as in Fig. 2d. Quantitatively, MOCA achieves performance close to the oracle model and substantially outperforms the sliding window approaches for all hazard rates.

Classification. In the classification setting, we apply MOCA to the Rainbow MNIST dataset of Finn et al. (2019) and the miniImageNet benchmark task (Vinyals et al., 2016). In Rainbow MNIST, MNIST digits have been perturbed via a color transformation, rotation, and scaling, and each task corresponds to a unique combination of these transformations. miniImagenet consists of 100 ImageNet categories (Deng et al., 2009), each with 600 RGB images of resolution 84×84. In our continual learning setting, we associate each class with a semantic label that is consistent between tasks. Specifically, we split the miniImageNet dataset in to five approximately balanced high level classes, which we refer to as super-classes, as five-way classification is standard for miniImageNet (Vinyals et al., 2016; Snell et al., 2017); details are provided in the appendix. Then, a new task corresponds to sampling a new class within each super-class, and the problem is to classify an image as belonging to a specific super-class. This enables knowledge re-use between classes, and corresponds to a continual learning scenario in which each super-class experiences distributional shift. Note that this is somewhat different from the typical task in few-shot learning, where classes have no *a priori* semantic meaning. In both experiments, MOCA outperforms baselines for all hazard rates. On Rainbow MNIST, MOCA approaches oracle performance, likely due in part to the fact that task change can usually be detected via a change in digit color.

References

Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv:0710.3742*, 2007.

Maruan Al-Shedivat, Trapit Bansal, Yuri Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *International Conference on Learning Representations (ICLR)*, 2018.

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *International Conference on Learning Representations (ICLR)*, 2019.

Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2016.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

Paul Fearnhead and Zhen Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2007.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *International Conference on Machine Learning (ICML)*, 2017.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. *International Conference on Machine Learning (ICML)*, 2019.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 1999.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 2014.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S.M. Ali Eslami, and Yee Whye Teh. Neural processes. *International Conference on Machine Learning (ICML)*, 2018.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical Bayes. *International Conference on Learning Representations (ICLR)*, 2018.

James Harrison, Apoorva Sharma, and Marco Pavone. Meta-learning priors for efficient online Bayesian regression. *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2018.

Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2016.

Xu He, Jakub Sygnowski, Alexandre Galashov, Andrei A Rusu, Yee Whye Teh, and Razvan Pascanu. Task agnostic continual learning via meta learning. *arXiv preprint arXiv:1906.05201*, 2019.

Khurram Javed and Martha White. Meta-learning representations for continual learning. *Neural Information Processing Systems (NeurIPS)*, 2019.

Ghassen Jerfel, Erin Grant, Thomas L Griffiths, and Katherine Heller. Online gradient-based mixtures for transfer modulation in meta-learning. *Neural Information Processing Systems (NeurIPS)*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

Jeremias Knoblauch and Theodoros Damoulas. Spatio-temporal bayesian on-line changepoint detection with model selection. *International Conference on Machine Learning (ICML)*, 2018.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *Neural Information Processing Systems (NeurIPS)*, 2013.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2019a.

Anusha Nagabandi, Chelsea Finn, and Sergey Levine. Deep online learning via meta-learning: Continual adaptation for model-based RL. *arXiv:1812.07671*, 2019b.

Ulrich Paquet. Empirical Bayesian change point detection. *Graphical Models*, 2007.

Y Saatci, R Turner, and CE Rasmussen. Gaussian process change point models. *International Conference on Machine Learning (ICML)*, 2010.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Neural Information Processing Systems (NeurIPS)*, 2017.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer, 2012.

Ryan Turner, Yunus Saatci, and Carl Edward Rasmussen. Adaptive sequential Bayesian change point detection. *NeurIPS Workshop on Nonparametric Bayes*, 2009.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Neural Information Processing Systems (NeurIPS)*, 2016.

Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 1985.

Robert C Wilson, Matthew R Nassar, and Joshua I Gold. Bayesian online learning of the hazard rate in change-point problems. *Neural Computation*, 2010.