

A Behavior-Aware Approach for Deep Reinforcement Learning in Non-stationary Environments without Known Change Points

Zihe Liu, Jie Lu, Guangquan Zhang and Junyu Xuan

Australian Artificial Intelligence Institute (AII), University of Technology Sydney

Zihe.Liu@student.uts.edu.au, {Jie.Lu, Guangquan.Zhang, Junyu.Xuan}@uts.edu.au

Abstract

Deep reinforcement learning is used in various domains, but usually under the assumption that the environment has stationary conditions like transitions and state distributions. When this assumption is not met, performance suffers. For this reason, tracking continuous environmental changes and adapting to unpredictable conditions is challenging yet crucial because it ensures that systems remain reliable and flexible in practical scenarios. Our research introduces Behavior-Aware Detection and Adaptation (BADA), an innovative framework that merges environmental change detection with behavior adaptation. The key inspiration behind our method is that policies exhibit different global behaviors in changing environments. Specifically, environmental changes are identified by analyzing variations between behaviors using Wasserstein distances without manually set thresholds. The model adapts to the new environment through behavior regularization based on the extent of changes. The results of a series of experiments demonstrate better performance relative to several current algorithms. This research also indicates significant potential for tackling this long-standing challenge.

1 Introduction

Deep reinforcement learning has extensive applications in economics [Mosavi *et al.*, 2020], energy engineering [Declarue *et al.*, 2020; Oikonomou *et al.*, 2023], medical analysis [Hu *et al.*, 2023; Tiwari *et al.*, 2023] and other domains, where policies are trained to make optimal sequential decisions in an assumed stationary environment. However, in practice, stationary environments are rare. Instead, the norm is non-stationary environments where the underlying environment can change in quite unpredictable and abrupt ways. For instance, outdoor robots must navigate constantly changing terrain and lighting levels, while financial markets should rapidly shift alongside breaking news and global events. Hence, ignoring the non-stationarity of underlying environments will frequently lead to poor performance even using a superior algorithm. There is no doubt that addressing this issue requires a dedicated strategy.

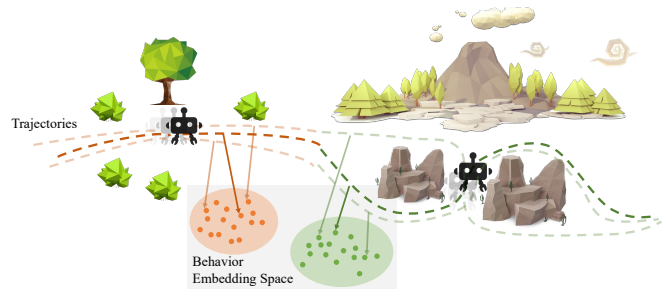


Figure 1: When an outdoor robot moves from flat terrain to mountains, its speed, direction, and acceleration control changes corresponding to the changing conditions. We believe these variations can be fully captured through behavior.

In prior work, several research teams have looked for solutions. Some have converted the problem into a continual multi-task reinforcement learning problem [Kirkpatrick *et al.*, 2017; Schwarz *et al.*, 2018], while others have transformed the issue into a meta reinforcement learning problem [Yu *et al.*, 2020; Xie *et al.*, 2021]. Yet the common thread in all these studies is that the change points need to be known in advance, as these change points are used to divide the non-stationary environment into multiple tasks. However, there are often no ready-to-use indicators for unpredictable changes. Furthermore, a typical continual learning setting focuses on preventing catastrophic forgetting, while remembering the knowledge from previous tasks may not contribute to the current adaptation, especially in more practical environments without cyclically recurring tasks. To address the absence of known change points, some research actively detects environmental changes using methods like reward-based detection [Lomonaco *et al.*, 2020; Kirkpatrick *et al.*, 2017] or state-based detection [Padakandla *et al.*, 2020]. However, the reward-based method generally requires timely rewards and manually set thresholds. In addition, the state information is not comprehensive and accurate enough for detection because different global behaviors may have the same final state or perform similar actions at a local level [Pacchiano *et al.*, 2020]. Therefore, changes in state alone do not serve as reliable indicators for determining environmental changes.

We posit that the agents in an environment can be better

characterized through their behavior. In our research, behavior represents the embeddings mapped from the sequences of states, actions and rewards during a period. As demonstrated in Fig. 1, when an outdoor robot encounters different environmental conditions, such as terrain, its speed and direction tend to demonstrate significant changes from those of the previous terrain. However, the separate variables like speed and direction at a few time steps can not describe the comprehensive trajectory change, making it challenging to understand and adapt to the new environment. In contrast, behavior can offer more comprehensive information from a global level. We believe that behavior distribution changes simultaneously reflect environmental changes and can help us adapt to new conditions, so our proposed method uses behavior as a core indicator and knowledge. We propose using these shifts in behavior distribution to detect environmental changes. Additionally, these changes indicate that departing from the behavior in the original environment is beneficial for optimal behaviors within the new conditions.

Inspired by this, we present a novel approach to detect environment changes by monitoring behavior distribution shifts based on the Wasserstein distance [Villani, 2009; Panaretos and Zemel, 2019]. The agent(s)' behavior is then regularized accordingly to help the policy steer away from the previous optimum and adapt to new environmental conditions. Experiments in benchmark environments prove our method to be effective and accurate compared to other methods. We propose a setting that enhances the applicability and effectiveness of reinforcement learning across diverse fields, from robotics navigating in dynamic landscapes to trading systems that can respond to volatile markets.

Our main contributions are summarized as follows,

- We propose an environmental change detection method, testing environmental change points through the Wasserstein distance between the global behavior information without manually setting thresholds.
- With detected change points, we introduce a policy adaptation method that facilitates faster deviation from the previous optimum and exploration of new behavioral regions. We adjust regularization based on the extent of change to ensure adaptability under various conditions.
- We provide an end-to-end framework called Behavior-Aware Detection and Adaptation (BADA) to collaborate environment change detection and adaptation by analyzing and employing behavior.

2 Related Work

Change detection in RL. Several partial models [Da Silva *et al.*, 2006; Hadoux *et al.*, 2014] have been published that represent environmental contexts using a quality signal, but neither method works well in complex scenarios. Online Parametric Dirichlet Change Point (ODCP) [Prabuchandran *et al.*, 2021] detects environmental changes by converting data into unconstrained multivariate data. At the same time, CRL-Unsup[Lomonaco *et al.*, 2020] uses the gap between short and long-term rewards as an indicator, which relies on manually selected thresholds. Liu *et al.* [Liu *et al.*, 2024] de-

tect the changes by analyzing the joint distribution of state and policy, lacking a comprehensive perspective over time.

Adaptive/Transfer RL. Another feature of CRL-Unsup [Lomonaco *et al.*, 2020] is that it adapts to new environments using elastic weight consolidation (EWC) [Kirkpatrick *et al.*, 2017]. In ODCP [Padakandla *et al.*, 2020], when a change point is detected, the Q value of the relevant model is used as an update parameter. Several approaches learn a latent representation incorporating shared and specific components from the source domain [Huang *et al.*, 2022; Zintgraf *et al.*, 2019; Trabucco *et al.*, 2022]. Some work[Huang *et al.*, 2022; Zintgraf *et al.*, 2019; Trabucco *et al.*, 2022] learn a latent representation encompassing shared and specific components from source domains. These methods typically have clear task definitions and differentiate between the source and target domain. By contrast, our method, BADA, is designed for sequential changes and continuously adapts to new tasks.

Continual RL. In continual scenarios, the focus tends to be placed on avoiding catastrophic forgetting, which is the tendency of a neural network to abruptly forget previously learned tasks upon learning a new task. Most schemes in continual learning are trained on pairs of separate tasks, and discrete transitions are often used to inform adaptations in the model. Some approaches add additional structures to the network model to resist forgetting [Zenke *et al.*, 2017; Schwarz *et al.*, 2018; Aljundi *et al.*, 2017] added additional structures to network models to resist forgetting. Other studies achieve this goal by introducing additional data or label inputs [Shin *et al.*, 2017; Lopez-Paz and Ranzato, 2017]. However, the change points of environments are usually unpredictable, which makes it difficult to deploy continual learning methods in such settings directly.

Meta RL. Meta reinforcement learning typically consists of meta-training and meta-testing, with the goal of learning a policy capable of adapting to new tasks from a given task distribution. Some methods learn latent variable models to infer the task embedding from current and past experiences. Off-policy reinforcement learning is then performed with this latent variable [Rakelly *et al.*, 2019; Bing *et al.*, 2023; Xie *et al.*, 2021]. Unlike meta-reinforcement learning, where the training and testing tasks are separate, BADA focuses on real-time adaptation during the training phase. Moreover, BADA does not assume that the tasks come from one distribution.

Multi-task RL. Multi-task learning within varied task families often faces the challenge of negative transfer among dissimilar tasks, which can impede training. Previous research addresses this issue by evaluating task relatedness using a validation loss for different tasks [Liu *et al.*, 2022; Fifty *et al.*, 2021; Standley *et al.*, 2020]. Other complementary methods for sharing information include sharing data, parameters or representations, and sharing behaviors [Yu *et al.*, 2021; Yu *et al.*, 2022; Sasaki and Yamashina, 2020; D'Eramo *et al.*, 2020]. The goal with multi-task settings is to train an agent to perform well at various tasks simultaneously. By contrast, BADA focuses on adapting to the current environment for optimal performance.

3 Methodology

3.1 Problem Formulation

A Markov decision process M is defined by a state space S , a starting state distribution $p_0(s)$, an action space A , a transition dynamics $\mathcal{P}(s_{t+1}|s_t, a)$, and a reward function $\mathcal{R} : S \times A \rightarrow \mathbb{R}$. A policy π_θ is parameterized by θ . The interaction trajectory $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \dots\}$ is collected by a policy π_θ . With a discount factor of γ , the optimal policy is the one that maximizes the expected discounted reward: $\pi^* = \arg \max_{\pi} \mathbb{E}_{\tau \sim \pi} [\sum_t \gamma^t \mathcal{R}(s_t, a_t)]$.

Standard reinforcement learning assumes that the underlying M is unknown but fixed. When this assumption does not stand, a reinforcement learning scheme for non-stationary environments must be implemented. Further, this paper targets a specific problem within non-stationary environments, in which the change happens suddenly, and the change points are unknown. Formally:

Problem 1. Let $\{M_{k=1:K}\}$ be a sequence of different MDPs with unknown switch points $\{C_1, \dots, C_{K-1}\}$ and arbitrary order. An agent will sequentially interact with $\{M_{k=1:K}\}$ with unknown change points, where the goal is to find an optimal policy to maximize the long-term cumulative reward:

$$\pi_{1:J}^* = \arg \max_{\pi_{1:J}} \mathbb{E}_{\tau \sim \pi} \left[\sum_{k=1:J} \sum_{t=C'_{k-1}}^{C'_k} \gamma^t \mathcal{R}(s_t, a_t) \right], \quad (1)$$

where $\{C'_1, \dots, C'_J\}$ are the detected change points.

Remark 1. Each MDP M_k is distinct, potentially differing in state spaces, transition dynamics, and reward functions.

Remark 2. The duration of the agent’s interaction in each MDP, $C_k - C_{k-1}$, is not predetermined and assumed.

To ensure the maximum long-term reward, the problem encompasses two sub-goals: detecting change points accurately and adapting to the new environment rapidly.

3.2 Behavior-based Change Detection

During the training process, the policy continuously interacts with the environment. Within each update epoch t , the trajectories collected by π_θ are denoted as $\tau = \{s_0, a_0, r_0, \dots, s_H, a_H, r_H\}$, where H is the step taken in this epoch. A behavioral embedding map $\Phi : \Gamma \rightarrow \mathcal{E}$ maps the trajectories into a behavioral latent space. In our particular implementation, this map function is a multilayer perceptron. The embedding \mathbb{P}_θ represents the behavior embedding distribution corresponding to policy π_θ at epoch t .

As mentioned previously, environmental non-stationarity leads to a shift in the trajectory. Therefore, the behavior distributions from two adjacent epochs $\{\mathbb{P}_{t-1}, \mathbb{P}_\theta\}$ are used to quickly identify the change points promptly. Here, the Wasserstein distance [Olkin and Pukelsheim, 1982; Panaretos and Zemel, 2019] is used as the measure for evaluating the difference between behavior trajectories. The Wasserstein distance originates from the optimal transport problem, which evaluates the cost required to transform one probability distribution into another. Given two distributions μ, ν , the Wasserstein distance is defined as

$$W(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu, \nu)} \int c(x, y) d\gamma(x, y), \quad (2)$$

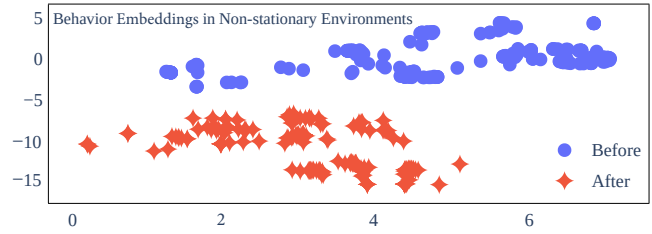


Figure 2: This figure presents a t-SNE plot of behavior. The distinct clusters demonstrate the significant impact of environmental changes on behavior and inspire us to use the behavior to adapt actively to coming changes.

where $\Pi(\cdot, \cdot)$ denotes the joint distribution with marginal distributions, and $c(\cdot, \cdot)$ denotes the cost function quantifying the distance between two points. If the cost of a move is simply the distance between the two points, then the optimal cost is identical to the definition of the Wasserstein 1-distance [Xu, 2019]. We calculate the distance by using the dual form of Eq. (2), which is defined as:

$$W(\mu, \nu) = \sup_{f_\mu, f_\nu} \int f_\mu d\mu(x) - \int f_\nu d\nu(y) \quad , \quad (3)$$

where $f_\mu, f_\nu : \mathbb{R}^d \rightarrow \mathbb{R}$ and $\text{Lip}(f_\mu) \leq 1$. The $\text{Lip}(f)$ denotes the minimal Lipschitz constant for the function f . To calculate the Wasserstein distance, the objective is to find the optimal f_μ^*, f_ν^* to maximize the integral.

Wasserstein distance is a metric that reflects the proximity between two distributions, even if no overlap components exist. This property is important for our problem because the agent may manifest completely different behavior before and after changes. Therefore, the support between these distributions on behavior spaces would be limited, and then a proper distribution distance definition for such a situation is crucial. Additionally, its symmetrical nature offers a more effective measure of the differences between distributions compared with other options, like KL divergence. For example, as Fig. 2 shows, when a policy is sequentially trained from one environment to another – say where the textures and lighting change – the agent’s behavior embedding distribution will show a distinct shift in distribution without overlapping. This observation can also help us identify these behavioral-level changes using the Wasserstein distance.

With the evaluated distance before and after a potential change point, we still need to decide on a change point, usually based on a manually determined threshold. It is difficult because it depends on the environment and behavior distributions, and what is even worse is that different change points may need different thresholds. Here, we propose to perform the permutation test [Welch, 1990; Van Borkulo *et al.*, 2022], which infers the presence of any change points. The permutation test is an exact statistical hypothesis test based on proof by contradiction. This method involves permuting the order of samples, recalculating statistical test metrics, constructing an empirical distribution, and then determining the p-value based on this distribution to make inferences.

To explain the permutation idea, given two samples from adjacent behavior embedding distributions $\mathbb{P}_\theta, \mathbb{P}_{t-1}$ and cal-

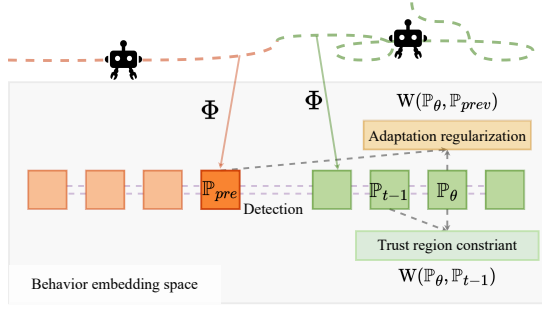


Figure 3: The BADA framework. When a change is detected through the behavior distribution permutation test, regularization will be added to deviate policy behavior from the previous optimum.

culate the test statistic $T = W(\mathbb{P}_\theta, \mathbb{P}_{t-1})$. The typical null hypothesis is given by:

$$H_0 : \mathbb{P}_\theta = \mathbb{P}_{t-1}, \quad (4)$$

i.e., all samples come from the same distribution. Then, for each permutation $e = 1, 2, \dots, E$, randomly permute the components of $\mathbb{P}_\theta \cup \mathbb{P}_{t-1}$, and split the permuted data into $\mathbb{P}_\theta^{(e)}, \mathbb{P}_{t-1}^{(e)}$ with the original sizes, then calculate test statistics $T_e = W(\mathbb{P}_\theta^{(e)}, \mathbb{P}_{t-1}^{(e)})$. By repeating the permutation and calculation, a p-value is given by

$$p = \frac{1}{E} \sum_{t=1}^E 1\{T_e \geq T\}, \quad (5)$$

where 1 is an indicator function. This test is guaranteed to control the type-I error [Good, 2013] because we evaluate the p-value of the test via the permutation approach. In addition, the non-parametric nature, i.e., that it does not rely on assumptions about data distribution. As Fig. 2 shows, the trajectory distribution usually does not conform to an easily computable and representable form of distribution. Therefore, using a permutation test is highly suitable for solving our problem. Suppose the p-value is lower than the significance level; in that case, the current epoch t is noted as a change point $c = t$, and $\mathbb{P}_{pre} = \mathbb{P}_{c-1}$ is the optimal behavior distribution corresponding to the previous environment.

3.3 Behavior-Aware Adaptation

Although the vanilla DRL can adapt to the new environment, especially in gradually changing environments, it typically requires many interactions that sample inefficient and generate a significant delay. With the detection signal from the above section, we aim to achieve fast adaption. Since our detection is based on Wasserstein distance, we follow Wasserstein-based policy gradient baseline - Behavior Guided Policy Gradients (BGPG) [Pacchiano *et al.*, 2020]. Its training objective (for stationary environment) is to maximize:

$$F(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_\theta} [\mathcal{R}(\tau)] - W(\mathbb{P}_\theta, \mathbb{P}_{t-1}), \quad (6)$$

where \mathbb{P}_{t-1} is the behavior distribution of last update epoch.

When a policy converges in one environment, the behavior will enter a relatively stable distribution, providing a basis

Algorithm 1 Behavioral Aware Detection and Adaptation (BADA)

Initialize: Policy π_θ , behavioral embedding mapping function Φ , and significance level α .

- 1: **for** Epoch $t = 1, 2, \dots$ **do**
- 2: Collect $\tau = \{s_0, a_0, r_0, \dots, s_H, a_H, r_H\}$ from the current environment.
- 3: Obtain behavior embedding \mathbb{P}_θ by behavioral embedding mapping function Φ .
- 4: Compute the original statistics $T = W(\mathbb{P}_\theta, \mathbb{P}_{t-1})$.
- 5: **for** Permute iteration $e = 1, 2, \dots, E$ **do**
- 6: Shuffle $\mathbb{P}_\theta \cup \mathbb{P}_{t-1}$ and split the data into $\mathbb{P}_\theta^{(e)}, \mathbb{P}_{t-1}^{(e)}$ and compute statistics $T_e = W(\mathbb{P}_\theta^{(e)}, \mathbb{P}_{t-1}^{(e)})$.
- 7: **end for**
- 8: Obtain the p-value $\frac{1}{E} \sum_{t=1}^E 1\{T_e \geq T\}$
- 9: **if** p-value $\leq \alpha$ at epoch c **then**
- 10: Save \mathbb{P}_{c-1} as previous behavior distribution \mathbb{P}_{pre} .
- 11: Update policy parameter by $\theta_{t+1} \leftarrow \theta_t - \alpha \nabla_\theta F(\theta_t)$ following Eq. (7)
- 12: **else**
- 13: Update policy parameter by $\theta_{t+1} \leftarrow \theta_t - \alpha \nabla_\theta F(\theta_t)$ following Eq. (6).
- 14: **end if**
- 15: Save $\mathbb{P}_{t-1} \leftarrow \mathbb{P}_\theta$ for environment change detection.
- 16: **end for**

for us to detect environmental changes. When a change point is detected at epoch c , indicating that a significant change in the environment has occurred, \mathbb{P}_{c-1} is saved as a previous optimal behavior distribution \mathbb{P}_{pre} . To assist the policy in quickly deviating from the optimal behavior of the previous environment, we propose to add a regularizer that maximizes the difference between the current behavior distribution and the previously converged behavior distribution. This new objective function is designed as follows:

$$F(\theta) = \mathbb{E}_{\tau \sim \mathbb{P}_\theta} [\mathcal{R}(\tau)] - W(\mathbb{P}_\theta, \mathbb{P}_{t-1}) + \delta W(\mathbb{P}_\theta, \mathbb{P}_{pre}), \quad (7)$$

where $\mathcal{R} = \sum A^{\pi_{t-1}}(s_i, a_i) \frac{\pi_\theta(a_i|s_i)}{\pi_{t-1}(a_i|s_i)}$, $A^{\pi_{t-1}}(s_i, a_i)$ is the advantage function, and \mathbb{P}_{pre} is the converged behavior distribution in the previous environment, and $\delta \in \mathbb{R}_{>0}$ is a hyper-parameter. Here, we use the adjacent behavior distance on the detected change point $W(\mathbb{P}_{c-1}, \mathbb{P}_c)$ as δ , depending on the extent of change. This self-adjusted coefficient ensures that the adaptation regularization has a greater impact as the level of environmental change increases.

If no change is detected, the adaptation term will not work, so δ will be set as zero. The first penalty constrains policy updates within a trust region, ensuring the validity of importance sampling. However, this constraint can lead to slow adaptation when the environment undergoes abrupt changes, as the policy hesitates to deviate from its previous optimal behavior. At the change point c , $\mathbb{P}_{pre} = \mathbb{P}_{c-1}$. Only following the first penalty term at this point might trap the policy in a suboptimal area for an extended period. Therefore, our second adaptation regularization serves as a contrastive term, steering the policy away from previous behavior. As the policy gradually adapts to the current environment, i.e., $t \gg c$, the adap-

tation term $W(\mathbb{P}_\theta, \mathbb{P}_{c-1})$ and the first term $W(\mathbb{P}_\theta, \mathbb{P}_{t-1})$ no longer conflict. The penalty constraints ensure performance improvement in a stationary environment, and the role of the adaptation term weakens as the policy moves away from the previous optimum.

With the optimal f_μ^*, f_ν^* according to Eq. (3), the regularization term in Eq. (7) is:

$$W(\mathbb{P}_\theta, \mathbb{P}_{pre}) \approx \mathbb{E}_{\tau \sim \mathbb{P}_\theta} [f_\mu^*(\tau)] - \mathbb{E}_{\phi \sim \mathbb{P}_{pre}} [f_\mu^*(\phi)]. \quad (8)$$

Maximizing this term can guide the optimization by favoring those trajectories that show more difference between old ones. When another change occurs, we consider only the preceding behavior distribution. We believe excessive constraints may lead to a narrow area and result in local optima. Therefore, focusing on the immediate historical behavior ensures adaptability to changing environments without introducing unnecessary complexities. This training goal allows us to scale to scenarios with multiple changes easily. Fig. 3 illustrates the adaptation scheme, and Algorithm 1 describes the complete BADA method in detail.

4 Experiments and Analysis

This section comprehensively evaluates our BADA method, addressing key questions: 1) Can BADA achieve higher rewards in environments without known change points? 2) Is behavior-based change detection superior to alternative methods? 3) Does BADA’s adaptation method outperform retraining and other adaptation approaches? 4) Can BADA maintain performance with frequent environmental changes? These inquiries guide our experiments and analysis.

4.1 Settings

Environments. We conducted all experiments within ViZ-Doom [Wydmuch *et al.*, 2019], a first-person shooting game with various scenarios. This environment allows reinforcement learning agents to be developed using only visual information (the screen buffer). We chose four scenarios to evaluate our proposed method. We employ distinct challenges and modifications to simulate dynamic environments for agent training. For example, as Fig. 4 shows, the environment transit from high-contrast *simpler_basic* to dimly lit *basic* settings, shift from defending a line in a rectangle map to defending a point in a circular map against enemies in *defend_the_line/center*. In addition, we adjust the number of enemies in *deadly_corridor* and change the medikit textures in the *health_gathering* scenario to represent new rooms. Agents need to respond to these changes.

Comparison methods. In all experiments, we used the PPO/TRPO update, and once the environment changed, the model could not access any information on the changed environmental conditions. Further, we compared BADA to three baseline methods as follows.

- PPO [Schulman *et al.*, 2017] and TRPO [Schulman *et al.*, 2015] without detection and adaptation;
- Behavior-based BGPG [Pacchiano *et al.*, 2020] without detection and adaptation;
- CRL-Unsup [Lomonaco *et al.*, 2020] with both detection and adaptation.

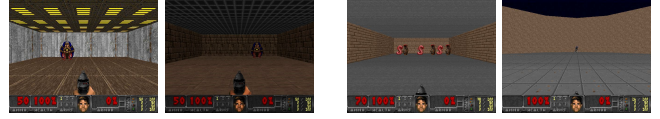


Figure 4: The simulated non-stationary environments. The left setting is from high-contrast *simpler_basic* to dimly lit *basic* scenario, and the right one is from *defend_the_line* with a rectangular map to *defend_the_center* with a circular map.

The agent architecture for all methods consisted of a 4-layer convolutional neural network (ConvNet) with 3x3 kernels featuring 16 maps, complemented by ReLU activation functions. This was followed by a fully connected layer that outputs a distribution of action sizes.

To evaluate performance in terms of environmental change detection, we compared BADA to:

- A permutation test using KL divergence
- A two-sample test using weighted maximum mean discrepancy (WMMD) [Bellot and van der Schaar, 2021]
- The online parametric Dirichlet change point (ODCP) [Prabuchandran *et al.*, 2021]
- CRL-Unsup [Lomonaco *et al.*, 2020], which is based on long and short-term rewards.

Metrics. One metric is the cumulative reward or reward curve, and the other metric is F1 Score $F_1 = \frac{2 * P * R}{P + R}$ [Sasaki, 2007], indicating the detection accuracy.

4.2 Overall performance

Cumulative rewards. As Fig. 5 shows, BADA (depicted in red) exhibits an accelerated increase in reward after the change point (marked by the vertical dashed line in each graph). The post-change point improvement in reward is not only attributed to the effectiveness of adaptation regularization, enabling the policy to deviate from its previous optimum swiftly but also indicates that BADA accurately responds to environmental changes.

In *basic*, when the lighting and wall texture of the room change, methods without adaptation see a significant performance drop. We can see that the CRL-Unsup method demonstrates a notable adaptation ability with a steady increase in rewards after environmental changes are initially detected, albeit slightly inferior to BADA. This indicates that our behavior-based regularization term enables faster adaptation to new environments. In *health_gathering*, we can see that the texture of the medikit has a lesser impact than the lighting level. This is evident from the results, where even methods without adaptation can regain relatively high rewards after several updates. Meanwhile, BADA’s reward continues to rise, demonstrating its adaptation capability in environments with relatively minor changes. In this scenario, CRL-Unsup seems to be prone to false detections of environmental changes. This leads to unnecessary adaptations and results in a less stable learning process. Notably, in the *deadly_corridor*, the reduction in the number of enemies does not yield additional rewards for well-trained agents who do not employ adaptation strategies. This could

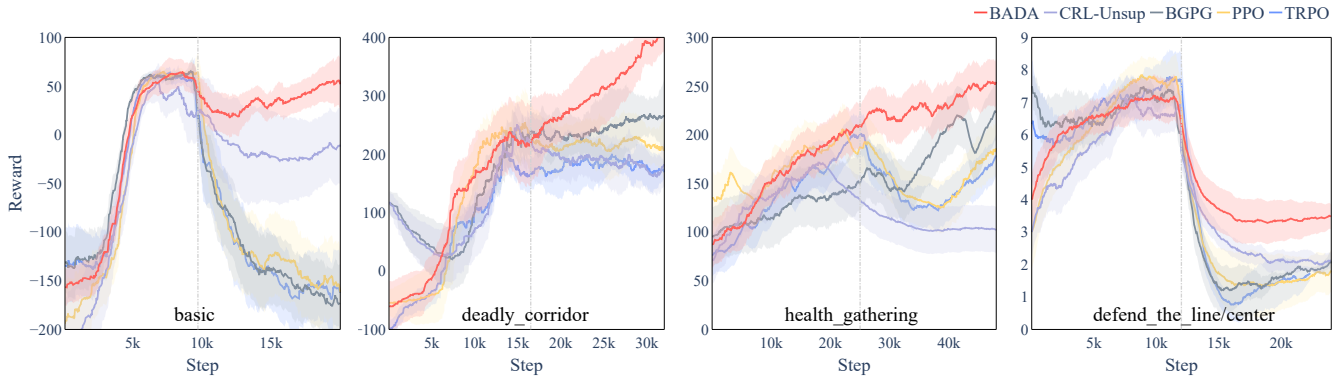


Figure 5: Performance comparison of different methods in non-stationary environments. The vertical dashed lines represent the points of environmental change, and the shaded areas around the reward lines indicate the standard deviation over different runs.

	<i>basic</i>	<i>health_gathering</i>
BADA(Ours)	0.95 ± 0.08	0.90 ± 0.11
Permu-KL	0.70 ± 0.09	0.50 ± 0.26
CRL-Unsup	0.80 ± 0.12	0.35 ± 0.16
WMMD	0.50 ± 0.13	0.56 ± 0.27
ODCP	0.55 ± 0.36	0.37 ± 0.07
	<i>deadly_corridor</i>	<i>defend_the_line</i>
BADA(Ours)	0.78 ± 0.16	0.86 ± 0.07
Permu-KL	0.69 ± 0.09	0.50 ± 0.26
CRL-Unsup	0.67 ± 0.21	0.72 ± 0.11
WMMD	0.47 ± 0.19	0.62 ± 0.15
ODCP	0.38 ± 0.20	0.50 ± 0.19

Table 1: Comparative F1 scores of change detection methods in non-stationary environments.

be attributed to the fact that they still follow their original behavior and fail to respond promptly to environmental changes. However, BADA reaches a higher reward faster and sustains an upward trend, showing its ability to learn from new environmental conditions continuously. In the scenarios of *defend_the_line/center*, map shape and defended goal changes force each method to learn a new task. In this context, BADA and CRL-Unsup outperform other baselines by quickly achieving higher scores on the new task, while BADA has superior performance compared to other methods. This indicates the effectiveness of steering away from the previous optimal strategy in discovering a new one.

Environment change detection accuracy. Tab. 1 lists the F1 scores for all the methods. As shown, BADA outperforms other methods in all scenarios. The lower accuracy of BADA in the *deadly_corridor* scenario is because of the reduced number of enemies, which has a less immediate impact on behavior compared to observable conditions like environmental lighting. The relatively poorer performance of the permutation test based on KL divergence (Permu-KL) compared to our Wasserstein-based approach can be attributed to the KL divergence being sensitive to the probability distribution’s representation in the data. In scenarios where the probability distributions of the environment states are sparse

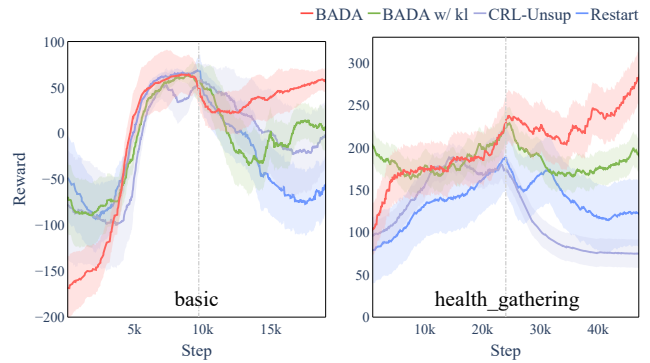


Figure 6: Cumulative rewards of adaptation strategies in non-stationary environments with known change points.

or have non-overlapping supports, KL divergence struggles to measure the distance between distributions accurately. By contrast, Wasserstein distance is based on the optimal transport problem, denoting the minimum “cost” of turning one distribution into the other. It is particularly beneficial in non-stationary reinforcement learning environments, which often feature abrupt and significant changes in distribution. Another observation is that CRL-Unsup performs relatively well but heavily relies on extensive tests to select the thresholds manually. By contrast, BADA detection does not require hyperparameters to be manually adjusted and provides a significance level at the same time. We also find that ODCP and WMMD are not efficient in image-based scenarios.

4.3 Ablation Study

Adaptation evaluation. To evaluate the performance of adaptation separately and confirm whether the adaptation scheme contributes to new training as opposed to simply re-training the agent following a reinforcement learning loss, we test the following comparison methods:

- Employing KL divergence instead of Wasserstein distance as the regularization term.
- CRL-Unsup with the EWC adaptation method.
- Restarting training following a traditional PPO scheme.

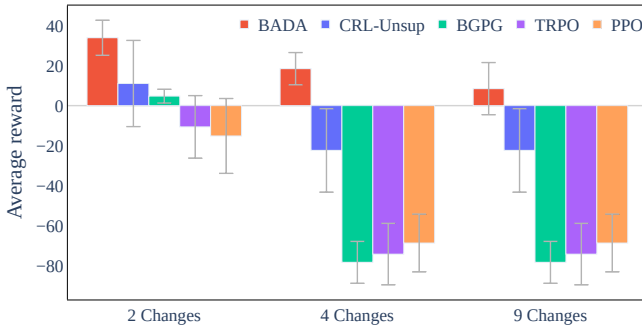


Figure 7: Average reward after the first change points in environments with increasing change points.

All methods are provided with the change points to initiate adaptation or retraining.

As shown in Fig. 6, the BADA method excels in adapting to environments with known change points. First, BADA consistently outperformed the ‘Restart’ approach, as seen by the quicker recovery and sustained improvement in rewards. This indicates that BADA deviates from the previous optimal and finds a new one rapidly, adapting a more efficient strategy than restarting training from scratch. Second, BADA surpassed the other adaptation strategies, with the Wasserstein Distance constraint proving superior to KL divergence. Further, BADA outperforms CRL-Unsup, demonstrating that our behavior-based adaptation is more effective than other methods. Overall, these results confirm that BADA has a superior ability to adapt to environmental changes.

Frequently changing environments. Change frequency could challenge BADA’s ability because it may affect whether the constrained distribution is the previous optimal, i.e., the policy might not have converged when the changes occurred. Fig. 7 provides a comparative overview of the different algorithms’ performances across the *basic* environment with varying numbers of change points. As indicated in red, BADA consistently achieved higher average rewards than the other methods when the environment changed in more frequently changing environments, demonstrating its robustness in dealing with multiple change points. We can see that when the number of change points increases from 2 to 4, performance does not drop significantly. However, as the number of change points increases to 9, the performance of all methods has marked declines. However, BADA shows the most minor decrease, maintaining a clear lead over the others. This indicates BADA’s superior adaptability in more complex environments with frequent changes. Also, Tab. 2 shows the detection accuracy in frequently changing environments. All methods will be influenced as the number of change points increases. Therefore, in extremely non-stationary environments, the policy may not have converged in each environment, resulting in behaviors that remain in random and chaotic patterns. This can limit BADA’s ability to detect and adapt based on behavior.

Parameter sensitivity. Fig. 8 shows the parameter sensitivity analysis for the adaptation regularization term $W(\mathbb{P}_\theta, \mathbb{P}_{pre})$ in Eq. (7). The result indicates a distinct peak in average re-

	2 changes	4 changes	9 changes
BADA(Ours)	0.89±0.12	0.78±0.16	0.56±0.20
Permu-KL	0.52±0.19	0.49±0.11	0.43±0.09
CRL-Unsup	0.71±0.13	0.60±0.17	0.37±0.08
WMMD	0.45±0.17	0.40±0.11	0.28±0.19
ODCP	0.25±0.15	0.21±0.11	0.20±0.13

Table 2: F1 scores for change detection methods across environments with increasing change points.

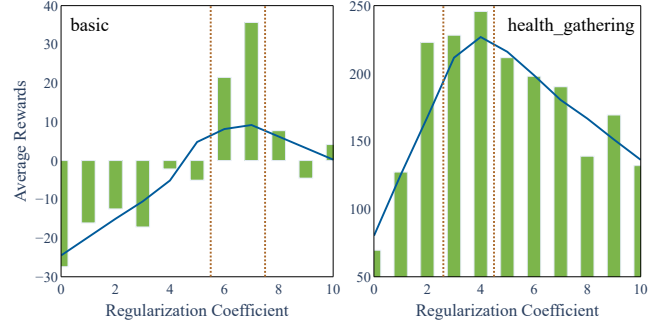


Figure 8: The parameter sensitivity analysis of the adaptation regularization. The orange lines represent the coefficient range we used.

wards for both environments. This peak represents an optimal value for the coefficient, consistent with the range of our adaptive coefficient (denoted in orange rectangles). The parameter we used represents the extent of environmental change, determining the level of adaptation based on the environment. The experiment proves that our self-adjusted coefficient, according to environmental changes, is effective. Also, the decline post-peak implies that an overly aggressive correction term could negatively impact the learning process. Therefore, an accurate balance of adaptation regularization is crucial to sustaining good performance. The empirical results show that tuning it according to change level is valid.

5 Conclusion

This paper addresses deep reinforcement learning in non-stationary environments without known change points by developing the Behavior-Aware Detection and Adaptation (BADA) framework. The behavior-based change detection method represents a novel approach to monitoring and responding to environmental shifts by closely analyzing policy behavior. This method has proven effective and accurate without any manually set threshold, allowing for timely adjustments to the learning strategy. Furthermore, the online adaptation mechanism integrates this behavioral information, providing a self-adjusted regularization term. The behavior-based regularization can help policy steer from suboptimal areas and find potential behavior in new conditions. The experimental results show its superior performance in accurately detecting changes and quickly adapting to new environments compared to other methods. A future extension could benefit from exploring mechanisms for off-policy adaptations, broadening BADA’s applicability in various RL settings.

Acknowledgements

This work is supported by the Australian Research Council under Australian Laureate Fellowships FL190100149 and Discovery Early Career Researcher Award DE200100245.

References

- [Aljundi *et al.*, 2017] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3366–3375, 2017.
- [Bellot and van der Schaar, 2021] Alexis Bellot and Mihaela van der Schaar. A kernel two-sample test with selection bias. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 205–214, 2021.
- [Bing *et al.*, 2023] Zhenshan Bing, David Lerch, Kai Huang, and Alois C. Knoll. Meta-reinforcement learning in non-stationary and dynamic environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(3):3476–3491, 2023.
- [Da Silva *et al.*, 2006] Bruno C Da Silva, Eduardo W Basso, Ana LC Bazzan, and Paulo M Engel. Dealing with non-stationary environments using context detection. In *International conference on Machine learning (ICML)*, pages 217–224, 2006.
- [Delarue *et al.*, 2020] Arthur Delarue, Ross Anderson, and Christian Tjandraatmadja. Reinforcement learning with combinatorial actions: An application to vehicle routing. *Advances in Neural Information Processing Systems*, 33:609–620, 2020.
- [D’Eramo *et al.*, 2020] Carlo D’Eramo, Davide Tateo, Andrea Bonarini, Marcello Restelli, Jan Peters, et al. Sharing knowledge in multi-task deep reinforcement learning. In *8th International Conference on Learning Representations, {ICLR} 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, pages 1–11. OpenReview. net, 2020.
- [Fifty *et al.*, 2021] Chris Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *Advances in Neural Information Processing Systems*, 34:27503–27516, 2021.
- [Good, 2013] Phillip Good. *Permutation tests: a practical guide to resampling methods for testing hypotheses*. Springer Science & Business Media, 2013.
- [Hadoux *et al.*, 2014] Emmanuel Hadoux, Aurélie Beynier, and Paul Weng. Sequential decision-making under non-stationary environments via sequential change-point detection. In *Learning over Multiple Contexts (LMCE)*, 2014.
- [Hu *et al.*, 2023] Mingzhe Hu, Jiahua Zhang, Luke Matkovic, Tian Liu, and Xiaofeng Yang. Reinforcement learning in medical image analysis: Concepts, applications, challenges, and future directions. *Journal of Applied Clinical Medical Physics*, 24(2):e13898, 2023.
- [Huang *et al.*, 2022] Biwei Huang, Fan Feng, Chaochao Lu, Sara Magliacane, and Kun Zhang. Adarl: What, where, and how to adapt in transfer reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022.
- [Kirkpatrick *et al.*, 2017] 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*, 114(13):3521–3526, 2017.
- [Liu *et al.*, 2022] Shikun Liu, Stephen James, Andrew Davidson, and Edward Johns. Auto-lambda: Disentangling dynamic task relationships. *Transactions on Machine Learning Research*, 2022.
- [Liu *et al.*, 2024] Ziheng Liu, Jie Lu, Junyu Xuan, and Guangquan Zhang. Deep reinforcement learning in non-stationary environments with unknown change points. *IEEE Transactions on Cybernetics*, 2024.
- [Lomonaco *et al.*, 2020] Vincenzo Lomonaco, Karan Desai, Eugenio Culurciello, and Davide Maltoni. Continual reinforcement learning in 3d non-stationary environments. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 248–249, 2020.
- [Lopez-Paz and Ranzato, 2017] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6467–6476, 2017.
- [Mosavi *et al.*, 2020] Amirhosein Mosavi, Yaser Faghan, Pedram Ghamisi, Puhong Duan, Sina Faizollahzadeh Ardabili, Ely Salwana, and Shahab S Band. Comprehensive review of deep reinforcement learning methods and applications in economics. *Mathematics*, 8(10):1640, 2020.
- [Oikonomou *et al.*, 2023] Katerina Maria Oikonomou, Ioannis Kansizoglou, and Antonios Gasteratos. A hybrid spiking neural network reinforcement learning agent for energy-efficient object manipulation. *Machines*, 11(2):162, 2023.
- [Olkin and Pukelsheim, 1982] Ingram Olkin and Friedrich Pukelsheim. The distance between two random vectors with given dispersion matrices. *Linear Algebra and its Applications*, 48:257–263, 1982.
- [Pacchiano *et al.*, 2020] Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, Krzysztof Choromanski, Anna Choromanska, and Michael Jordan. Learning to score behaviors for guided policy optimization. In *International Conference on Machine Learning*, pages 7445–7454, 2020.
- [Padakandla *et al.*, 2020] Sindhu Padakandla, KJ Prabuchandran, and Shalabh Bhatnagar. Reinforcement learning algorithm for non-stationary environments. *Applied Intelligence*, 50(11):3590–3606, 2020.
- [Panaretos and Zemel, 2019] Victor M Panaretos and Yoav Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6:405–431, 2019.

- [Prabuchandran *et al.*, 2021] KJ Prabuchandran, Nitin Singh, Pankaj Dayama, Ashutosh Agarwal, and Vinayaka Pandit. Change point detection for compositional multivariate data. *Applied Intelligence*, pages 1–26, 2021.
- [Rakelly *et al.*, 2019] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning (ICML)*, pages 5331–5340, 2019.
- [Sasaki and Yamashina, 2020] Fumihiko Sasaki and Ryota Yamashina. Behavioral cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2020.
- [Sasaki, 2007] Yutaka Sasaki. The truth of the f-measure. *Teach tutor mater*, 1(5):1–5, 2007.
- [Schulman *et al.*, 2015] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.
- [Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [Schwarz *et al.*, 2018] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning (ICML)*, pages 4528–4537, 2018.
- [Shin *et al.*, 2017] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2990–2999, 2017.
- [Standley *et al.*, 2020] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *International Conference on Machine Learning*, pages 9120–9132. PMLR, 2020.
- [Tiwari *et al.*, 2023] Prayag Tiwari, Abdullah Lakhani, Rutvij H Jhaveri, and Tor-Morten Gronli. Consumer-centric internet of medical things for cyborg applications based on federated reinforcement learning. *IEEE Transactions on Consumer Electronics*, 2023.
- [Trabucco *et al.*, 2022] Brandon Trabucco, Mariano Phielipp, and Glen Berseth. Anymorph: Learning transferable policies by inferring agent morphology. In *International Conference on Machine Learning*, pages 21677–21691. PMLR, 2022.
- [Van Borkulo *et al.*, 2022] Claudia D Van Borkulo, Riet van Bork, Lynn Boschloo, Jolanda J Kossakowski, Pia Tio, Robert A Schoevers, Denny Borsboom, and Lourens J Waldorp. Comparing network structures on three aspects: A permutation test. *Psychological methods*, 2022.
- [Villani, 2009] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.
- [Welch, 1990] William J Welch. Construction of permutation tests. *Journal of the American Statistical Association*, 85(411):693–698, 1990.
- [Wydmuch *et al.*, 2019] Marek Wydmuch, Michał Kempka, and Wojciech Jaśkowski. ViZDoom Competitions: Playing Doom from Pixels. *IEEE Transactions on Games*, 11(3):248–259, 2019. The 2022 IEEE Transactions on Games Outstanding Paper Award.
- [Xie *et al.*, 2021] Annie Xie, James Harrison, and Chelsea Finn. Deep reinforcement learning amidst continual structured non-stationarity. In *International Conference on Machine Learning (ICML)*, pages 11393–11403, 2021.
- [Xu, 2019] Lihu Xu. Approximation of stable law in wasserstein-1 distance by stein’s method. *The Annals of Applied Probability*, 29(1):458–504, 2019.
- [Yu *et al.*, 2020] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *The Conference on Robot Learning (CoRL)*, pages 1094–1100, 2020.
- [Yu *et al.*, 2021] Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Sergey Levine, and Chelsea Finn. Conservative data sharing for multi-task offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:11501–11516, 2021.
- [Yu *et al.*, 2022] Tianhe Yu, Aviral Kumar, Yevgen Chebotar, Karol Hausman, Chelsea Finn, and Sergey Levine. How to leverage unlabeled data in offline reinforcement learning. In *International Conference on Machine Learning*, pages 25611–25635. PMLR, 2022.
- [Zenke *et al.*, 2017] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*, pages 3987–3995, 2017.
- [Zintgraf *et al.*, 2019] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR, 2019.