Bio-inspired meta-learning for active exploration during non-stationary multi-armed bandit tasks

George Velentzas, Costas Tzafestas, Mehdi Khamassi

Bio–inspired meta–learning for active exploration during non–stationary multi–armed bandit tasks

George Velentzas, Costas Tzafestas
School of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
Email: geovelentzas@gmail.com, ktzaf@cs.ntua.gr

Mehdi Khamassi
Institute of Intelligent Systems and Robotics
Sorbonne Universités, UPMC Univ Paris 06, CNRS
F-75005 Paris, France
Email: mehdi.khamassi@upmc.fr

***Abstract*—Fast adaptation to changes in the environment requires agents (animals, robots and simulated artefacts) to be able to dynamically tune an exploration-exploitation trade-off during learning. This trade-off usually determines a fixed proportion of exploitative choices (i.e. choice of the action that subjectively appears as best at a given moment) relative to exploratory choices (i.e. testing other actions that now appear worst but may turn out promising later). Rather than using a fixed proportion, non-stationary multi-armed bandit methods in the field of machine learning have proven that principles such as exploring actions that have not been tested for a long time can lead to performance closer to optimal – bounded regret. In parallel, researches in active exploration in the fields of robot learning and computational neuroscience of learning and decision-making have proposed alternative solutions such as transiently increasing exploration in response to drops in average performance, or attributing exploration bonuses specifically to actions associated with high uncertainty in order to gain information when choosing them. In this work, we compare different methods from machine learning, computational neuroscience and robot learning on a set of non-stationary stochastic multi-armed bandit tasks: abrupt shifts; best bandit becomes worst one and vice versa; multiple shifting frequencies. We find that different methods are appropriate in different scenarios. We propose a new hybrid method combining bio-inspired meta-learning, kalman filter and exploration bonuses and show that it outperforms other methods in these scenarios.**

***Index Terms*—Bandits; Decision Making; meta-learning; Active exploration; kalman filter; reinforcement learning; multi-armed bandit**

I. Introduction

Optimal action selection from a number of distinctive alternatives in unknown environments, is not only a subject of Game theory, but has thoroughly being studied in the Machine Learning and Artificial Intelligence fields in general. Its recent recapturing of attention is particularly due to its characteristic as a keystone in reinforcement learning research: The former can be seen as a special case of the latter where a single decision step is required to get feedback from the environment. Sophisticated trial and error strategies have been developed, from genetic algorithms [1], to entropy minimization techniques [2], which have led the scientific community to establish fundamental benchmarks for proper evaluation of decision making agents. Stochastic multi-armed bandits constitute such a benchmark, where an agent chooses an action from a number of discrete choices at each time-step of its lifespan, receives a reward with some stochasticity or noise, and then feedbacks its base of knowledge in order to re-establish its policy of action selection while appropriately handling exploration versus exploitation trade-off [3]. A concrete real-life example can be considered when a human has to choose between two coffee machines at his office, both delivering the same type of coffee, but each machine having a different degree of reliability. The taste or the quantity of coffee fluctuates from day to day in both machines, one of them being on average a little bit better than the other one. The human can not evaluate the machine based on a single shot, but needs to learn by trial and error before sticking to (exploiting) his preferred machine.

Stationary scenarios have been studied for more than a decade. The EXP3 algorithm [4] enhanced the use of Boltzman softmax function in order to achieve optimal regret – the regret being defined as the difference between optimal performance and accumulated reward. A Bayesian approach proposed in [5] was shown to be also optimal, and the family of Upper-Confidence-Bound (UCB) algorithms presented in [3], using *optimism under the face of uncertainty*, constitute a simple and powerful solution. The limits of efficiency for the above algorithms though is that they remain in the frame of *stationary* environments, for as shown in [6] an algorithm that achieves optimal regret in stationary cases is lower bounded by $T/logT$ in non-stationary ones.

Non-stationary environments can be either *drifting* or *abrupt*, with the later comprising a more challenging problem, since the dynamics of variations in the history of rewards can not be easily used for predictions of future optimal policy changes. Back to our real-life example, one of the coffee machines may have been repaired or even upgraded during the night, so that our human subject needs to discover by himself that it is now better than the other machine. This is impossible mission if he had previously decided that he prefers the other machine and never tries the one which is now repaired (i.e. zero exploration). In [6], *Discounted-UCB (D-UCB)* (firstly presented in [7]) and *Sliding Window-UCB (SW-UCB)* have been analyzed and analytically proven to achieve a sufficient upper bound of regret, upon proper parameter choice, resulting from a prior knowledge of the environment dynamics – the number or rate of optimal arm changes, as well as the

time horizon. The *Kalman Filter-Multi Armed Normal Bandit (KF-MANB)* algorithm proposed in [8] constitutes a Bayesian approach which also indicated promising results with an elegant and intuitive implementation. Finally, the *Adapt-EvE* algorithm has shown a great performance during empirical validation at the PASCAL-EvE 2006 challenge, using a Page Hinkley statistics oriented change point detector, and a UCB-tuned algorithm as decision maker.

In the fields of robot learning [9]–[11] and computational neuroscience of learning and decision-making [12]–[16], researchers classically tackle the exploration-exploitation trade off with a Boltzmann softmax function, sometimes including exploration bonuses for specific actions [17], [18]. This has provided evidence of great performance, yet several limitations can be considered. In cases with constant inverse temperature parameter β – controlling the exploration rate in the softmax function – or constant increase of β, the exploration level is often not sufficient for quickly detecting environmental changes in non-stationary scenarios. In [19], a biologically plausible method was proposed for adaptation of meta parameters with meta-learning in reinforcement learning, relating the feedback component as the phasic and tonic components of dopamine neurons firing. In the same work, it was proposed that the use of inverse temperature as a meta-parameter, properly adapted at each time step based on local variations of reward running averages, can lead to efficient performance in non stationary reinforcement learning scenarios. Here we provide empirical evidence, that a simple update rule for inverse temperature meta-parameter in Boltzmann softmax exploration function, can lead to a meta-learning algorithm for bandits that exhibits a good performance in many cases, yet suffers from large regret at cases when a non optimal arm becomes optimal without other alterations of the environment.

We then propose a hybrid algorithm that incorporates both meta-learning for bandits and sibling kalman filters of KF-MANB, testing its empirical performance on a set of different non-stationary bandit setups where we variate the most crucial components of a stochastic and changing environment. We tune the parameters of each algorithm for a mid-case, and compare their efficiency in terms of cumulative regret.

Section II describes related work and algorithms as also a simplistic implementation of meta learning for bandits. Section III describes the hybrid algorithm MLB-KF which incorporates meta-learning for bandits and KF-MANB. In section IV we investigate different setups, providing simulations of experiments and results. In section V we make the total evaluation and discussion, while section VI concludes the paper.

II. Problem Formulation and Related Work

Stochastic multi armed-bandits can be considered as having a set of arms $\mathcal{K} = \{1, 2, ..., K\}$ of a casino machine, where a decision maker chooses an action $a \in \mathcal{K}$ at every timestep $t \in \mathcal{T} = \{1, 2, ..., T\}$ of its lifespan T (also called the time horizon). He then receives a reward $r_t(a)$ with probability $p_t(a)$ and zero otherwise. While interested in arm's expected value of reward, without loss of generality, we can assume Bernoulli arms where $\forall(a, t) \in \mathcal{K} \times \mathcal{T}$ the rewards are binary, $r_t(a) \in \{0, 1\}$. By choosing Bernoulli arms, the expected value $\mathbb{E}[r_t(a)]$ for every arm reward is equal to the probability $p_t(a)$. When the environment is stationary, $p_{t+1}(a) = p_t(a)$ for all time steps. For non-stationary environments the above rule does not stand. Specifically in drifting environments $|p_{t+1}(a) - p_t(a)| < \epsilon$, where ϵ is a small constant, while in abruptly changing environments $\exists t : |p_{t+1}(a) - p_t(a)| > \delta$, where δ is a sufficiently large value in terms of probabilities. Both ϵ and δ can be used as a measure of the environment dynamics.

Denoting a^* the optimal arm to choose, the regret at every time step is then defined as $\mathbb{E}[R_t(a^*) - R_t(a)]$ (with R_t denoting the random variables of rewards), and the total cumulative regret is

$$TR = \sum_{t=1}^{T} \mathbb{E}[R_t(a^*) - R_t(a)] = \sum_{t=1}^{T} (p_t(a^*) - p_t(a)) \quad (1)$$

which constitutes of a measure of evaluation, with lower total regret denoting better decision makers. For stationary environments the total regret is lower bounded by $O(logT)$, though an algorithm that achieves optimal regret in these environments cannot achieve a total regret lower than $T/logT$ in abruptly changing environments as shown in [6]. The above formulation can be used for testing where the evaluator has prior knowledge of the arm probabilities (while the decision maker does not). The above will also be used for evaluation of all algorithms on our test cases of the experiments section.

A. Discounted UCB

D-UCB, proposed in [7] and mathematically analyzed in [6], has been shown to achieve efficient empirical performance by using a discount factor γ for calculating the discounted per arm reward. UCB algorithms make use of Hoeffding inequality to produce an exploration bonus for each arm, that can be added to the discounted per arm reward make estimations in the face of uncertainty. The actions are then taken with respect to the arm that has the potential to be optimal, rather than the currently best arm based on the empirical discounted average. The value $n_t(a)$, describing the number of times that each arm has been chosen until time step t, is also discounted with γ parameter, therefore

$$n_t(a; \gamma) = \sum_{s=1}^{t} \gamma^{t-s} \mathbb{1}_{\{a_s = a\}} \quad (2)$$

where $\mathbb{1}_{\{.\}}$ denotes the indicator function. The average discounted reward $\overline{r}_t(a)$ for each arm is therefore defined as

$$\overline{r}_t(a; \gamma) = \frac{1}{n_t(a; \gamma)} \sum_{s=1}^{t} \gamma^{t-s} r_s(a) \mathbb{1}_{\{a_s = a\}} \quad (3)$$

and the padding function or exploration bonus $c_t(a)$ for each arm is defined by

$$c_t(a;\gamma,\xi) = 2B\sqrt{\xi \log(\sum_{a=1}^{K} n_t(a))/n_t(a)} \tag{4}$$

where B denotes an upper bound for rewards and ξ a parameter. The decision maker takes an action at each time step according to the rule $a_t = \arg\max_a (\overline{r}_t(a) + c_t(a))$, receives a reward and updates the above values. Initialization also has to take place by choosing every arm once at the beginning of each session.

B. Sliding Window UCB

SW-UCB, is similar to D-UCB in that they belong to the same family of algorithms proposed in [3]. Instead of computing the discounted per arm reward, it uses a history of the τ most recent rewards and actions taken. The number $n_t(a)$ that each arm has been chosen with only the sliding window as memory is

$$n_t(a;\tau) = \sum_{s=t-\tau+1}^{t} \mathbb{1}_{\{a_s=a\}} \tag{5}$$

adjusted appropriately when $t < \tau$. The average per arm reward, using the sliding window of width τ, is then

$$\overline{r}_t(a;\tau) = \frac{1}{n_t(a;\tau)} \sum_{s=t-\tau+1}^{t} r_s(a)\mathbb{1}_{\{a_s=a\}} \tag{6}$$

and the exploration bonus has to take in mind the transitional phase where the window width is larger than the current time step, so

$$c_t(a;\tau,\xi) = B\sqrt{\frac{\xi}{n_t(a)} \log(min(t,\tau))} \tag{7}$$

with the decision maker following the same rule as in D-UCB, that is $a_t = \arg\max_a (\overline{r}_t(a) + c_t(a))$. Initialization has also to take place by pulling each arm once.

C. Kalman Filter – Multi Armed Normal Bandit

With the use of sibling Kalman Filters, KF-MANB proposed in [8] has shown evidence of satisfactory robustness and performance for both stationary and non-stationary environments. Each arm can be modeled with a normal distribution of mean $\mu_t(a)$ and variance $\sigma_t^2(a)$. It uses two parameters, σ_{ob}^2 and σ_{tr}^2, which relate to the stochastic part and the non-stationary part of the environment respectively. It starts with an initial set of values for means $\mu_1(a)$ and variances $\sigma_1^2(a)$ of each arm $a \in \mathcal{K}$, takes a sample s_a from the respective distribution and makes the action choice according to the rule $a_t = \arg\max_a(s_a)$. The reward $r_t(a)$, for $a = a_t$, is then used to update the distribution of the respective arm with

$$\mu_{t+1}(a) = \frac{(\sigma_t^2(a) + \sigma_{tr}^2) r_t(a) + \sigma_{ob}^2 \mu_t(a)}{\sigma_t^2(a) + \sigma_{tr}^2 + \sigma_{ob}^2} \tag{8}$$

$$\sigma_{t+1}^2(a) = \frac{(\sigma_t^2(a) + \sigma_{tr}^2)\sigma_{ob}^2}{\sigma_t^2(a) + \sigma_{tr}^2 + \sigma_{ob}^2} \tag{9}$$

while for arms $a \neq a_t$ the means $\mu_{t+1}(a)$ maintain their previous value, and a transitional variance is added to $\sigma_{t+1}^2(a)$ in order to enhance exploration of non-chosen arms as shown below

$$\mu_{t+1}(a) = \mu_t(a) \text{ and } \sigma_{t+1}^2(a) = \sigma_t^2(a) + \sigma_{tr}^2 \tag{10}$$

From (9) it can be deduced that the variance of the chosen arm decreases, as $\sigma_{t+1}^2(a) < \min\{(\sigma_t^2(a) + \sigma_{tr}^2), \sigma_{ob}^2\}$, while a low-pass filter in (8) updates the mean with an adaptive learning rate.

D. Adapt-EvE

Adapt-EvE with meta bandits, proposed in [20], makes use of a change point detector based on Page Hinkley statistics. It starts with DUCB-tuned (DUCBT) algorithm (or simply UCB-tuned; UCBT) which adaptively changes ξ parameter by using an upper bound of variances for rewards of each arm. DUCBT replaces the exploration bonus of Eq.4 with

$$c_t(a;\gamma,\xi) = \sqrt{\frac{2}{n_t(a)} \log(\sum_{a=1}^{K} n_t(a)) \min\{1/4, Var(R_t(a))\}} \tag{11}$$

where $Var(R_t(a))$ denotes an empirical upper bound of variance for the rewards of arm a. For the Page Hinkley change point detector, the average reward $\overline{r}_t$ is used at every time step to compute $m_t = \sum_{s=1}^{t}(r_s - \overline{r}_s + \delta)$ (where δ is a tolerance parameter). Then $M_t = \max\{m_1, m_2, ..., m_t\}$ and the value $PH_t = M_t - m_t$ is finally compared with a threshold λ. If PH_t is greater than λ, a change point is detected. Instead of reseting the history of DUCBT, a second bandit is then created and initialized. Also a meta-bandit, either using DUCBT or UCBT algorithm (with the former denoting a meta-ρ-bandit), makes a decision on which bandit should decide for the next action, the new or the old one. After t_m time steps, the bandit that has been chosen mostly by meta-bandit, is the only one to remain. The above procedure decreases type I errors (defined as incorrect change point detections), although when being in this meta-bandit phase PH detector does not monitor the rewards, therefore no new change points can be detected.

E. Meta-learning for Bandits - MLB

In [21] was proposed that the difference between mid-term rewards and long-term punishments can be used to explain the variations of negative and positive affects in nature. In [19] this idea was implemented with the dynamic tuning of meta-parameters in reinforcement learning, to properly adjust the exploration-exploitation trade-off in non-stationary environments. Here we present a modification of meta-reinforcement learning, for proper use on a bandit problem (MLB). The main

idea remains to tune the inverse temperature of the softmax Boltzmann function for re-engaging exploration by decreasing its value appropriately. In short, increases in performance (as measured by relative variations of short- and long-term reward running averages) lead to increases of the exploitation of learned arm values, so that the agent can reach a nearly optimal performance. Conversely, drops in the average reward can be interpreted as signs of a change in the task conditions and thus as a need to re-explore.

With β_t denoting the inverse temperature meta-parameter, and $Q_t(a)$ the action value of arm a, the probability of pulling arm a at each time step t is

$$P(a|\beta_t) = \frac{exp(\beta_t Q(a))}{\sum_{a \in \mathcal{K}} exp(\beta_t Q(a))} \tag{12}$$

The immediate reward r_t received from action $a = a_t$, is then used to update the mid-term reward $\overline{r}_t$, and the long-term reward $\overline{\overline{r}}_t$,

$$\overline{r}_t = \alpha_m r_t + (1 - \alpha_m)\overline{r}_{t-1} \tag{13}$$

$$\overline{\overline{r}}_t = \alpha_\ell \overline{r}_t + (1 - \alpha_\ell)\overline{\overline{r}}_{t-1} \tag{14}$$

where α_m and α_ℓ are learning rates, inversely proportional to two time constants τ_1 and τ_2 respectively, as defined in [19]. The action value $Q(a)$ of the pulled arm, and the inverse temperature meta-parameter β_t are then updated as follows

$$Q_{t+1}(a) = (1 - \alpha_Q)Q_t(a) + \alpha_Q r_t \tag{15}$$

$$\beta_{t+1} = max\{\beta_t + \eta(\overline{r}_t - \overline{\overline{r}}_t)), 0\} \tag{16}$$

where η is a parameter of choice and α_Q a learning rate.

In [19] the proposition was to also use α_Q as a dynamically tuned meta-parameter for reinforcement learning approaches. However, here we explore how the simple biologically plausible modification to the inverse temperature, making the use of softmax Boltzmann comparable and even better in some cases, with the state-of-the art algorithms for non-stationary environments. With the above update rule of (16), β_t increases when the mid-term reward is greater than the long-term. One can view this as an evidence that the recent actions are more optimal, therefore exploitation may be increased. In the opposite case when $\overline{r}_t < \overline{\overline{r}}_t$, the recent actions denote that the present policy has lower performance in getting rewards compared to the past, therefore exploration should be encouraged by reducing β_t.

III. Hybrid Meta learning with Kalman Filters

Here we propose a new algorithm as a hybrid model that integrates the sibling kalman filters of KF-MANB and the core of MLB. With the description of our modification of meta-learning for bandits, this new algorithm can now be described in a straight forward way, with a simple substitution. Following a proposition of computational neuroscience exploration models [17], [18], the action values $Q(a)$ used in the softmax function for decision-making in (12) are complemented by an arm-specific exploration bonus proportional to the uncertainty associated with each arm. To do so, action values $Q(a)$ are simply replaced with a trivial linear combination of the mean and the standard deviation of each arm's distribution, according to the sibling kalman filters of KF-MANB as follows

$$Q_t(a) = \mu_t(a) + \phi\sigma_t(a) \tag{17}$$

Therefore, the probability an arm a to be chosen at time step t, given the updated inverse temperature β_t is

$$P(a|\beta_t) = \frac{exp(\beta_t(\mu_t(a) + \phi\sigma_t(a)))}{\sum_{a \in \mathcal{K}} exp(\beta_t(\mu_t(a) + \phi\sigma_t(a)))} \tag{18}$$

where $\mu_t(a)$ and $\sigma_t(a)$ follow the update rules of KF-MANB and ϕ is a constant of choice. This algorithmic procedure is summarized in Algorithm 1.

Algorithm 1 MLB-KF

```
Choose parameters α_Q, α_m, α_ℓ, η, φ
Initialize β_1, and μ_1(a), σ_1(a) ∀a ∈ K
for t = 1, 2, ..., T do
    select action a_t ∈ K from distribution of (18)
    observe the immediate reward r_t
    update r̄_t, r̿_t with (13), (14)
    for a = a_t update μ_{t+1}(a), σ²_{t+1}(a) with (8), (9)
    for all arms a ≠ a_t update μ_{t+1}(a), σ²_{t+1}(a) with (10)
    update β_{t+1} with (16)
end for
```

The idea of adding an exploration bonus to each arm with the use of an uncertainty measure added to the exponential argument of the softmax function is not new. When the Gaussian distribution of an arm has high variance, the respective arm should be explored more. However when the environment suggests that no evidence of change is observed, KF-MANB suffers from exploration since a transitional variance is always added to the non-pulled arms. Meta-learning is also expected to display large regret at cases where a non-optimal arm becomes optimal after long periods of stationarity, since the "rising" arm may never be chosen, also due to floating point computational restrictions which affect the large values on the computations in exponentials. The above implementation is a trade off between those two cases, as we empirically present evidence that it incorporates both the benefits of meta-learning fast adaptation and small regret on many test cases, as also the

robustness and great performance of KF-MANB (observed in the experiments section of the next pages).

IV. Experiments and Results

Regarding experimental non-stationary setups for bandits, different features have to be taken into consideration. For abruptly changing environments, the rate at which switchings take place is one of the most important issues. Here we will have a small horizon of 10000 time steps, where the optimal arm will change from 1 to 10 times during this lifespan (approximately). Comparing with other existing setups, one can consider this to be a fast changing environment, however we believe that in many real life situations, one time step (also called episode) can be related to one day (or a few actions like advertisement placements), therefore the need to find fast adaptive algorithms, with adaptation to be efficiently performed in a few hundreds or thousands of timesteps in such cases is necessary.

Another major issue is the type of switching, which can take place either *globally* (i.e at some time step all arm probabilities change), or *per arm*, (i.e the probability $p_t(a)$ of each arm may change independently of others). The difference Δ_g between the expected reward of the optimal arm and the second best arm is also another feature, with which an equivalent of resolution limit measure of the decision maker can be evaluated. Special cases where the optimal arm becomes sub-optimal after long periods of stationarity, as also cases where the worst arm becomes the optimal one while all other arms remain stationary in terms of rewards, are a few more cases to explore. Usually, in most simulations presented in the literature, algorithms are tuned appropriately before testing by selecting the best parameters for the test case. This may present evidence of the optimal behavior, however the environmental stochasticity should include variations of all the above features. Here we tune the parameters of each algorithm to a specific non-stationary stochastic setup with 5 arms, evaluate their performance on this setup, and test on different setups with variations of all features, exploring robustness and performance in terms of total cumulative regret.

In problem set 1, we investigate the performances while changing the arm probabilities globally with a random walk, altering the rate of change points. In problem set 2 we alter

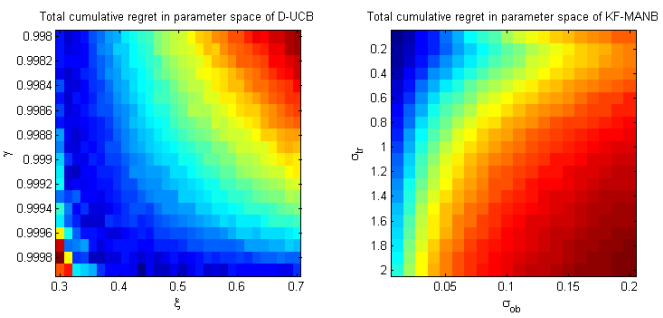

Fig. 1. left: total cumulative regret in parameter space of D-UCB, right: total cumulative regret in parameter space of KF-MANB

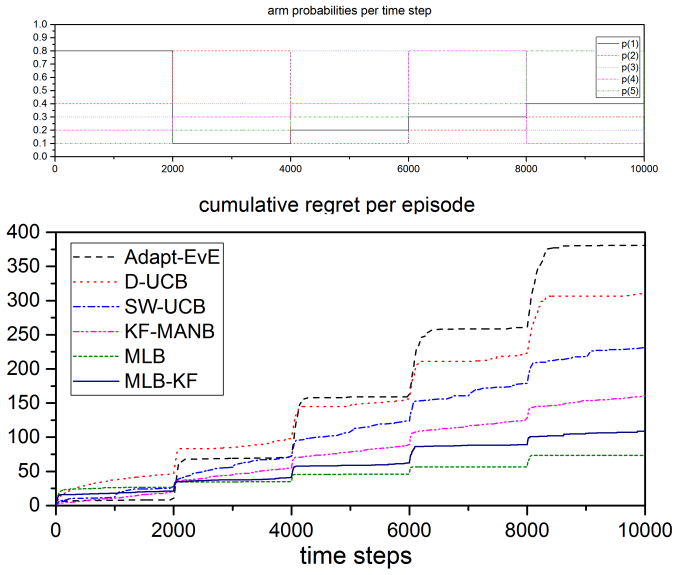

Fig. 2. top: the probabilities per arm at each time step, bottom: the averaged cumulative regret for each algorithm for the specific test case

the gap Δ_g between the optimal and second best arm, as also the number of change points with global switchings, uniformly distributed at fixed time steps. In problem set 3 we investigate performances with simple arm switches between best and worst expected value, keeping all other arms stationary.

A. Parameter Tuning

Parameter tuning is usually one of the most time consuming procedures before the evaluation of algorithms. For D-UCB and SW-UCB Garivier et al. in [6] provide close forms for computing the parameters in order to assure an upper bound of total cumulative regret. Nevertheless prior knowledge regarding the number of change points and the time horizon is needed. These parameters do not ensure the best performance but only an upper bound of regret. In order to evaluate all algorithms in an empirically fair way, we find here the optimal parameters in terms of empirical performance (i.e minimum average total cumulative regret).

The setup for parameter optimization can be considered as a mid-case scenario, yet there is no metric for such an estimate. We consider 5 arms, with initial arm probabilities of 0.8, 0.4, 0.3, 0.2, 0.1 for each arm respectively, with a circular shift of those at every 2000 time steps as can be seen in Fig.2. We run 200 sessions, calculated the mean total cumulative regret on a sparse parameter grid space and rerun for areas with the best observed performance. For Adapt-EvE we had to act differently due to its nature. We changed the Page Hinkley statistics parameters, δ and λ, observed the change points detected by adding a "1" to a sequence of zeros, at the respected time step. We repeated for 10 sessions and smoothed the averaged binary outcome. We then computed the correlation coefficients between the outcomes, and the true binary change points sequence signal (also smoothed by a

Hanning window). Hence we found areas of the parameter space where these coefficients had the largest value. We used UCBT as a basic decision maker for Adapt-EvE. The window size of the meta-bandit's lifespan t_m was then empirically chosen after observing simulation results. In Fig.1 the cumulative regret in parameter space of D-UCB and KF-MANB can be seen as an example. The other algorithms have similar parameter spaces, although the computational complexity rises exponentially with the increase of parameters (referred to as the *curse of dimentionality*).

For D-UCB the parameters chosen were $\xi = 0.22$ and $\gamma = 0.9999$, for SW-UCB $\xi = 0.3$ and $\tau = 998$, for Adapt-EvE $\delta = 0.13$, $\lambda = 40$, $t_m = 50$ and used UCBT as decision makers and as meta-bandit. For KF-MANB $\sigma_{ob} = 0.2$, $\sigma_{tr} = 0.01$, initializing all means and variances to 0.5. For MLB $\alpha_Q = 0.14$, $\alpha_m = 1/15$, $\alpha_\ell = 1/350$, $\eta = 0.44$. For MLB-KF we kept the parameters found for MLB and KF-MANB and only tuned $\phi = 1.5$. All the parameters were kept as above for all problem sets. From the average cumulative regret for each algorithm, as seen in Fig.2 it is deduced that MLB achieved the best performance, by demonstrating both exploitative behavior when needed (deduced from the flat horizontal regret), as also adjusted exploration at change points. KF-MANB was the third best algorithm, as MLB-KF demonstrated a performance in-between the former two. SW-UCB also suffered from exploration even after stationarity was settled (since it keeps only a window of memory). D-UCB suffered from inertia at switch points but achieved a flat behavior after learning the optimal arm. Adapt-EvE suffered from regret of the meta-bandit phase which is required for its implementation. Adapt-EvE demonstrated small regret after adaptation though, which is the main reason that it has been shown to be powerful in non-stationary environments with lager intervals of stationarity than here. In terms of variances (which are not shown in the figures) D-UCB and Adapt-EvE had larger variances than all others, while KF-MANB

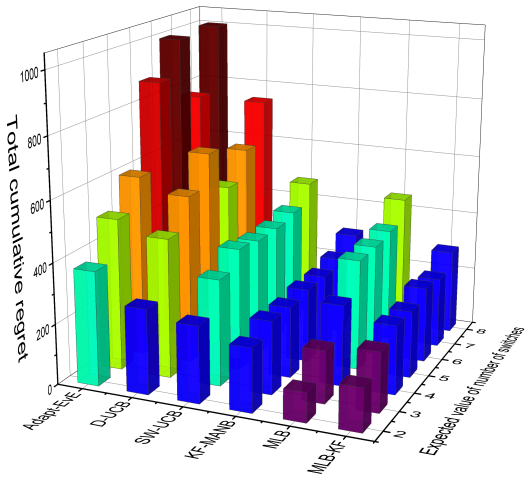

Fig. 3. total regret for different values of expected number of change points

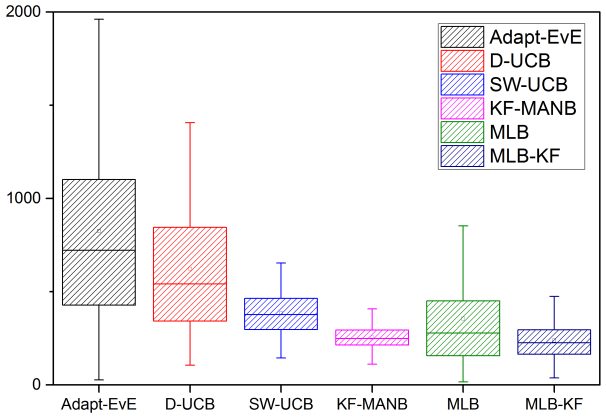

Fig. 4. box plots of total regret for all values of change point rates tested

achieved the smallest value and MLB-KF the second best. MLB had many outliers, while MLB-KF had not. MLB-KF demonstrated evidence that it inherited both the robustness of KF-MANB, as also the average performance of MLB.

B. Problem Set I

In this problem set, a global change point cp, may occur at every time step t with a constant probability h (i.e $p_t(cp) = h$). When a change point occurs, all arm probabilities are then re-sampled from a uniform distribution in [0, 1]. For each subproblem (using only the set of probabilities generated for each arm with the above type of random walk), we run each algorithm for 20 sub-sessions, then regenerated another problem set with the same probability h, and repeated the procedure for 100 hyper-sessions. We then calculated the average total cumulative regret with the assumption that the results are representative due to central limit theorem. We increased h and repeated all the above, for $h \in [2/10000, 8/10000]$ with a step of 1/10000. Therefore, an average of 2 to 8 global change points occur (we have an horizon of 10000) for every step.

From the results shown in Fig.3, it was observed that MLB had the best average performance for small expected number of change points, while MLB-KF was the second best. When the rate increased, MLB suffered from high variance (not shown here) which also increased its regret. KF-MANB was very robust at all cases, denoting on average a very good performance for all values of h tested. MLB-KF inherited both the good behavior of MLB for low switching rates, as also the robustness and performance of KF-MANB for higher rates, with even a slightly better performance. Adapt-EvE (using the parameters as chosen and with UCBT as basic decision agent), demonstrated large regret values, as also variance and dramatically reduced its performance for high switching rates. This was also the case with D-UCB, due to inertia created from past decisions and rewards. SW-UCB had an overall good and stable average performance at all cases, though from the simulations it is clear that MLB-KF exhibited the best performance between all of the algorithms, as also shown in

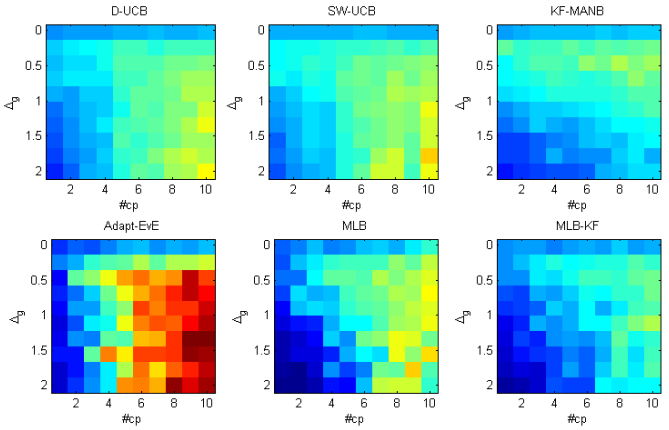

Fig. 5. top left: Total cumulative regret of D-UCB for different gaps Δ_g and number of switches $\#cp$, top middle: SW-UCB, top right: KF-MANB, bottom left: Adapt-EvE, bottom middle: MLB, bottom right: MLB-KF. The blue areas denote small values of regret, while red areas denote high values. All figures have the same reference values for proper comparison.

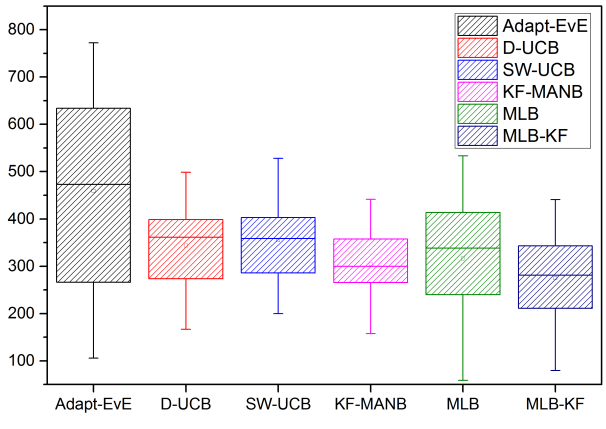

Fig. 6. averaged total regret from problem sets with different gaps Δ_g and number of change points $\#cp$

the box plots of Fig.4, where the distributions of total regret for all changing point rates can be observed.

C. Problem Set II

In these scenarios, we evaluated the performance of all algorithms, by altering the gap Δ_g, which denotes the difference between the expected reward of the optimal arm and the second best arm, while also circularly changing the arm probabilities at fixed time steps. The best optimal arm always had a probability of 0.8, the second best $0.8 - \Delta_g$, the third $0.8 - 2\Delta_g$ and so forth. For our experiments we begun with $\#cp = 1$ (where $\#cp$ denotes the number of change points). This change point occurred at $t = T/2$, when the probability of the best arm dropped from 0.8 to 0.8-Δ_g, the probability of the second best dropped from 0.8-Δ_g to 0.8-2Δ_g (and so forth for other intermediate arms), while the probability of the worst arm increased from 0.8-4Δ_g to 0.8. We altered Δ_g from 0.02 to 0.2 with a step size of 0.02 for each independent test case, while we simulated all algorithms on each case for 200 sessions, observing the average final cumulative regret. We then increased $\#cp$ by one, fixed the change point time steps to be evenly distributed over the timesteps, and repeated the procedure. In Fig.5 a visualization of the performance for each algorithm is presented, for different gaps Δ_g (rows), and $\#cp$ (columns).

The results once again provided evidence that MLB-KF combines the behavior of both MLB and KF-MANB in a satisfactory manner. The blue bottom left area of the performance test space of MLB, was inherited by MLB-KF. The satisfactory performance of KF-MANB, on the right side of the test space, was also inherited by MLB-KF. D-UCB and SW-UCB exhibited similar performances, while Adapt-EvE achieved great performance for small number of change points, but dramatically increased its regret on this type of fast changing environment (as also noted in problem set I). For a total average evaluation, we observed the distribution of each algorithm's total cumulative regret, for all Δ_g and $\#cp$. The results are here shown as box plots in Fig.6. MLB-KF demonstrated the lowest total average regret, with also sufficient interquartile range, better than that of MLB but slightly larger than that of KF-MANB. D-UCB had a better performance in this stochastically changing environment, than the one of problem set I.

D. Problem Set III

Here we investigated the performance of all algorithms on 4 different cases of single arm switching. In a Best-Worst scenario (BW) the optimal arm abruptly becomes the worst while in Worst-best (WB) the worst arm becomes the optimal one. With the same ideas we also investigated BWB and WBW cases. The initial probabilities are initialized as 0.8, 0.5, 0.4, 0.3, 0.2 for each arm respectively. For BW case the optimal arm of probability 0.8 abruptly drops to 0.1 at $t = T/2$. In WB case the worst arm with probability 0.2 abruptly rises to 0.9 at $t = T/2$. For BWB the best arm drops to 0.1 at $t = T/3$ and rises back to 0.8 at $t = 2T/3$ and for WBW case the worst arm rises to 0.9 at $T/3$ and falls back to 0.2 at $t = 2T/3$. We run 200 sessions for each algorithm and calculated the average performance in terms of regret as seen in Fig.7 for all 4 scenarios.

In BW case, MLB and MLB-KF achieved the optimal performance. This is because reduced immediate rewards lead to a decay of mid-term reward $\overline{r}_t$. Yet the long-term reward $\overline{\overline{r}}_t$ has higher inertia. Therefore their difference $\overline{r}_t - \overline{\overline{r}}_t$ becomes less than zero, resulting in a decay of β_t, re-engaging exploration efficiently. KF-MANB was the second best, while D-UCB once again suffered from inertia.

In WB case MLB exhibited very large values of regret. From the graphs of Fig. 7, it seems that the policy of action choice remained unchanged. The same ocurred with Adapt-EvE, while D-UCB achieved the best performance. SW-UCB and KF-MANB exhibited similar performances. It can be seen that, MLB-KF followed the behavior of MLB at first, but when MLB started to perform badly, it adapted its behavior by denoting similar performance with KF-MANB. This example

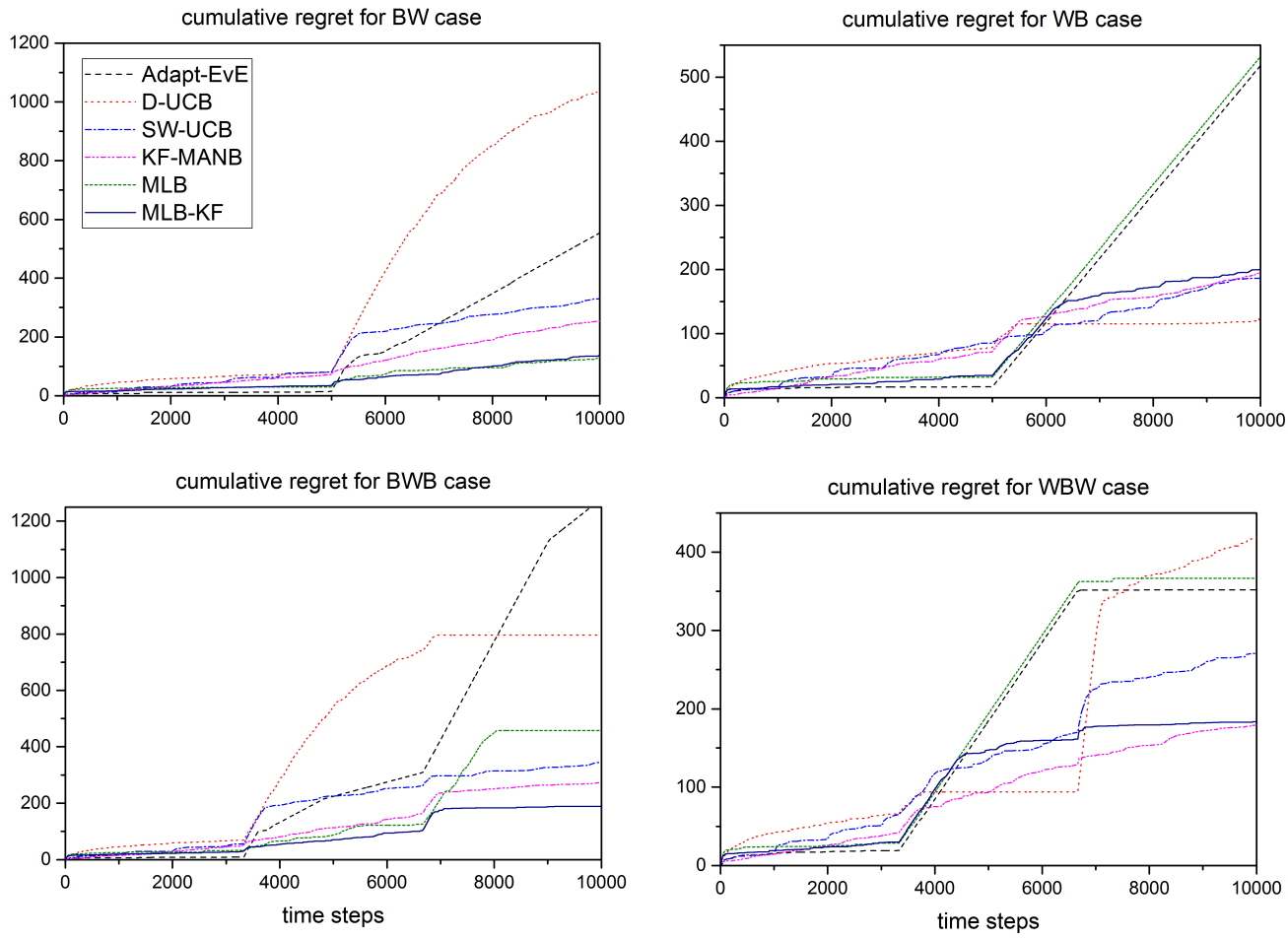

Fig. 7. top left: best-worst case, top right: worst-best case, bottom left: best-worst-best case, bottom right: worst-best-worst case

encapsulates the core of this hybrid algorithm in the best manner.

In BWB a combination of the former two take place, where MLB-KF demonstrated the best performance, with KF-MANB running second and SW-UCB third. MLB adapted very fast at the first change point (since this was a BW case), but its regret increased on the next change point (since this is a WB case). In WBW case MLB-KF achieved similar performance with KF-MANB, where the two of them exhibited better performance than all others.

V. Total Evaluation and Discussion

Testing an algorithm on a scenario that was also used in order to tune its parameters, should restrict its evaluation of performance to concern only this specific environment. Here we avoided this bias by testing the algorithms on many different scenarios, altering the most crucial components of a stochastic and non-stationary multi armed bandit, while keeping the parameters that were tuned on a simple mid-case. However, it should also be mentioned that when we tested the algorithms on the same scenario that was used for parameter tuning, meta-learning achieved the best performance than all others, while the hybrid model of meta-learning and sibling kalman filters, achieved the second best.

Without taking in mind the performance of the hybrid algorithm, it was observed that on an average picture, KF-MANB and MLB battled for the first place. MLB demonstrated very low regret in many cases, but also high variance and low robustness. On the other hand, KF-MANB had the most robust behavior, by also achieving one of the best performances. A trade-off between their performances was embodied with the use of a hybrid algorithm, which we named MLB-KF.

In nature, simple life events like everyday choices and actions have usually faster dynamics of stochasticity in terms of rewards, than the ones that have been tested in most of the literature. The evaluation of decision makers on different rates of abrupt changes, was therefore one of the many cases that we explored. More specifically, the presence of global change points in problem set 1 – occurring in a stochastic manner – led MLB to exhibit the best performance for small expected number of change points, while KF-MANB demonstrated both robustness and low regret. MLB-KF inherited the best from these performances, by following the regret dynamics of the most optimal algorithm between the two, during the time steps

of each session.

Distinguishing the best action when the expected rewards between them vary from smallest values to largest ones has also a significant meaning. The short term regret when choosing a non optimal action at the presence of a small gap may be low, but the cumulative regret after long periods will be increased, at least linearly with time. In problem set 2, we altered the difference between the expected reward of the optimal arm and the second best, considering that these alterations can provide an equivalent measure of the resolution limit that each algorithm can achieve. We also altered the change point rate, with a deterministic manner for better visualization of results, to investigate how the environmental change rate also affects the learning rate (as well as how the adaptation is correlated with this gap). From our simulations it was deduced that our hybrid algorithm MLB-KF demonstrated the best behavior on these cases, demonstrating both robustness and low regret.

To properly test exploration-exploitation trade-off, cases where the learned optimal action becomes the less optimal should be tested. Similarly, cases where the learned optimal action restrains its expected rewards but a non-optimal action becomes optimal should also be tested. In problem set 3, MLB adapted greatly in the former situation while D-UCB adapted greatly in the latter. However both of them demonstrated very large regret in the opposite case. MLB-KF once again demonstrated the best average performance on these types of setup. Cases where abrupt changes of each arm take place independently (instead of globally) were also tested, as well as cases with drifting changes of environment instead of abrupt (not shown here). Likewise, MLB-KF demonstrated evidence of an overall best empirical performance, yet these situations can be evaluated independently and are not on the scope or our work here.

VI. Conclusion

In this work we presented empirical evidence that a hybrid algorithm which makes use of a bio-inspired approach to properly tune exploration with meta-learning, combined with a sibling Kalman filter to estimate each arm's action value, adding a measure of uncertainty as an exploration bonus, can lead to an adaptive learning strategy which efficiently manages the exploration-exploitation trade-off dilemma.

We evaluated the performance of some of the best state-of-the-art algorithms, namely: KF-MANB [8] , D-UCB [7], SW-UCB [6] and Adapt-Eve [20], with two new algorithms proposed in this paper: a modification of meta-learning for reinforcement learning proposed in [19] termed MLB (meta-learning for bandits) and a new hybrid, named MLB-KF (meta-learning for bandits with kalman filters), while manipulating the most important components of the environment, using the same set of parameters found after proper tuning on a mid-case setup.

The results of our simulations suggest that a hybrid model inheriting and combining the adaptive behavior from both KF-MANB and the meta-learning algorithm with a proper trade-off between performance (in terms of regret and fast-adaptation) and robustness, resulted in efficient empirical behavior in a series of simple non-stationary scenarios. These results seem promising relative to real life applications due to the higher rate of alterations of the environment in terms of episodes, although more complex setups should be investigated in the future.

Acknowledgment

This research work has been partially supported by the EU-funded Project BabyRobot (H2020-ICT-24-2015, grant agreement no. 687831), by the Agence Nationale de la Recherche (ANR-12-CORD-0030 Roboergosum Project and ANR-11-IDEX-0004-02 Sorbonne-Universités SU-15-R-PERSU-14 Robot Parallearning Project), and by Labex SMART (ANR-11- LABX-65 Online Budgeted Learning Project).

References

[1] D. Koulouriotis and A. Xanthopoulos, "Reinforcement learning and evolutionary algorithms for non-stationary multi-armed bandit problems," *Applied Mathematics and Computation*, vol. 196, no. 2, pp. 913 – 922, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0096300307007448

[2] A. Sokolov, J. Kreutzer, C. Lo, and S. Riezler, "Learning structured predictors from bandit feedback for interactive nlp," in *ACL*. ACL, 2016.

[3] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.

[4] R. Allesiardo and R. Fraud, "Exp3 with drift detection for the switching bandit problem," in *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, Oct 2015, pp. 1–7.

[5] E. Kaufmann, O. Cappé, and A. Garivier, "On bayesian upper confidence bounds for bandit problems." in *AISTATS*, 2012, pp. 592–600.

[6] A. Garivier and E. Moulines, "On upper-confidence bound policies for non-stationary bandit problems," *arXiv preprint arXiv:0805.3415*, 2008.

[7] L. Kocsis and C. Szepesvári, "Discounted ucb," in *2nd PASCAL Challenges Workshop*, 2006, pp. 784–791.

[8] O.-C. Granmo and S. Berg, *Solving Non-Stationary Bandit Problems by Random Sampling from Sibling Kalman Filters*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 199–208.

[9] J. Kober, J. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, pp. 1238–1274, 2013.

[10] M. Khamassi, S. Lallée, P. Enel, E. Procyk, and P. Dominey, "Robot cognitive control with a neurophysiologically inspired reinforcement learning model," *Frontiers in Neurorobotics*, vol. 5:1, 2011.

[11] F. Benureau and P.-Y. Oudeyer, "Diversity-driven selection of exploration strategies in multi-armed bandits," in *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. IEEE, 2015, pp. 135–142.

[12] K. Doya, "Metalearning and neuromodulation." *Neural Netw*, vol. 15, no. 4-6, pp. 495–506, 2002.

[13] N. Daw, Y. Niv, and P. Dayan, "Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control." *Nat Neurosci*, vol. 8, no. 12, pp. 1704–1711, 2005.

[14] N. D. Daw and K. Doya, "The computational neurobiology of learning and reward," *Current opinion in neurobiology*, vol. 16, no. 2, pp. 199–204, 2006.

[15] M. Khamassi, P. Enel, P. Dominey, and E. Procyk, "Medial prefrontal cortex and the adaptive regulation of reinforcement learning parameters," *Progress in Brain Research*, vol. 202, pp. 441–464, 2013.

[16] M. van der Meer, Z. Kurth-Nelson, and A. D. Redish, "Information processing in decision-making systems," *The Neuroscientist*, vol. 18, no. 4, pp. 342–359, 2012.

[17] N. D. Daw, J. P. O'Doherty, P. Dayan, B. Seymour, and R. J. Dolan, "Cortical substrates for exploratory decisions in humans," *Nature*, vol. 441, no. 7095, pp. 876–879, 2006.

[18] M. J. Frank, B. B. Doll, J. Oas-Terpstra, and F. Moreno, "Prefrontal and striatal dopaminergic genes predict individual differences in exploration and exploitation," *Nature neuroscience*, vol. 12, no. 8, pp. 1062–1068, 2009.
[19] N. Schweighofer and K. Doya, "Meta-learning in reinforcement learning." *Neural Networks*, vol. 16, no. 1, pp. 5–9, 2003.
[20] C. Hartland, S. Gelly, N. Baskiotis, O. Teytaud, and M. Sebag, "Multi-armed bandit, dynamic environments and meta-bandits," 2006.
[21] R. L. Solomon and J. D. Corbit, "An opponent-process theory of motivation: I. temporal dynamics of affect." *Psychological review*, vol. 81, no. 2, p. 119, 1974.