Collective Intelligence in Decision-Making with Non-Stationary Experts

AXEL ABELS*, Machine Learning Group, Université Libre de Bruxelles, Belgium, AI Lab, Vrije Universiteit Brussel, Belgium, and FARI Institute, Université Libre de Bruxelles - Vrije Universiteit Brussel, Belgium
VITO TRIANNI, Institute of Cognitive Sciences and Technologies, National Research Council, Italy
ANN NOWÉ, AI Lab, Vrije Universiteit Brussel, Belgium and FARI Institute, Université Libre de Bruxelles - Vrije Universiteit Brussel, Belgium
TOM LENAERTS, Machine Learning Group, Université Libre de Bruxelles, Belgium, AI Lab, Vrije Universiteit Brussel, Belgium, Center for Human-Compatible AI, UC Berkeley, USA, and FARI Institute, Université Libre de Bruxelles - Vrije Universiteit Brussel, Belgium

When sufficient experience to make informed decisions is unavailable, expert advice can help us navigate uncertainty. As expertise evolves, driven by continuous learning in human experts or model updates in artificial experts, it is crucial to adopt adaptive approaches. Existing methods for exploiting non-stationary experts focus on competing with the single best expert. In contrast, this work harnesses the power of collective intelligence to facilitate better decision-making in the face of evolving expertise or dynamic environments. To achieve this, we propose the novel CORVAL approach which optimally combines the insights of multiple experts. By adapting to drifts in expertise, our novel approach can surpass the performance of the single best expert as well as previous approaches. Empirical evaluations on a diverse range of non-stationary problems, including active learning applications, showcase the improved performance of our approach in collective decision-making scenarios.

JAIR Associate Editor: Roni Stern

JAIR Reference Format:
Axel Abels, Vito Trianni, Ann Nowé, and Tom Lenaerts. 2025. Collective Intelligence in Decision-Making with Non-Stationary Experts. *Journal of Artificial Intelligence Research* 83, Article 9 (July 2025), 27 pages. DOI: 10.1613/jair.1.16228

1 Introduction

When faced with complex problems, decision-makers can choose to delegate the task to a collective of experts, be they human or artificial. These experts use their knowledge to provide advice, which the decision-maker can aggregate in some intelligent manner to make an appropriate decision. As accurate estimates of experts' quality are often unavailable in advance, the decision-maker must learn to effectively exploit the advice of the collective through a process of trial and error.

*Corresponding Author.

Authors' Contact Information: Axel Abels, ORCID: 0000-0003-2784-8653, axel.abels@ulb.be, Machine Learning Group, Université Libre de Bruxelles, Brussels, Belgium and AI Lab, Vrije Universiteit Brussel, Brussels, Belgium and FARI Institute, Université Libre de Bruxelles - Vrije Universiteit Brussel, Brussels, Belgium; Vito Trianni, ORCID: 0000-0002-9114-8486, vito.trianni@istc.cnr.it, Institute of Cognitive Sciences and Technologies, National Research Council, Rome, Italy; Ann Nowé, ORCID: 0000-0001-6346-4564, ann.nowe@vub.be, AI Lab, Vrije Universiteit Brussel, Brussels, Belgium and FARI Institute, Université Libre de Bruxelles - Vrije Universiteit Brussel, Brussels, Belgium; Tom Lenaerts, ORCID: 0000-0003-3645-1455, tom.lenaerts@ulb.be, Machine Learning Group, Université Libre de Bruxelles, Brussels, Belgium and AI Lab, Vrije Universiteit Brussel, Brussels, Belgium and Center for Human-Compatible AI, UC Berkeley, Berkeley, California, USA and FARI Institute, Université Libre de Bruxelles - Vrije Universiteit Brussel, Brussels, Belgium.

DOI: 10.1613/jair.1.16228

In particular, we are interested in the use of collectives of experts for bandit problems [34, 4], wherein a learner acquires increasing knowledge about the quality of the alternatives at its disposal as it makes decisions. Bandits have been widely used to model real-world problems such as content recommendation [42] or clinical trials [37]. The need for exploration, i.e., the need to acquire more certainty about the quality of each alternative, makes the problem especially challenging in high-risk settings. When the learner is endowed with little or no prior knowledge about the problem, the opportunity to query a collective of experts may boost success. Consider medical diagnostics as an example. Here, blindly administering medicines to learn the optimal treatment for a set of symptoms is unreasonable. Because initial uncertainty can be catastrophic, medical experts can provide a learner with the guidance required to decide on a treatment. Utilizing a collective, as opposed to a single expert, has the potential to improve performance thanks to the diversity of perspectives provided by different experts. Learning algorithms provide a solution by identifying the optimal way in which to make decisions based on the collective's knowledge.

Algorithms for bandits with expert advice learn to make such decisions in bandit settings [4]. By acquiring advice from experts, learners can significantly improve their performance, provided they are able to prioritize the advice of stronger experts. However, as human experts acquire or lose knowledge in a field, their usefulness can change. Similarly, artificial experts — for example machine learning models — can become increasingly relevant as they are updated based on the latest data or understanding. Algorithms for bandits with expert advice must therefore react to changes in expertise in order to guarantee sustained performance.

While adaptive methods such as EXP4.S or AdaBinGreedy have been proposed to deal with non-stationary experts [26], they focus on tracking the best expert over time. When no expert in the collective is optimal, the algorithm inevitably suffers a loss in performance relative to optimality, as the best achievable performance — that of the best expert — is not optimal.

Bearing this limitation in mind, our work aims to leverage the collective more effectively. In particular, we posit that by learning a policy which acts on the optimal aggregation of experts — rather than acting on the single best expert — we can avoid the drawbacks of previous approaches. Specifically, this allows us to learn a policy which is not bounded by the performance of the single best (moving) expert in the way previous methods such as EXP4.S are. We thus enable the emergence of collective intelligence: performance superior to the best expert in the group.

In addition, we solve previous approaches' reliance on knowledge about the expected rate of change. In particular, methods such as EXP4.S [26] need to be tuned to the expected rate at which experts change. If this rate is unknown, these methods either adapt too little or too much, resulting in sub-optimal performance. To alleviate this requirement, we propose a two-level approach which on one hand maintains multiple instances of the non-stationary algorithms, and then dynamically aggregates the most appropriate instances according to the observed rate of change. Our method innovates on existing corralling techniques (e.g., [3, 27]) by transitioning from policy selection to estimate aggregation.

Previous methods, as well as the required background knowledge, are presented in Section 2. We then explore the theoretical benefits of optimizing towards collective intelligence in Section 3 and provide a practical approach for realizing this goal. We provide theoretical results for this approach in Section 4 and compare them to bounds of previous algorithms. In Section 5 we introduce CORVAL, which dynamically adjusts its adaptiveness to the experts' non-stationarity. We conduct an extensive empirical evaluation of our novel method and compare it to a range of baselines in Section 6.1. Through this evaluation we demonstrate that our proposed method provides a significant improvement in performance over previous adaptive algorithms for a wide variety of configurations. We further show that unlike previous algorithms, which require their adaptiveness to be tuned to match changes in expertise, our novel approach is more robust in terms of its adaptiveness parameter. We conclude by applying our methods to active learning in Section 6.2, demonstrating improved performance on a concrete real-world problem.

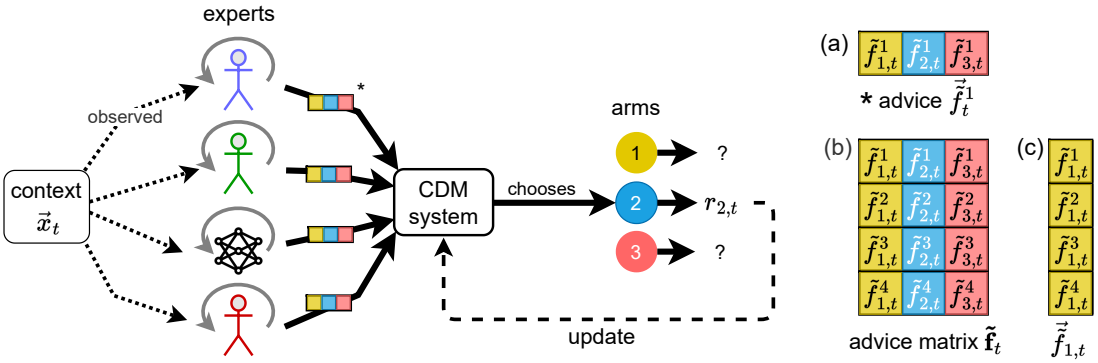

Fig. 1. Illustration of a CDM system aggregating the advice of $N = 4$ experts to select one of the $K = 3$ arms. Each expert observes the context at time t and then provides an advice vector to the CDM system (e.g., (a) $\vec{\tilde{f}}^1_t$ is the first expert's advice). Note that one can have both human and artificial experts as depicted in the figure. Based on the chosen arm's outcome ($r_{2,t}$, sampled from a distribution with mean $f(2, \vec{x}_t)$), the CDM system updates how it aggregates the expert advice. Independently, experts may change over time (as indicated by the grey self-loops), inducing a non-stationary aspect to the problem. The advice matrix (b) $\tilde{\mathbf{f}}_t$ is a concatenation of the advice vectors, and (c) $\vec{\tilde{f}}_{1,t}$ is the advice for arm 1.

To summarize, our key contributions are:

- We expose the limitations of current approaches to bandits with non-stationary experts and show the potential for leveraging collective intelligence by framing the problem as a non-stationary contextual bandit.
- We introduce a novel two-level approach to adapt to the experts' rate of change, thereby eliminating the requirement for prior knowledge about this rate.
- We theoretically evaluate the benefits of this approach compared to previous approaches. In particular, we show the collective tends to be less non-stationary than individuals within the collective.
- We present comprehensive empirical support for the improvements provided by our novel approach, including an application to active learning on real world data sets.

2 Bandits with Non-Stationary Expert Advice

The observation that real-world processes are rarely stationary induces the need for methods that can adapt to change. We first formalize the non-stationary setting and present prior work as well as the foundation for our novel approach. The use of non-stationary experts has been explored in related problems such as prediction with expert advice [16] or inverse reinforcement learning [25]. The focus of this work however is bandit problems, which we formalize in the following section and for which we present state-of-the-art approaches. For reference, we provide a notation table in the supplementary material.

2.1 Bandits with Expert Advice

As illustrated by Figure 1, we consider a sequential decision-making problem wherein a learner has to repeatedly choose one out of K arms with stochastic outcomes. The quality of each arm k is determined by a time-dependent context $\vec{x}_t$ (i.e., a set of d features characterizing the current decision), and an *a-priori* unknown function $f : [K] \times \mathbb{R}^d \to [0, 1]$ which maps a context and arm to an expected reward. For example, if the context represents a patient's symptoms and medical history, the possible treatments form a set of arms, with higher rewards for

the most beneficial treatments, in terms of some measure which captures the interest of the stakeholders, such as QALY [39], cost-effectiveness ratio, or simply treatment effectiveness. In the *contextual bandit problem* [13, 36], a learner approximates the function f to make its decisions. In settings wherein learning this approximation is infeasible because of the problem's complexity, we can instead exploit existing expertise. This is the focus of *bandits with expert advice* [4, 5, 7, 2].

In order to choose arms that maximize the expected reward the learner then relies on a set of N experts who provide advice in the form of an approximation of the true reward function f, i.e. $\vec{\tilde{f}}^{\,n}_t : [K] \times \mathbb{R}^d \to [0, 1]$. When it is clear from context, we will omit the function parameters for brevity; we will denote expert n's advice for a given arm k at time t by $\tilde{f}^n_{k,t} := \tilde{f}^n_t(k, \vec{x}_t)$, and its vector of advice by $\vec{\tilde{f}}^n_t := \{\tilde{f}^n_{1,t}, ..., \tilde{f}^n_{K,t}\}$. Similarly, we let the vector of advice for a given arm k be $\vec{\tilde{f}}_{k,t} := \{\tilde{f}^1_{k,t}, ..., \tilde{f}^N_{k,t}\}$. The aim of the learner is then to learn how to act on expert advice, specifically, to learn a policy which maps the $N \times K$ advice matrix $\tilde{\boldsymbol{f}}_t := \{\vec{\tilde{f}}^{\,1}_t, ..., \vec{\tilde{f}}^{\,N}_t\}^\intercal$ to a probability distribution over the arms. After choosing an arm k_t (e.g., a treatment) at time t, the learner observes a reward r_t sampled from a reward distribution with mean $f(k_t, \vec{x}_t)$. The aim of the learner is to maximize the average reward collected over T steps, i.e., $\overline{R} = \frac{1}{T}\sum_{t=1}^{T} r_t$.

Non-Stationarity. Changes in the knowledge or behavior of experts, as well as changes in the reward function itself, can introduce a change in the optimal aggregation of expert advice. Specifically, we consider experts' approximations to be time-dependent, such that $\exists n, t : \tilde{f}^n_t \neq \tilde{f}^n_{t+1}$. We consider performance in terms of dynamic regret, in other words, we aim to compete with the best expert at each step. Let $(k^n_t)^T_{t=1}$ be the sequence of arms chosen by expert n such that $k^n_t = \arg\max_{k\in[K]} \tilde{f}^n_{k,t}$ and let

$$\mathcal{R}_{\text{IND}} = \sum_{t=1}^{T} \max_{n\in[N]} f(k^n_t, \vec{x}_t)$$

be the cumulative reward of the best individual in the expert set. Let $(k_t)^T_{t=1}$ be the sequence of arms pulled by a learner ALG. The dynamic regret of ALG with respect to the best individual is:

$$R^{\text{ALG}}_{\text{IND}} = \mathcal{R}_{\text{IND}} - \sum_{t=1}^{T} f(k_t, \vec{x}_t) \tag{1}$$

The rate of change $\Delta \in [0, T]$ captures how rapidly the rewards of experts change:

$$\Delta = \sum_{t=1}^{T} \max_{n\in[N]} |f(k^n_t, \vec{x}_t) - f(k^n_{t-1}, \vec{x}_{t-1})| \tag{2}$$

In words, this sums the maximal change in expected reward across experts for each time step. Expressing regret in terms of Δ can account for both sporadic large changes and/or continuous small changes.

In the following sections we present existing methods for minimizing this measure of regret and highlight their limitations.

2.1.1 Reduction to MAB.

A straightforward approach to dealing with non-stationary expert advice in bandits is to reduce the problem to a simple multi-armed bandit (MAB), wherein arms are experts to be followed, and choosing an arm means acting on the corresponding expert's advice. Non-stationary variants of classic MAB algorithms such as UCB [19] or

Thompson Sampling [34] can then be applied to this *meta-bandit* to solve the problem. Variants adapt to non-stationarity by introducing discounting (such as d-TS [31] or D-UCB [17]), by applying them on a sliding window of experiences (as in SW-TS [35] or SW-UCB [17]), or by restarting the learning process (e.g. ReRUN-UCB-V [38])

While a larger discount (resp. a smaller window) increases adaptiveness, the single-expert update of the MAB reduction limits it. In particular, the adaptiveness of this approach is limited by the number of steps required to update all expert estimates, which scales linearly with the number of experts. As a result, it is impossible for this meta approach to deal with high non-stationarity.

To see this, consider the case where $T/\Delta < N$. Since Δ is the cumulative magnitude of change, the ratio T/Δ can be viewed as a rough timescale for change: how many steps pass, on average, before one unit of change accumulates. If this timescale is smaller than N, i.e., $T/\Delta < N$, the learner cannot query all experts even once before the environment shifts significantly. This makes it impossible for meta-bandit approaches to keep up with high non-stationarity, especially when N is large.

This highlights the need for algorithmic solutions whose adaptiveness is not similarly limited by the number of experts. Specialized methods allow the learner to update its beliefs about all experts simultaneously, enhancing performance for large expert sets.

2.1.2 *Specialized Methods.*

The EXP4.S algorithm [26] — a non-stationary extension of EXP4 [4] — maintains a probability distribution over the experts which it uses to sample one expert to act on at each time step. The weight of each expert in this distribution is updated as a function of each expert's estimated reward, which is computed from the observed reward and from the experts' advice. Similarly to d-TS or D-UCB, EXP4.S decays the probability distribution over the experts at each step. In doing so, the impact of older updates is repeatedly mitigated, effectively increasing the relative importance of recent updates, and thus recent experiences. The strength of this decay is controlled by a parameter μ. If the rate of change Δ is known, we can tune μ accordingly to optimize performance [26]. When μ is unknown, it can be desirable to design algorithms which adapt to this unknown quantity. The AdaBinGreedy algorithm [26] achieves this through a statistical test which detects non-stationarity. Its analysis shows that this allows AdaBinGreedy to asymptotically outperform EXP4.S when the rate of change is large. However, this algorithm is impractical, as the statistical tests are trivially false for most realistic time horizons[1]. The same authors propose other alternatives to EXP4.S which require similar tuning and result in worse performance guarantees. While they have the advantage of being less computationally demanding, computational cost is not the focus of this work. We are primarily interested in fully exploiting the diversity of smaller expert sets, for which the computational savings would be negligible, especially when they come at the cost of significant performance losses. The Ada-ILTCB$^+$ algorithm of Chen et al. 2019 further builds on the work of Luo et al. 2018 but is similarly rendered impractical by bounds used in statistical tests, which are trivially false for most reasonable time horizons.

Similarly to the MAB reduction, EXP4.S results in decisions based on the advice of a single expert (sampled from the distribution over the experts). This again bounds performance by that of the best expert in a window of time. Note that, as decisions are based on a single expert's advice, these algorithms are sensitive to individual changes in expertise. In particular, a large change in the quality of the expert with the highest weight will significantly impact performance until weights are adjusted. This is also captured by the rate of change as defined in Equation 2, which contains a maximization over all the experts for each step. In addition, EXP4.S is not directly applicable to the value advice we consider here. Instead, we can apply EXP4.S over the set of policies of size N induced

[1]The statistical tests used in AdaBinGreedy allow for an asymptotic analysis with favorable theoretical guarantees. However, as we show in the supplementary information, those asymptotic guarantees rely on a statistical test which is trivially false for any time horizon $T < 5.9 \times 10^7$. In practice, any application with more frequent changes would therefore be treated as a stationary problem by AdaBinGreedy, thus failing to address our needs.

by greedily acting on each expert's advice. This mapping necessarily induces a loss of information available to the aggregation algorithm. Existing methods to optimize directly over the set of regressors, such as Regressor Elimination [2] and RegCB [15], assume stationarity, and are therefore not applicable. Moreover, as these methods converge towards a single expert (regressor), they also fail to achieve collective intelligence.

In order to realize the full potential of expert sets, we propose in the following section to achieve collective intelligence by optimizing towards an aggregate of expert advice.

3 Enabling Collective Intelligence Through Dynamic Aggregates

We say that collective intelligence emerges when a solution obtained through the collective (of experts) is superior to any individual in the collective. Collective intelligence can thus be achieved if an aggregate of expert advice is superior to any single expert's advice. Our first contribution is the proposal of this collective intelligence approach to bandits with non-stationary experts as well as the analysis of its applicability and benefits.

Similar to previous work designed for the stationary setting [1], we propose to treat each expert's advice as a feature based on which the true outcome is approximated. We thus assume the existence of a sequence of functions $\mathcal{E}_t : \mathbb{R}^N \to [0, 1]$ from expert advice to expected rewards, such that $f(k, \vec{x}_t) = \mathcal{E}_t(\vec{\tilde{f}}_t^{\,1}(k, \vec{x}_t), ..., \vec{\tilde{f}}_t^{\,N}(k, \vec{x}_t))$. In words, the expected outcome of an arm is a function of the expert advice provided for said arm. Note that greedily acting on the current estimate is typically a flawed approach, as the stochastic bandit feedback necessitates a careful balance of exploration and exploitation to ensure proper convergence. In addition, $\mathcal{E}$ is not fixed throughout the learning process, as the experts themselves are non-stationary. This non-stationarity must thus also be accounted for in the decision process. In particular, our aim is to find the optimal linear aggregation of expert advice at any time step. That is, we assume rewards are the result of a linear mapping of expert advice with some conditionally subgaussian noise ϵ:

$$r_{k,t} = \langle \vec{\tilde{f}}_{k,t}, \vec{\theta}_t^* \rangle + \epsilon$$

Where $\vec{\tilde{f}}_{k,t}$ is the k-th column of the advice matrix, i.e., the N-sized vector of advice for arm k at time t. Our goal is thus to select arms maximizing the expected reward, $\mathbb{E}[r_{k,t}] = \langle \vec{\tilde{f}}_{k,t}, \vec{\theta}_t^* \rangle$, which typically involves finding the minimizing weights at each time step:

$$\vec{\theta}_t^* = \underset{\vec{\theta} \in \mathbb{R}^N}{\arg\min} \sum_{k \in [K]} (f(k, \vec{x}_t) - \langle \vec{\tilde{f}}_{k,t}, \vec{\theta} \rangle)^2 \tag{3}$$

Note that in the bandit with expert advice setting, the weights can be intuitively interpreted as the importance we afford to each expert.

Let $(k_t^*)_{t=1}^T$ be the sequence of arms chosen by acting on the optimal moving weights, i.e., $k_t^* = \arg\max_{k \in [k]} \langle \vec{\tilde{f}}_{k,t}, \vec{\theta}_t^* \rangle$, the cumulative reward of this optimal aggregate is then

$$\mathcal{R}_{\text{AGG}} = \sum_{t=1}^{T} f(k_t^*, \vec{x}_t)$$

As before, let $(k_t)_{t=1}^T$ be the sequence of arms pulled by a learner ALG. The notion of dynamic regret in this setting is now with respect to the best aggregate:

$$R_{\text{AGG}}^{\text{ALG}} = \mathcal{R}_{\text{AGG}} - \sum_{t=1}^{T} f(k_t, \vec{x}_t) \tag{4}$$

Note that typically, we have that $\mathcal{R}_{\text{AGG}} > \mathcal{R}_{\text{IND}}$[2], and therefore this notion of regret is harder than the previously introduced $R^{\text{ALG}}_{\text{IND}}$.

As before, it is useful to quantify the non-stationarity. We thus let Δ_θ be the linear weight analog of the rate of change: $\sum_{t=2}^{T} \mathbb{E}[|\langle \theta^*_{t-1}, \vec{\tilde{f}}_{k,t}\rangle - \langle \theta^*_t, \vec{\tilde{f}}_{k,t}\rangle|] = \sum_{t=2}^{T} \mathbb{E}[|\langle \theta^*_{t-1} - \theta^*_t, \vec{\tilde{f}}_{k,t}\rangle|] \leq \Delta_\theta$. In words, Δ_θ bounds the cumulative error induced by applying at each timestep the optimal weights of the previous timestep. The larger this rate of change, the more the optimal linear aggregation of expert advice changes. Due to this non-stationarity in the weights, simple ridge regression (as in previous works involving linear models for bandits [13, 1]) is inappropriate.

Instead, we consider the use of existing algorithms for non-stationary linear bandits. In order to track this optimal weight vector, we make use of D-LinUCB [32], a non-stationary algorithm for the linear bandit. It achieves non-stationarity by iteratively decaying the weights of its linear model. In doing so it increases the importance of recent experiences. By re-purposing non-stationary linear bandit algorithms in this way we can optimize towards the best aggregate of expert advice and thus achieve collective intelligence.

The resulting approach maintains a linear mapping of expert advice to arm outcomes which prioritizes accuracy on recent experiences. Note that while we make use here of D-LinUCB, we could make use of other non-stationary linear bandit algorithms, such as dLinUCB [40] to solve the constructed *meta* bandit. We thus distinguish this approach to bandits with non-stationary experts — which consists in constructing a meta contextual bandit and solving it with a non-stationary bandit algorithm — from the algorithm used to solve the constructed bandit. We refer to the former as MCB[μ] (short for **M**eta-**C**MA**B** with μ-adaptiveness), where the hyper-parameter μ controls the algorithm's adaptiveness.

Compared to the methods presented in the previous sections we distinguish two significant advantages.

Collective Intelligence. First, similarly to Meta-CMAB [1] in the stationary setting, we leverage collective intelligence which allows us to optimize towards a stronger aggregator. In most realistic settings the aggregate of expert advice is a better estimate than the single best expert [1]. Therefore, by acting on the best aggregate of expert advice we outperform the single best expert and thus enable collective intelligence.

Non-Stationarity. In addition, this aggregation of multiple experts is less sensitive to individual shifts in expertise. In contrast to previously introduced approaches, the model learned by this approach is less likely to concentrate its weights on a single expert. Individual changes in expertise therefore tend to have a lower impact on the model learned by MCB, resulting in an optimization towards a target which is less non-stationary.

In particular, we can bound shifts of the optimal model as a function of the shift in individual experts. Assume that at time $t+1$, experts undergo a shift in advice bounded by δ_{t+1}, such that

$$\max_{n\in[N]} \mathbb{E}[\max_{k\in[K]} |\tilde{f}^n_{k,t} - \tilde{f}^n_{k,t+1}|] = \delta_{t+1} \tag{5}$$

The expectation above, and in what follows, is over the set of contexts (and arms if not specified within the expectation).

The expected shift of the linear model then depends on how experts shift. Assume θ^*_t is drawn at random from the N-dimensional simplex, such that $\|\theta^*_t\|_1 = 1$ [3]. Let the instantaneous shift of the aggregation from t to $t+1$ be

$$\delta^\theta_{t+1} = \mathbb{E}[|\langle \theta^*_t, \vec{\tilde{f}}_{k,t}\rangle - \langle \theta^*_t, \vec{\tilde{f}}_{k,t+1}\rangle|] = \mathbb{E}[|\langle \theta^*_t, \vec{\tilde{f}}_{k,t} - \vec{\tilde{f}}_{k,t+1}\rangle|]$$

[2]Assuming no expert is perfect, and experts are not fully correlated, the best linear combination of expert advice is better than the single best expert [8, 1].

[3]Note that while this assumption might not hold exactly, it is typically the case that $\|\theta^*_t\|_1 = 1 \pm \epsilon$, with ϵ relatively small [8]

Since each expert's shift in advice is bounded by δ_t, we have that

$$\delta^{\theta}_{t+1} = \mathbb{E}[|\langle \theta^*_t, \vec{f}_{k,t} - \vec{f}_{k,t+1} \rangle|] \leq \mathbb{E}[\|\theta^*_t\|_1 \delta_{t+1}] = \delta_{t+1}$$

Where the last equality follows from the assumption that $\|\theta^*_t\|_1 = 1$. Note, however, that the maximization in (5) allows for a single expert to change by δ_{t+1}, while other experts remain unchanged. Assume this expert is the n-th expert. We then have that

$$\delta^{\theta}_{t+1} = \mathbb{E}[|\langle \theta^*_t, \vec{f}_{k,t} - \vec{\tilde{f}}_{k,t+1} \rangle|] \geq \mathbb{E}[\theta^*_{t,n} | \tilde{f}^n_{k,t} - \tilde{f}^n_{k,t+1}|] = \mathbb{E}[\theta^*_{t,n} \delta_{t+1}] = \delta_{t+1}/N$$

where the last equality follows from $\mathbb{E}[\theta^*_{t,n}] = 1/N$.
We therefore have that $\delta_{t+1}/N \leq \delta^{\theta}_{t+1} \leq \delta_{t+1}$, depending on how widespread expertise change is, with the lower bound occurring when a single expert changes, and the upper bound occurring when all experts change by the same amount. Typically we can expect the latter to be rare, in which case δ^{θ}_{t+1} is typically strictly smaller than δ_{t+1}, and the linear model thus shifts more slowly than individual experts.

4 Theoretical Results

While the advantages of optimizing towards an aggregate can be significant, they are only relevant if the algorithm is able to converge towards this aggregate. In this section we use a unified notation to restate theoretical bounds on the regret of previous methods and compare them to the bounds which our novel approach achieves. We distinguish regret with respect to the best expert ($R^{\text{ALG}}_{\text{IND}}$, as defined in Equation 1), and regret with respect to the best aggregate ($R^{\text{ALG}}_{\text{AGG}}$, as defined in Equation 4).

These bounds provide an indication of the influence of different factors (the number of experts, arms, time steps, and the non-stationarity) on the expected regret. The analysis of MAB algorithms which are to be applied to the MAB reduction, as well as EXP4.S, is done with respect to the single best (moving) expert. In other words, bounds express how far these algorithms are from equaling the performance of the moving single best expert.

Theorem 1. *The dynamic regret of SW-UCB applied to the reduction to a MAB is* $R^{MAB}_{IND} = O(\sqrt[3]{N(\Delta+1)T^2})$

Cheung et al. 2019 provide an upper bound on the regret of SW-UCB with respect to the best dynamic arm of

$$O(\sqrt[3]{d(B_T+1)T^2}) \tag{6}$$

with d the problem dimension (i.e., the number of arms), and B_T— the variation budget— a bound on the changes in expected means of the MAB's arms. By reducing the problem of bandits with expert advice to a non-stationary bandit, we create a MAB with $d = N$ arms (one per expert). B_T thus bounds the constructed MAB's arms. In this reduction, each of these arms represents an expert, and B_T is thus equivalent to Δ as defined in Equation 2. Substituting d and B_T, we obtain the bound of Theorem 1. Hence, over T time steps, the difference in cumulative reward between the best moving expert and D-UCB applied to the reduced non-stationary bandit grows sub-linearly with the time horizon (T) and with the number of experts (N). Note that any alternative approach based on a reduction to a MAB (such as those based on sliding windows, SW-UCB, or variants of the exponential weights EXP3 algorithm such as EXP3.S [4] or REXP3 [6]) will display a similar dependence on the size of the expert set.

The dynamic regret of EXP4.S as analyzed by Luo et al. 2018 is

$$R^{\text{EXP4.S}}_{\text{IND}} = O(\sqrt{TK \log N} + \sqrt[3]{T^2 \Delta K \log N})$$

Hence, compared to D-UCB, EXP4.S achieves better bounds with respect to the number of experts, but worse bounds with respect to the number of arms.

While EXP4.S applied over the expert set cannot attain collective intelligence, a naive alternative is to expand the expert set to include all linear aggregations. While this expanded expert set would be infinite, we can consider a finite approximation in the form of an N-dimensional grid over the linear weights space. Each point in this grid — corresponding to one weight vector — induces a policy. The resulting policy set, over which we can apply EXP4.S, is then exponential in the number of experts, and we therefore obtain a regret of

$$R_{\text{AGG}}^{\text{EXP4.S}} = O\left(\sqrt{TKN} + \sqrt[3]{T^2\Delta' KN}\right)$$

Wherein Δ' bounds the non-stationarity of the expanded expert set. Note that since this expanded expert set is a superset of the set of base experts, and the definition of Δ involves a maximization over all experts, we have that $\Delta' \geq \Delta$. While this approach can attain collective intelligence, it comes at the cost of a significantly worse dependence on N. Moreover, as the size of this expanded set grows exponentially with the number of base experts, the computational cost grows identically, rendering the approach impractical. Finally, EXP4.S would fail to exploit the structure inherent to the grid. Specifically, points which are close to each other in the grid are likely to result in similarly performing policies, which this approach fails to explicitly exploit.

Oracle-based approaches proposed by Luo et al. 2018 allow us to efficiently optimize over larger policy sets. However, in addition to impracticalities previously discussed (specifically the trivially false non-stationarity tests for smaller time horizons, see Supplementary Section B), the regret guarantees of these approaches grow with $\log N$, rather than EXP4.S' $\sqrt{\log N}$, thus resulting in regret which grows linearly with the number of experts when considering the grid approximation of the set of linear aggregators. For example, when taking the policy set to be again a grid approximation of all linear aggregators (of size exponential in N), the dynamic regret of Ada-BinGreedy is

$$R_{\text{AGG}}^{\text{ADA}} = \tilde{O}(KN(\sqrt[5]{\Delta' T^4} + \sqrt[3]{T^4}))$$

Wherein $\tilde{O}$ suppresses logarithmic factors in K, T and $1/\delta$.

Theorem 2. *The dynamic regret of MCB solved with D-LinUCB is* $R_{AGG}^{MCB} = O(\sqrt[3]{N^2\Delta_\theta T^2})$

If the assumptions underlying the analysis of D-LinUCB hold for MCB, its regret can be bounded by $O(d^{2/3}B^{1/3}T^{2/3})$ [32], where $B = \Delta_\theta$ bounds non-stationarity as before, and d denotes the number of dimensions of the contextual bandit to which it is applied. For MCB each expert's advice is used to construct one feature, and therefore the contextual bandit we construct has dimensionality $d = N$. The analysis of D-LinUCB relies on the following assumptions, note that we adapt the notation to conform to our reduction:

(1) The features are bounded, in the sense that there is some L such that $\|\vec{\tilde{f}}_{k,t}\|_2 \leq L \,\forall\, k, t$
(2) The optimal weights are bounded, i.e., there is some S such that $\|\theta_t^*\|_2 \leq S \,\forall\, t$
(3) The expected reward of the linear model is bounded, i.e., $|\langle \vec{\tilde{f}}_{k,t}, \theta_t^* \rangle| \leq 1 \,\forall\, k, t$

We assume experts are aware of the bounds on the reward, and therefore provide similarly bounded advice in $[0,1]^K$, features are therefore bounded, matching assumption 1 above. It is not as obvious that assumption 2 always holds. However, previous work on combining forecasters using linear weights suggests optimal weights tend to sum to at most 1 [8], suggesting this assumption tends to hold. Finally, we consider rewards within $[0,1]$, and therefore $\langle \vec{\tilde{f}}_{k,t}, \theta_t^* \rangle \in [0,1]$, should hold, confirming assumption 3 also holds.

The analysis of D-LinUCB therefore is valid when applied to the constructed contextual bandit. The bound on the regret of MCB can thus be obtained by substituting N for d, and Δ_θ for B in the bound of D-LinUCB, resulting in the bound of Theorem 2.

Compared to EXP4.S and the MAB reduction, this regret bound has a higher dependence on the number of experts: $O(\sqrt[3]{N^2})$ compared to $O(\sqrt{\log(N)})$ or $O(\sqrt[3]{N})$ respectively. This suggests that asymptotically, the

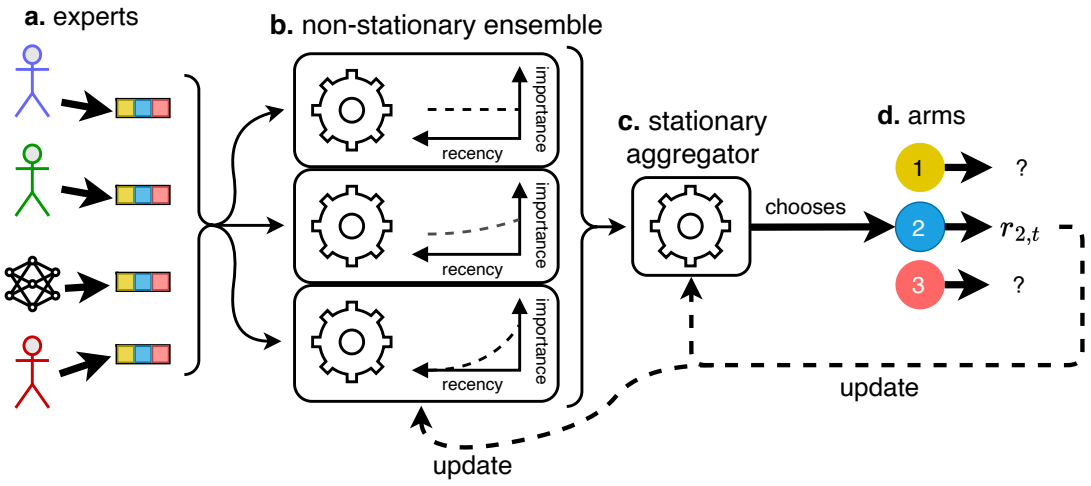

Fig. 2. The CDM system consists of two levels, first, an ensemble of non-stationary learners (e.g., EXP4.S, or MCB) with varying levels of non-stationarity (parameterized by μ). To determine which level of non-stationarity we should act on, we feed these learners' outputs into a secondary learner (c.) which uses the first layer to select an arm. The chosen arm's reward is then used to update the base learners as well as the aggregator.

performance of the MCB approach will degrade more as we increase the number of experts. Notwithstanding this issue, the approach has some advantages. First, there is no dependence on the number of arms. Secondly, in contrast to earlier approaches, the model learned by MCB is not limited to the performance of the single best expert (as EXP4.S or the reduction to a MAB are). Instead, the dynamic regret is with respect to the single best linear combination of expert advice. In other words, the best policy attainable by MCB is typically superior to the best policy attainable by EXP4.S or the MAB reduction. This is particularly useful if there is a large gap between the optimal linear mapping and the best expert. In other words, as the potential for collective intelligence increases, the potential benefits of the MCB approach improve. For example, when experts are uncorrelated but otherwise homogeneous in their level of expertise, the best linear model can significantly improve over the best expert [1]. Additionally, this approach further benefits when the rate at which the linear model changes (Δ_θ) is smaller than the rate at which experts change (Δ), which, as we show in Section 3, is likely in real-world scenarios.

5 Adaptive Non-Stationarity

The effectiveness of the methods discussed so far hinges on specific hyperparameters that control their adaptiveness. Notably, EXP4.S and D-TS rely on a $\mu \in [0, 1)$, where a larger μ increases adaptiveness. Similarly, MCB (leveraging D-LinUCB) is controlled by a discount factor, $\gamma = 1 - \mu$, where adaptiveness decreases as γ approaches 1. Ideally, these parameters should be tuned to each problem instance, which is challenging without prior knowledge of the rate at which experts change.

To address this, we propose a two-level approach, as illustrated in Figure 2. In practice, we maintain a set of bandit-with-expert-advice algorithms (the base learners) of varying degrees of non-stationarity, and use a meta learner to identify the best level of non-stationarity. Hence, while each base learner operates as if the experts change at the rate dictated by its parameter μ, the meta algorithm aggregates these individual learners into one

policy tailored to the actual rate of change. This allows the system to dynamically adjust its adaptiveness to the observed changes. For example, if experts change rapidly, the meta learner will assign more weight to base learners that favor recent experiences, while, in contrast, if experts change slowly, it will favor base learners with low levels of adaptiveness.

The concept of meta algorithms for managing non-stationary bandits was previously explored by [27] and built upon the work of [3] who introduced CORRAL, which learns to act according to the best-performing bandit algorithm's policy. Specifically, at time t, CORRAL randomly samples a learner m with probability p_t^m and then chooses the arm selected by that learner.

While CORRAL's design — which only requires chosen arms — allows it to be applied over any bandit algorithm, it potentially overlooks the more nuanced information some bandit algorithms can provide. D-LinUCB (and thus also MCB as presented in Section 3), for instance, can estimate the expected reward for each arm at every timestep, offering nuance that CORRAL does not exploit. This limitation motivates our introduction of the alternative approach we lay out in the next section.

5.1 Value-Based Corralling

Many bandit algorithms, such as the classical UCB [19, 4], Thompson Sampling [34], LinUCB [13], as well as the non-stationary algorithm presented in this work, MCB with D-LinUCB, act on a combination of two values: the reward estimate and an exploration term. UCB for example selects the arm k that maximizes the index $I_{k,t} = \bar{r}_{k,t} + \sqrt{\frac{\log(t)}{C_{k,t}}}$, where $\bar{r}_{k,t}$ is the estimate of arm k's reward after t timesteps, and the second term is a confidence interval around this estimate.

When applying CORRAL over a set of algorithms that each maintain such indices, we in practice select learner m's indices with probability p_t^m and then select the arm with the highest index. Because this approach discards the nuance present in these indices, we propose as an alternative to act on a linear combination of all base learners' indices.

In particular, let $\bar{r}_{k,t}^m$ be base learner m's estimate for arm k at time t, and let $c_{k,t}^m$ be that same learner's exploratory bonus for that same arm. We propose to maintain a weight w_t^m for each learner, which we use to compute a combined index:

$$I_{k,t} = \sum_m w_t^m \left(\bar{r}_{k,t}^m + c_{k,t}^m\right)$$

Within this sum, $\sum_m w_t^m \bar{r}_{k,t}^m$ can be interpreted as a combined estimate of arm k's expected reward. The weights $\mathbf{w}_t = \{w_t^1, ..., w_t^m\}$ can therefore be chosen by minimizing the error of this estimate:

$$\mathbf{w}_t = \arg\min_{\mathbf{w}} \mathbb{E}\left[\left(r_{k,t} - \sum_m w^m \bar{r}_{k,t}^m\right)^2\right] \tag{7}$$

Because the combined index differs from the base learners' indices, the resulting policy and reward signal are biased relative to those of the base learners. Therefore, feeding this biased reward signal back to the base learners would skew their learning process.

To address this, CORRAL uses inverse propensity weighting to construct feedback that individual learners can use to update their policies. Specifically, the feedback for learner m takes the form $\frac{r_{k_t,t}}{p_t^{m_t}} \mathbb{1}\{m_t = m\}$, where m_t is the learner chosen by CORRAL at time t.

Since our proposed approach does not maintain a probability distribution over the base learners, we instead compute inverse weights based on the arms. Specifically, let $p_{k,t}^m$ be base learner m's probability for arm k at time t. And let $p_{k,t}$ be the probability of the meta algorithm for arm k. To adapt the feedback obtained by the

meta algorithm's policy, each base learner m should then weigh the observation at time t by $\frac{p^m_{k,t}}{p_{k,t}}$. Since the probability of selecting arm k at time t is $p_{k,t}$, this ensures that the reward observed by learner m matches its policy: $\mathbb{E}[r^m_t] = \sum_k \frac{p^m_{k,t}}{p_{k,t}} r_{k,t} p_{k,t} = \sum_k p^m_{k,t} r_{k,t}$

The resulting algorithm, which we outline in Algorithm 1, significantly departs from CORRAL by fully exploiting the base learners' estimates. We refer to this novel approach as CORVAL (**COR**ralling through **VAL**ue Aggregation).

Algorithm 1 CORVAL

Require: number of base learners M

$\mathbf{w}_1 = \mathbf{0}^M$
for $m = 1, 2, ..., M$ **do**
    Initialize m-th learner with adaptiveness $\mu_i \propto 2^{-m}$ (e.g., MCB with $\gamma = 1 - \mu$)
for $t = 1, 2, ..., T$ **do**
    each learner m observes expert advice and uses it to estimate the reward vector $\bar{\mathbf{r}}^m_t$ as well as an exploration term $\mathbf{c}^m_t$
    select arm $k_t = \arg\max_k \sum_m w^m_t (\bar{r}^m_{k,t} + c^m_{k,t})$
    pull arm k_t and collect resulting reward r_t
    $\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \sum_{t'=1}^{t} \left(r_{k_{t'},t'} - \sum_m w^m \bar{r}^m_{k_{t'},t'}\right)^2 + \lambda ||\mathbf{w}||^2$ ▷ compute empirically optimal weights, e.g., through Ridge regression
    for $m = 1, 2, ..., M$ **do**
        Perform update of learner m with reward r_t weighted by $\frac{p^m_{k,t}}{p_{k,t}}$

6 Empirical Validation

To validate our algorithm in practice, we first present extensive empirical results on synthetic problems, and then confirm its viability on a real-world problem of active learning.

6.1 Synthetic Setting

In order to demonstrate the performance of our algorithms in different settings, we evaluate them on a spectrum of bandits that differ in their reward distributions. In particular, given K arms, we randomly assign the values $\{(\frac{0}{K-1})^s, (\frac{1}{K-1})^s, ..., (\frac{K-1}{K-1})^s\}$ to the arms, where $s \sim \mathcal{U}(0, 4)$ controls how strongly the reward is concentrated on the best arm. This results in reward distributions which cover a spectrum whose extremes are characterized by either small increments between arms or rewards concentrated on a single arm, as illustrated in Figure 3. In turn, this provides learners with more or less sparse feedback.

6.1.1 Temporal Non-stationarity.

For a given period ($\tau \in \{100, 500, 2500\}$), we generate non-stationary expertise by averaging 100 randomly sampled sine waves each with period $\hat{\tau} \sim \mathcal{N}(\tau, \tau/2)$. This produces expertise that displays both small and large changes in expertise, as illustrated in Figure 3. At any time step, the expertise value gives the probability that an expert provides honest (i.e., better than random) advice.

We evaluate the performance of the chosen algorithms in terms of reward over $T = 5000$ steps averaged over 200 simulations[4]. Alongside averages, we show 95% simultaneous confidence bands (see Supplementary Section

[4]Code to reproduce these results is available at https://github.com/axelabels/CDM_NONSTAT.

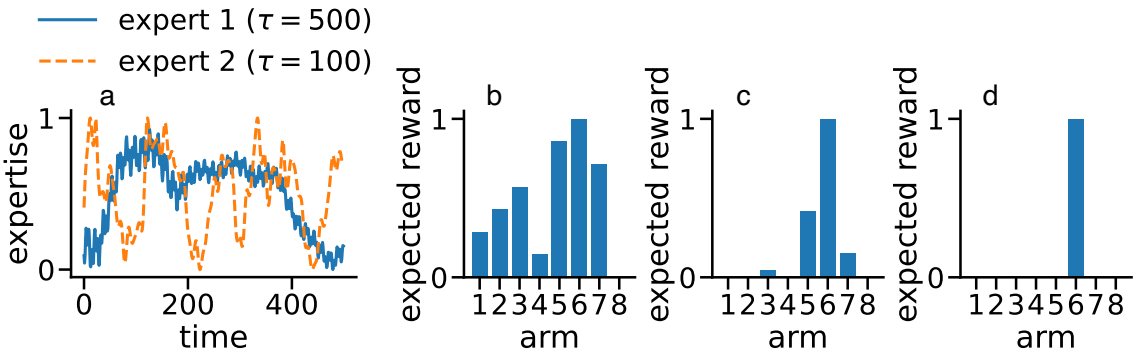

Fig. 3. a. Expertise over time for different periods (τ). b-d. three illustrative reward distributions, corresponding respectively to $s = 0$, $s = 2.5$, and $s = 5$ (see Section 6.1). d. in particular is akin to a classification task wherein a singular arm provides a reward.

C). We compare CORRAL and CORVAL to the MCB base learners, as well as to the best expert. Comparisons to EXP4.S and D-TS are provided in the supplementary material.

6.1.2 *Results and Discussion.*

How Stationary Algorithms Fail to Adapt to Changes. We show in Figure 4 how the performance of the selected algorithms changes over time and contrast this with the performance of the best moving expert. First, we observe that when experts are non-stationary (Figure 4a-c), the stationary MCB[$\mu = 0$] algorithm degrades towards random performance. This is in contrast with the static setting (Figure 4d) where the stationary variant is optimal. As the rate of change of experts increases (lower periods), different variants of MCB perform strongest. For longer periods (Figure 4c), the less dynamic MCB[$\mu = 0.005$] performs strongly, while for shorter periods (Figure 4a-b), MCB[$\mu = 0.05$] achieves higher rewards. In contrast, when the algorithm is too adaptive (MCB[$\mu = 0.5$]), it fails to learn at all, resulting in consistently poor performance.

The CORRAL algorithm tries to automatically identify which of these variants is strongest, and as a result it tends to converge towards the strongest variant. In contrast, CORVAL is not limited to selecting a single variant, but instead acts on a combination of MCB variants, enabling it to surpass the best variant for any given period. Interestingly, this ultimately allows CORVAL to surpass the single best expert, even when the best MCB variant fails to pass this same threshold (e.g., Figure 4a). Finally, as the period decreases, the problem becomes more challenging, as shown by the reduced convergence speed of CORVAL and CORRAL.

Collective Intelligence. Figure 5 compares the performance of the different algorithms to that of the best expert as we increase the number of experts. This figure shows that CORVAL typically surpasses the performance of the single best expert. In contrast, CORRAL and MCB variants only surpass the best expert for longer periods. In general, as we increase the number of experts, there is a trade-off between increasing the problem's difficulty (in the sense that more experts make it harder to find the single best combination of experts), and boosting the collective's potential (as a larger number of experts can lead to superior collective intelligence). Our results indicate that the benefits generally outweigh the increased complexity, except for the shortest period (100). In this case, the additional complexity introduced by rapid changes leads to decreased performance for larger group sizes.

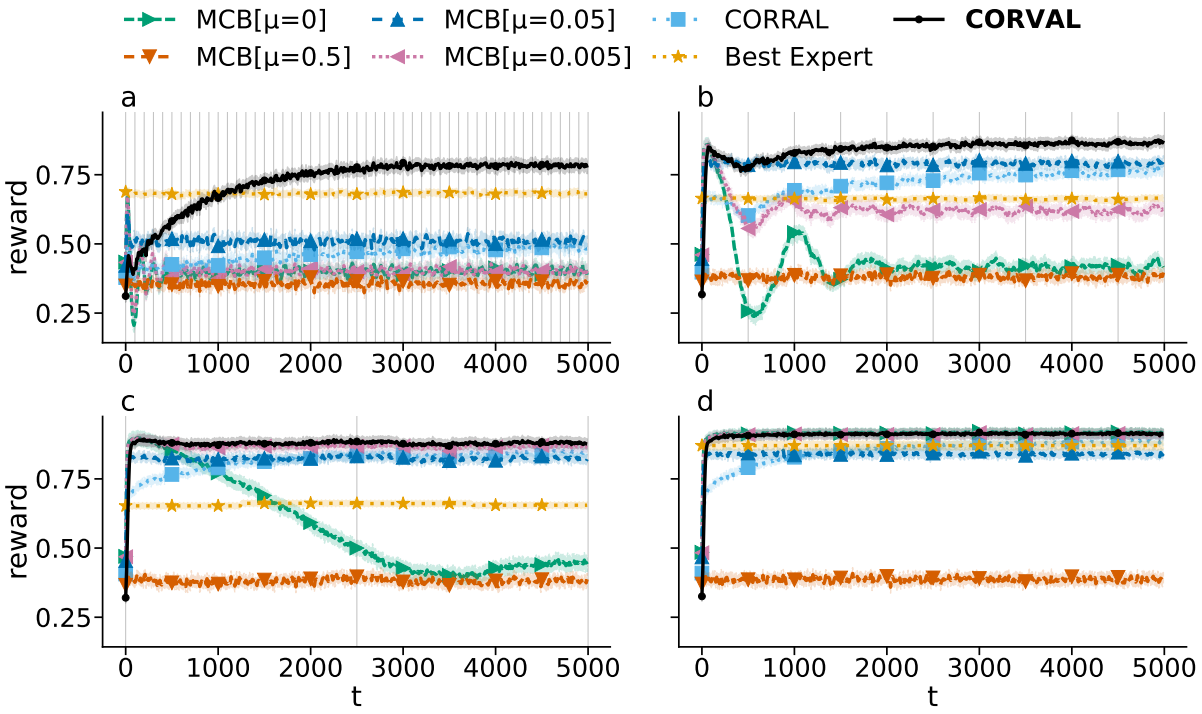

Fig. 4. Reward as a function of time for different periods averaged over all configurations. Vertical lines indicate periodicity (a. 100, b. 500, c. 2500, d. static). The dashed orange line marks the expected performance of the best moving expert. Shaded areas contain the 95% simultaneous confidence bands.

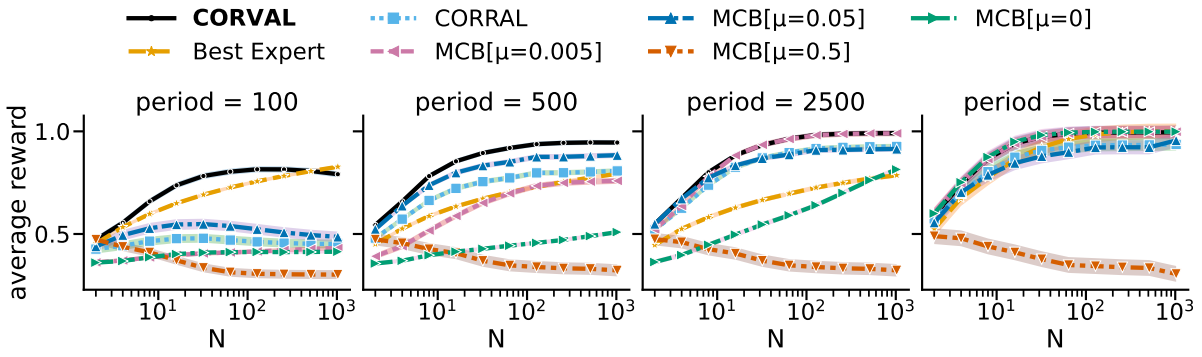

Fig. 5. Average reward ($\overline{R}$ as defined in Section 2.1) over $T = 5000$ steps for increasing expert counts (N). Shaded areas contain the 95% simultaneous confidence bands.

Impact of Misspecification. While we assumed in Section 3 that the true outcome could be estimated by a linear combination of expert advice, real-world problems will often involve some degree of misspecification [18]. This

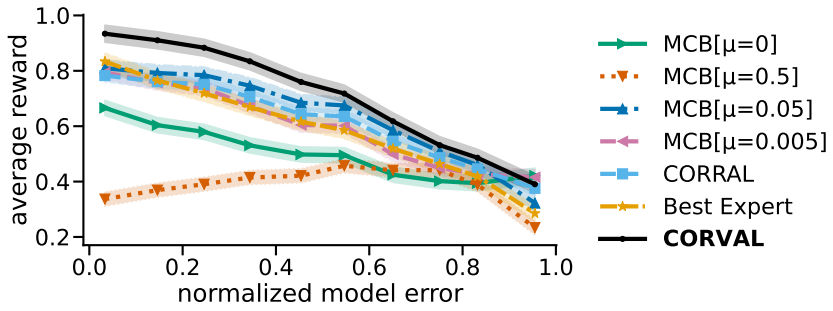

Fig. 6. Average reward ($\overline{R}$ as defined in Section 2.1) as a function of normalized model error (see Equation 8). The dashed orange line marks the expected performance of the best expert. Shaded areas contain the 95% simultaneous confidence bands.

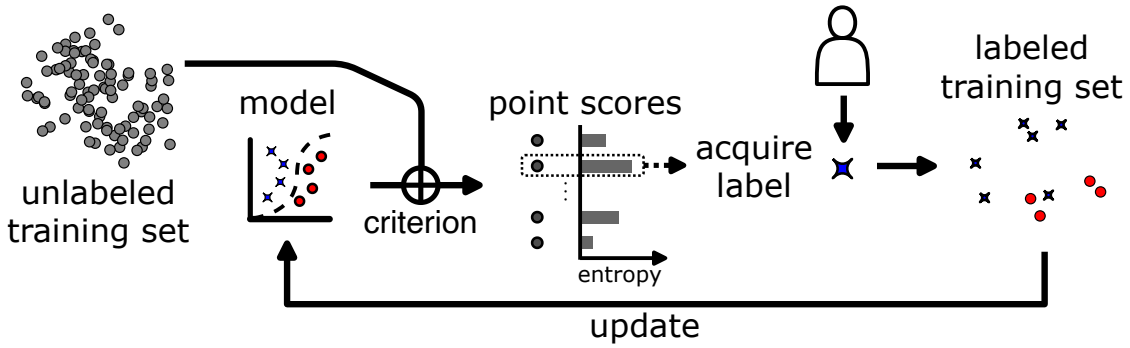

Fig. 7. Active learning diagram. When labeling the entire training set is too costly, active learning focuses on acquiring labels for the most informative data points. This can be done by evaluating the points according to some criterion, such as the model's entropy for each point. The point with the highest score is labeled (often by a human expert) and incorporated into the training set, which is then used to update the model, repeating the cycle.

misspecification measures how strongly rewards diverge from the best linear model. In particular, let $\vec{\theta}_t^*$ be the optimal weights at time t, i.e., those minimizing the error at that timestep (see Equation 3). We quantify the normalized model error as:

$$\frac{1}{KT}\sum_{t=1}^{T}\sum_{k\in[K]}\frac{|f(k,\vec{x}_t)-\langle\vec{f}_{k,t},\vec{\theta}_t^*\rangle|}{|f(k,\vec{x}_t)-\mathbb{E}[f(k',\vec{x}')]|} \tag{8}$$

Where the denominator is the expected error of a naive model which always predicts the expected reward.

Figure 6 shows how performance degrades as the error of the best possible linear model increases. In particular, for the highest levels, performance degrades to random performance. Note, however, that in such a configuration, expert advice generally has no correlation with the outcomes, making any bandit-with-expert-advice algorithm ineffective.

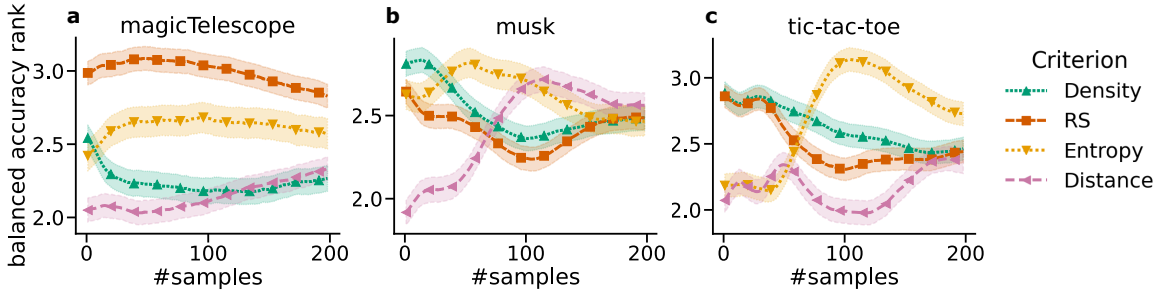

Fig. 8. Ranked balanced accuracy as a function of the number of samples for each criterion. For three illustrative data sets, these plots rank the balanced test accuracy we would obtain if we were to follow the given criterion at the given number of samples. As the number of observed samples increases, the ranking of the different criteria can shift, suggesting non-stationarity. In addition, the strongest criteria are dependent on the data set. See Section 6.2.1 for a description of these criteria.

6.2 An Application to Active Learning

When fully labeling a data set is too expensive, instances for which a label is queried must be carefully selected. Active Learning [33] tackles this problem by iteratively querying new labels on which the existing model can be trained further (see Figure 7). Depending on the problem, different querying strategies can be successful [14, 30, 29]. For example, the uncertainty criterion [24] queries points for which the current model's uncertainty is maximized. Another strategy consists of querying points which are most distant from the already queried points. However, it is typically impossible to know in advance which criterion is most suited to the problem at hand. What is more, as training progresses, the best criterion might change [14, 30]. For example, some criteria can initially provide a rapid increase in accuracy, while other criteria provide fine-tuning in the later stages (see Figure 8).

Methods such as DUAL [14] attempt to accommodate this change by switching from one strategy to another in a fixed manner. Similarly, Pang et al. 2018 frame this problem as one of bandits with expert advice, and apply a non-stationary variant of EXP4 to dynamically switch to the most appropriate criterion as training progresses. This latter approach therefore leverages bandits with expert advice algorithms to solve the active learning problem. As outlined in Algorithm 2, this can be achieved by (i) considering unlabeled points as arms of a bandit, (ii) mapping scores provided by criteria to an advice matrix which can be utilized by expert advice algorithms, and (iii) computing a reward which is conducive to high accuracy on the test set. For this third step, previous works have used training set accuracy as the reward signal [30]. However, we argue this signal is flawed for two main reasons. First, an improvement in accuracy typically results from the combination of several queries. By using accuracy as a reward signal, only the last query in a combination is rewarded. In addition, accuracy on the training set often differs from the accuracy on the test set, and improvements in training accuracy do not necessarily reflect improvements in the test set, especially for the relatively small sample sizes involved in active learning. Therefore, based on the intuition that points whose true labels differ from the model's predictions are informative, we propose to use surprise as a reward signal. Specifically, let c be the true label of a queried point, and let p_c be the model's predicted probability that the point belongs to label c. We define the reward as $r = (1 - p_c)\mathbb{1}\{p_c < 1/2\}$. In particular, this reward is 0 if the model correctly assigns the point to its true class, and otherwise is large when the model's error for that point is large. Our experimental comparison of this surprise reward with the training accuracy reward — given in supplementary Figure 14 — confirms the benefits of this alternative reward signal.

Algorithm 2 Active learning as a bandit with expert advice

Require: unlabeled data set $\mathcal{U}$, labeled data set $\mathcal{L}$, test data set $\mathcal{T}$, N criteria, a classifier

Sample initial point p_0 and query its label l_0
$\mathcal{U} \leftarrow \mathcal{U} \setminus \{p_0\}$, $\mathcal{L} \leftarrow \mathcal{L} \cup \{(p_0, l_0)\}$ ▷ add labeled pair to history
Initialize classifier
Initialize experts algorithm (e.g., EXP4.S)
Initialize N criteria
for $t = 1, 2, ..., T$ **do**
    Form advice matrix from criterion scores $\tilde{\boldsymbol{f}}_t = \{\vec{s}^1(\mathcal{L}, \mathcal{U}), ..., \vec{s}^N(\mathcal{L}, \mathcal{U})\}$
    Experts algorithm chooses arm (i.e., point p_t) and queries its label l_t
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{p_t\}$, $\mathcal{L} \leftarrow \mathcal{L} \cup \{(p_t, l_t)\}$ ▷ add labeled pair to history
    Update classifier and estimate balanced accuracy $\tilde{a}_t$ on $\mathcal{L}$
    Update experts algorithm with $r_t = \tilde{a}_t$

6.2.1 *Experimental Setting.*

Following previous works [30, 12, 23, 22], we evaluate performance on 20 data sets which have previously been used to evaluate active learning approaches, namely bank-marketing, calhousing, cod-rna, credit-g, diabetes, eeg-eye-state, electricity, ibn-sina, ijcnn1, kc2, kdd99_10perc, magicTelescope, mozilla4, musk, ozone-level-8hr, qsar-biodeg, steel-plates-fault, svmguide3, tic-tac-toe, and zebra. These data sets present a wide variety in the number of features and instances and in their instance distributions. This in turn induces a wide variety in the usefulness of different criteria, both across data sets and across time. For each data set, we set aside $1/3^{rd}$ of the data points as test set and run the active learning set-up for 100 steps on the remaining data. Following Chu and Lin 2016, we train a Logistic Regression classifier [21]. We measure and report performance in terms of balanced accuracy on the test set averaged over 400 simulations per data set. At each time step, the following criteria (i.e., experts) provide advice (also known as a score in active learning) on which point to query next in order to maximize the classifier's accuracy on the complete data set.

- Entropy: A point's score is proportional to the classifier's entropy for that point, i.e., let p_i^c be the classifier's estimated probability that point i belongs to class c, that point's entropy is $-\sum_c p_i^c \log(p_i^c)$ [20].
- Distance: A point's score is its distance to the nearest already labeled points [9].
- Density: Uses gaussian mixture models to estimate high density regions of the problem space and uses that density as a score [14].
- Representative Sampling (RS): High uncertainty points are assigned as score their negated distance to the high uncertainty centroid [41]. That is, the high uncertainty point which is closest to all other high uncertainty points. Points for which certainty is high (distance > 1) are assigned a score of $-\infty$.
- Random: All points are assigned the same score.

6.2.2 *Results and Discussion.*

To estimate the improvement in performance brought by non-stationarity, we compare the performance of selected algorithms in Figure 9. Balanced accuracy scores show that in general, our approach outperforms other approaches in this task. Figure 9**a** in particular shows that on average CORRAL and the MCB alternatives perform similarly, but CORVAL is able to outperform them. This suggests that while no single adaptiveness level is superior on average, by tailoring the adaptiveness to the problem at hand, CORVAL benefits. Figure 9**b** further compares CORVAL with the individual criteria. In particular, this shows that on average, CORVAL is able to outperform all criteria. Note that while the performance in Figure 9 is relatively uniform, there is a large heterogeneity from data set to data set, as illustrated by supplementary Figures 12 and 13. In particular, those detailed plots confirm

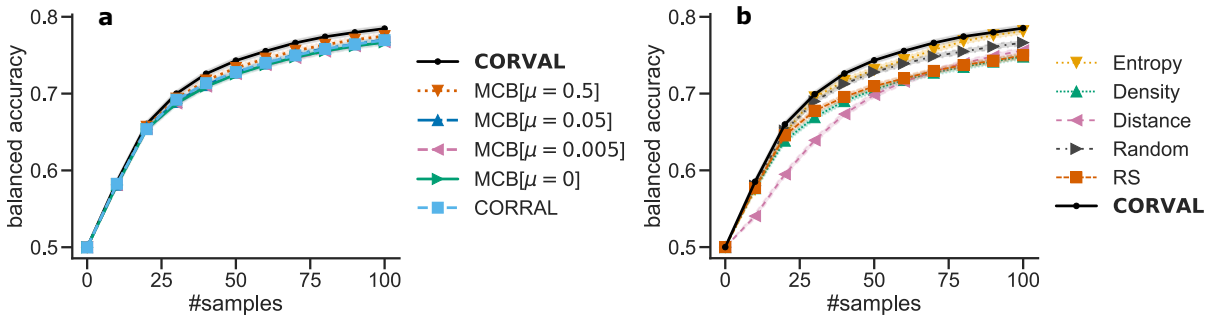

Fig. 9. Balanced accuracy as a function of the number of samples averaged over all data sets. **a.** contrasts the selected algorithms, while **b.** contrasts CORVAL with the individual criteria (dashed lines). Shaded areas contain the 95% simultaneous confidence bands.

that (i) there is not one single criterion that performs strongly for all data sets and (ii) there is some degree of non-stationarity, as the performance of criteria changes over time. By learning to aggregate the criteria while taking into account this non-stationarity, CORVAL learns to match the strongest criteria over time, resulting in overal improved performance.

7 Conclusion

In this work we propose a novel approach for aggregating the advice of non-stationary expert collectives when optimizing bandit problems. Methods which account for temporal non-stationarity are essential in real-world settings in which experts are likely to demonstrate changing ability. For groups of human experts, for example, continuous learning can induce a shift in performance in some or all experts. To handle such shifts, we first proposed MCB[μ], which makes use of parameter decay to continuously adapt its mapping from advice to outcomes in order to maintain strong performance throughout. In order to tune MCB's adaptiveness to the available expert set, we also proposed the CORVAL algorithm, which maintains multiple instances of MCB and dynamically aggregates their estimates. Whereas previous methods act on a single expert's advice, our approach is more collective, in the sense that several experts impact decisions. As a result, individual expert changes have a lower influence on its actions, leading to stronger performance. We empirically demonstrated this performance improvement on a wide variety of non-stationary configurations, across which our approach is able to consistently outperform other adaptive algorithms. This novel method has the potential to improve performance in decision-making tasks involving expert advice, whether human or artificial. Active learning, for example, can be framed as a problem of bandits with expert advice, and our results show improved performance enabled by our novel approach.

Acknowledgments

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: A.A. is supported by a post-doctoral grant (Chargé de Recherche) by the National Fund for Scientific Research (F.N.R.S.) of Belgium. T.L. is supported by the F.N.R.S. [grant numbers 31257234 and 40007793], the Fonds Wetenschappelijk Onderzoek (F.W.O.) [grant number G.0391.13N], the Service Public de Wallonie Recherche [grant n°2010235–ARIAC by DigitalWallonia4.ai]. T.L and A.N. benefit from the support of the Flemish Government through the AI Research Program. T.L., V.T and A.N. acknowledge the support by TAILOR, a project funded by EU Horizon 2020 research and innovation program [grant number 952215]. The resources and services

used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

References

[1] A. Abels, T. Lenaerts, V. Trianni, and A. Nowé. 2023. Dealing with expert bias in collective decision-making. *Artificial Intelligence*, 320, 103921. DOI: https://doi.org/10.1016/j.artint.2023.103921.

[2] A. Agarwal, M. Dudík, S. Kale, J. Langford, and R. Schapire. 2012. Contextual bandit learning with predictable rewards. In *Artificial Intelligence and Statistics*. PMLR, 19–26.

[3] A. Agarwal, H. Luo, B. Neyshabur, and R. E. Schapire. 2017. Corralling a band of bandit algorithms. In *Conference on Learning Theory*. PMLR, 12–38.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47, 235–256.

[5] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32, 1, 48–77.

[6] O. Besbes, Y. Gur, and A. Zeevi. 2014. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in neural information processing systems*, 27.

[7] A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R. Schapire. 2011. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 19–26.

[8] L. Breiman. 1996. Stacked regressions. *Machine learning*, 24, 1, 49–64.

[9] K. Brinker. 2003. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, 59–66.

[10] Y. Chen, C.-W. Lee, H. Luo, and C.-Y. Wei. 2019. A new algorithm for non-stationary contextual bandits: efficient, optimal and parameter-free. In *Conference on Learning Theory*. PMLR, 696–726.

[11] W. C. Cheung, D. Simchi-Levi, and R. Zhu. 2019. Learning to optimize under non-stationarity. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 1079–1087.

[12] H.-M. Chu and H.-T. Lin. 2016. Can active learning experience be transferred? In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 841–846.

[13] W. Chu, L. Li, L. Reyzin, and R. Schapire. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 208–214.

[14] P. Donmez, J. G. Carbonell, and P. N. Bennett. 2007. Dual strategy active learning. In *Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. Proceedings 18*. Springer, 116–127.

[15] D. Foster, A. Agarwal, M. Dudik, H. Luo, and R. Schapire. 2018. Practical contextual bandits with regression oracles. In *International Conference on Machine Learning*. PMLR, 1539–1548.

[16] P. Gaillard, G. Stoltz, and T. Van Erven. 2014. A second-order bound with excess losses. In *Conference on Learning Theory*. PMLR, 176–196.

[17] A. Garivier and E. Moulines. 2011. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*. Springer, 174–188.

[18] A. Ghosh, S. R. Chowdhury, and A. Gopalan. 2017. Misspecified linear bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence* number 1. Vol. 31.

[19] J. C. Gittins. 1979. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 41, 2, 148–164.

[20] A. Holub, P. Perona, and M. C. Burl. 2008. Entropy-based active learning for object recognition. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 1–8.

[21] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant. 2013. *Applied logistic regression*. John Wiley & Sons.

[22] W.-N. Hsu and H.-T. Lin. 2015. Active learning by learning. In *Proceedings of the AAAI Conference on Artificial Intelligence* number 1. Vol. 29.

[23] S.-J. Huang, R. Jin, and Z.-H. Zhou. 2010. Active learning by querying informative and representative examples. *Advances in neural information processing systems*, 23.

[24] D. D. Lewis and W. A. Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR'94*. Springer, 3–12.

[25] A. Likmeta, A. M. Metelli, G. Ramponi, A. Tirinzoni, M. Giuliani, and M. Restelli. 2021. Dealing with multiple experts and non-stationarity in inverse reinforcement learning: an application to real-life problems. *Machine Learning*, 110, 9, 2541–2576.

[26] H. Luo, C.-Y. Wei, A. Agarwal, and J. Langford. 2018. Efficient contextual bandits in non-stationary worlds. In *Conference On Learning Theory*. PMLR, 1739–1776.

[27] H. Luo, M. Zhang, P. Zhao, and Z.-H. Zhou. 2022. Corralling a larger band of bandits: a case study on switching regret for linear bandits. In *Conference on Learning Theory*. PMLR, 3635–3684.
[28] J. L. Montiel Olea and M. Plagborg-Møller. 2019. Simultaneous confidence bands: theory, implementation, and an application to svars. *Journal of Applied Econometrics*, 34, 1, 1–17.
[29] C. Nachtegael, J. De Stefani, and T. Lenaerts. 2023. A study of deep active learning methods to reduce labelling efforts in biomedical relation extraction. *PloS one*, 18, 12, e0292356.
[30] K. Pang, M. Dong, Y. Wu, and T. M. Hospedales. 2018. Dynamic ensemble active learning: a non-stationary bandit with expert advice. In *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2269–2276.
[31] V. Raj and S. Kalyani. 2017. Taming non-stationary bandits: a bayesian approach. *arXiv preprint arXiv:1707.09727*.
[32] Y. Russac, C. Vernade, and O. Cappé. 2019. Weighted linear bandits for non-stationary environments. *Advances in Neural Information Processing Systems*, 32, 12040–12049.
[33] B. Settles. 2009. Active learning literature survey.
[34] W. R. Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25, 3/4, 285–294. DOI: 10.2307/2332286.
[35] F. Trovo, S. Paladino, M. Restelli, and N. Gatti. 2020. Sliding-window thompson sampling for non-stationary settings. *Journal of Artificial Intelligence Research*, 68, 311–364.
[36] M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini. 2013. Finite-time analysis of kernelised contextual bandits. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, 654–663.
[37] S. Villar, J. Bowden, and J. Wason. 2015. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical Science*, 30, (May 2015), 199–215. DOI: 10.1214/14-STS504.
[38] C.-Y. Wei, Y.-T. Hong, and C.-J. Lu. 2016. Tracking the best expert in non-stationary stochastic environments. *Advances in neural information processing systems*, 29.
[39] S. J. Whitehead and S. Ali. 2010. Health outcomes in economic evaluation: the qaly and utilities. *British medical bulletin*, 96, 1, 5–21.
[40] Q. Wu, N. Iyer, and H. Wang. 2018. Learning contextual bandits in a non-stationary environment. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 495–504.
[41] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. 2003. Representative sampling for text classification using support vector machines. In *European conference on information retrieval*. Springer, 393–407.
[42] C. Zeng, Q. Wang, S. Mokhtari, and T. Li. 2016. Online context-aware recommendation with time varying multi-armed bandit. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2025–2034.

A Notation Table

Symbol	Description
K	Number of arms
N	Number of experts
M	Number of base learners
T	Number of timesteps
$\vec{x}_t$	Context vector at time t
k_t	Arm pulled at time t
$r_{k,t}$	Reward of arm k at time t
r_t	Reward at time t, shorthand for $r_{k_t,t}$
$\bar{R}$	Average reward t
$\mathcal{R}$	Cumulative reward t
R	Regret t
Δ	Experts' rate of change
Δ_θ	Linear model's rate of change
μ	Decay parameter
γ	Discount parameter
f	Expected reward function
$\tilde{f}$	Estimate of f
c^n	Expert n's confidence
$\tilde{f}^n$	Expert n's estimate of f
$\tilde{\mathbf{f}}_t$	Advice matrix at time t
$\vec{\tilde{f}}^n_t$	Expert n's advice vector at time t
$\tilde{f}^n_{k,t}$	Expert n's advice for arm k at time t
$\vec{\tilde{f}}_{k,t}$	Advice vector for arm k at time t
$\vec{\theta}_t$	Weight vector at time t
$\vec{\theta}^*$	Optimal weight vector
$\vec{\theta}_{t,n}$	Expert n's weight at time t
$O(\cdot)$	Asymptotic upper bound
$\Omega(\cdot)$	Asymptotic lower bound)
$\mathcal{U}(a,b)$	Uniform distribution over $[a,b]$
$\mathcal{N}(a,b)$	Normal distribution with location a and scale b
$[N]$	Set of integers from 1 to N
$p_{k,t}$	Probability of selecting arm k at time t
p^m_t	Probability of selecting base learner m at time t
$p^m_{k,t}$	Learner m's probability of selecting arm k at time t
$\bar{r}^m_t$	Learner m's reward vector estimate at time t
c^m_t	Learner m's exploration terms vector at time t

B On AdaBinGreedy's NonstatTest

The statistical test of AdaBinGreedy [26] verifies whether one reward estimate exceeds another reward estimate by at least $2(\alpha_A + \alpha_B)$:

$$\hat{\mathcal{R}}_A(\hat{\pi}_A) > \hat{\mathcal{R}}_A(\hat{\pi}_B) + 2(\alpha_A + \alpha_B) \tag{9}$$

Let $\mathcal{I}$ be some interval, the quantities $\alpha_{\mathcal{I}}$ and $\beta_{\mathcal{I}}$ for this interval are defined as follows:

$$\alpha_{\mathcal{I}} = 2\sqrt{\frac{Kln(4 * T^2N/\delta)}{|\mathcal{I}|}} + \frac{Kln(4 * T^2N/\delta)}{|\mathcal{I}|}$$

$$\beta_{\mathcal{I}} = 2\sqrt{\frac{Kln(4 * T^2N/\delta)}{\mu_{\mathcal{I}}|\mathcal{I}|}} + \frac{Kln(4 * T^2N/\delta)}{\mu_{\mathcal{I}}|\mathcal{I}|}$$

We note first that $\mu_{\mathcal{I}} \leq 1/K$, and thus $\alpha_A \leq \alpha_B$. In addition, since rewards are in $[0, 1]$, it follows that the reward estimate (barring any excessive variance introduced by importance sampling) is similarly bounded. Thus, for the statistical test to be passable, α_A should be no larger than $1/4$.

We further have that the size of any interval α is bounded by $\sqrt{T/2}$, wherein $T/2$ is the largest block size, and $\sqrt{T/2}$ is the size of a bin within the largest block.

Thus, even in the simplest case wherein $K = 2$,$N = 2$, and $\delta = 1$, we have that

$$\alpha_A \geq 2\sqrt{\frac{2ln(8T^2)}{\sqrt{T/2}}} + \frac{2ln(8T^2)}{\sqrt{T/2}}$$

For inequality (9) to not be trivially false, the right hand side of the above inequality should be lower than $1/4$, which requires that $T \gtrapprox 5.9e7$. For most real-world applications the bounds used in the statistical test are therefore too large to appropriately detect non-stationarity.

C Simultaneous Confidence Bands

Simultaneous confidence bands [28] provide global uncertainty quantification over an estimated function, ensuring with high probability (e.g., 95%) that the entire true function lies within the band. This contrasts with pointwise intervals, which only guarantee coverage at individual points.

A common method for constructing such bands is the sup-t approach [28], which calibrates the band width using the distribution of the maximum absolute deviation between the estimator and the true function. Let $\hat{g}(L)$ denote an estimator of the target function $g(L)$, evaluated at a finite set of values $\{L_1, \ldots, L_m\}$. In our setting, $\hat{g}(L)$ is the empirical mean of some quantity (e.g., average reward) for a given parameter L (e.g., number of experts).

The sup-t band around this mean takes the form

$$\hat{g}(L_i) \pm c, \quad \text{for all } i = 1, \ldots, m,$$

where the critical value c is chosen to ensure simultaneous coverage:

$$\mathbb{P}\left(\max_{1 \leq i \leq m} |\hat{g}(L_i) - g(L_i)| \leq c\right) \geq 1 - \alpha.$$

We estimate the critical value c using a bootstrap procedure. In this approach, repeated bootstrap samples are drawn from the original data, and the function is re-estimated on each sample to obtain a distribution of bootstrapped functions $\hat{g}^*(L_i)$. For each bootstrap draw, the maximal deviation

$$S^* = \max_{1 \leq i \leq m} |\hat{g}^*(L_i) - \hat{g}(L_i)|$$

is computed, and the $(1 - \alpha)$ quantile of the empirical distribution of S^* is taken as the estimated critical value $\hat{c}_{1-\alpha}$. The resulting band,

$$\hat{g}(L_i) \pm \hat{c}_{1-\alpha},$$

forms a symmetric, simultaneous confidence band with asymptotic coverage probability at least $1 - \alpha$.

D Additional Results

While our main text uses MetaCMAB as a base algorithm (MCB[$\mu = 0.05$] for example solves MetaCMAB through D-LinUCB with $\gamma = 1 - \mu$), alternative bandits with (non-stationary) expert advice algorithms exist. We here compare their performance to CORVAL as well as the MCB variants. In particular we consider D-TS and EXP4.S. We show both performance as a function of the number of experts (Figure 10) and as a function of the number of timesteps (Figure 11). These plots align with previous results in the stationary setting [1], where the stationary EXP4 tends to outperform the MAB reduction (here, D-TS), but is in turn outperformed by the Meta-CMAB approach (here, MCB). Interestingly, D-TS performs on par with EXP4.S for small expert sets, but quickly degrades for larger sets. This is because, in contrast with EXP4.S and MCB, D-TS only updates its estimates about one expert at each timestep. When the number of experts is too large, the number of steps required to update all experts becomes excessive.

In addition, for the active learning problem, Figures 12 and 13 show the performance detailed by dataset. These reveal that while CORVAL is not the strongest method on every individual dataset, it is consistently competitive with the best-performing criterion across all datasets, something no single baseline achieves. This consistency leads to strong average performance and highlights CORVAL's robustness across varied learning scenarios. In contrast, several other methods exhibit higher variance: although they may outperform CORVAL on specific datasets, they also suffer sharp failures on others (e.g., bank-marketing, ijcnn1 in the case of Entropy). CORVAL avoids these extremes, maintaining reliable performance throughout. Its strong average performance is thus not driven by dominance on any single task, but by consistently avoiding poor outcomes while remaining competitive overall.

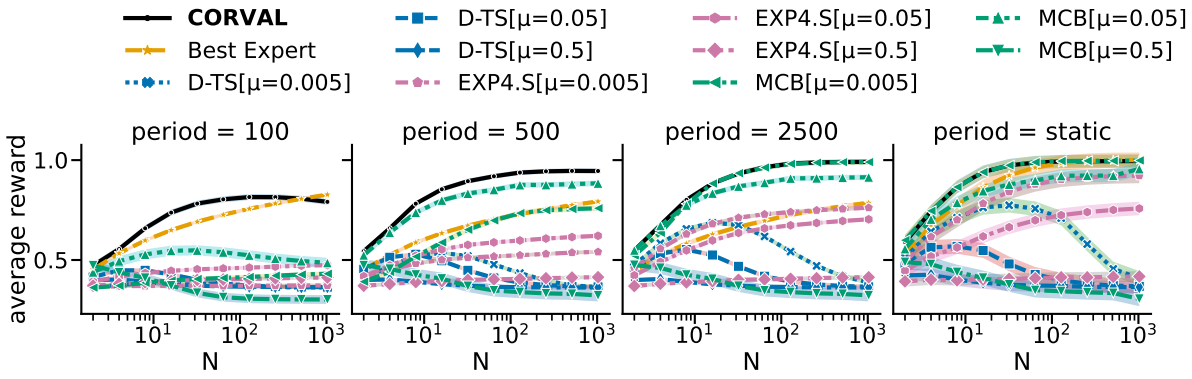

Fig. 10. Average reward ($\overline{R}$) over $T = 5000$ steps for increasing expert counts (N). Shaded areas contain the 95% simultaneous confidence bands.

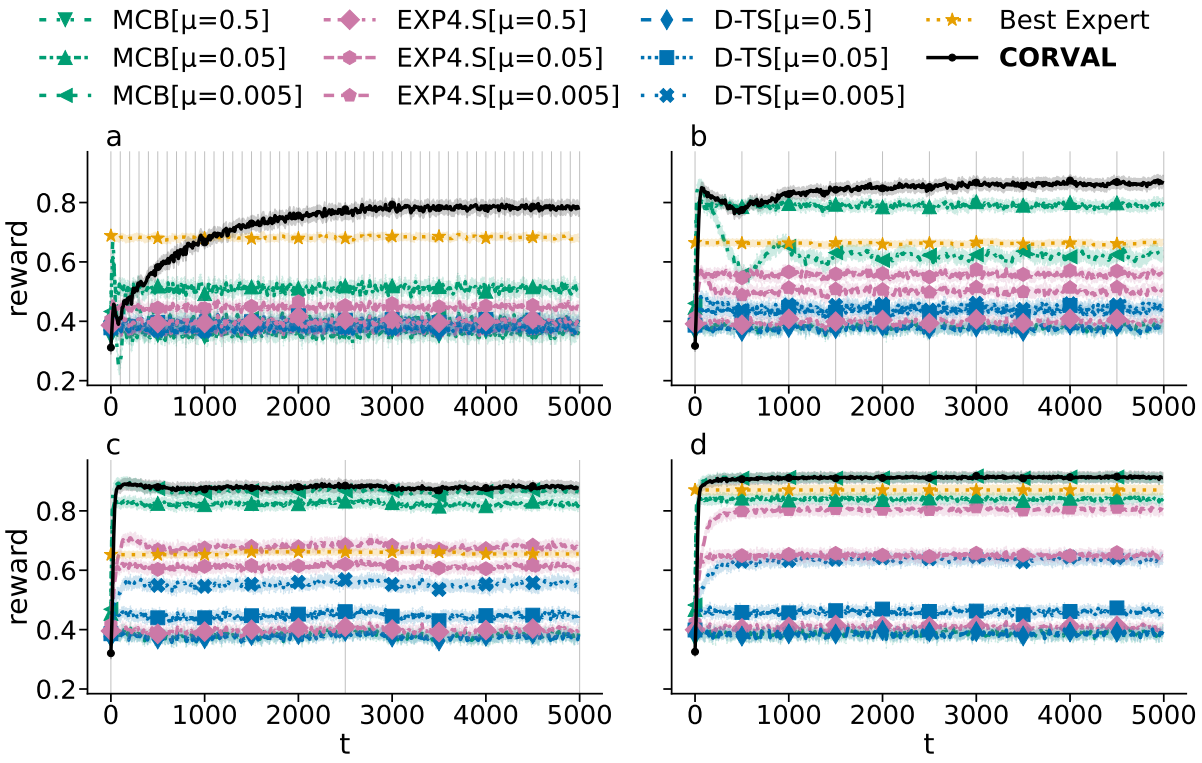

Fig. 11. Reward as a function of time for different periods averaged over all configurations. Vertical lines indicate periodicity (a. 20, b. 100, c. 500, d. static). The dashed orange line marks the expected performance of the best moving expert. Shaded areas contain the 95% simultaneous confidence bands.

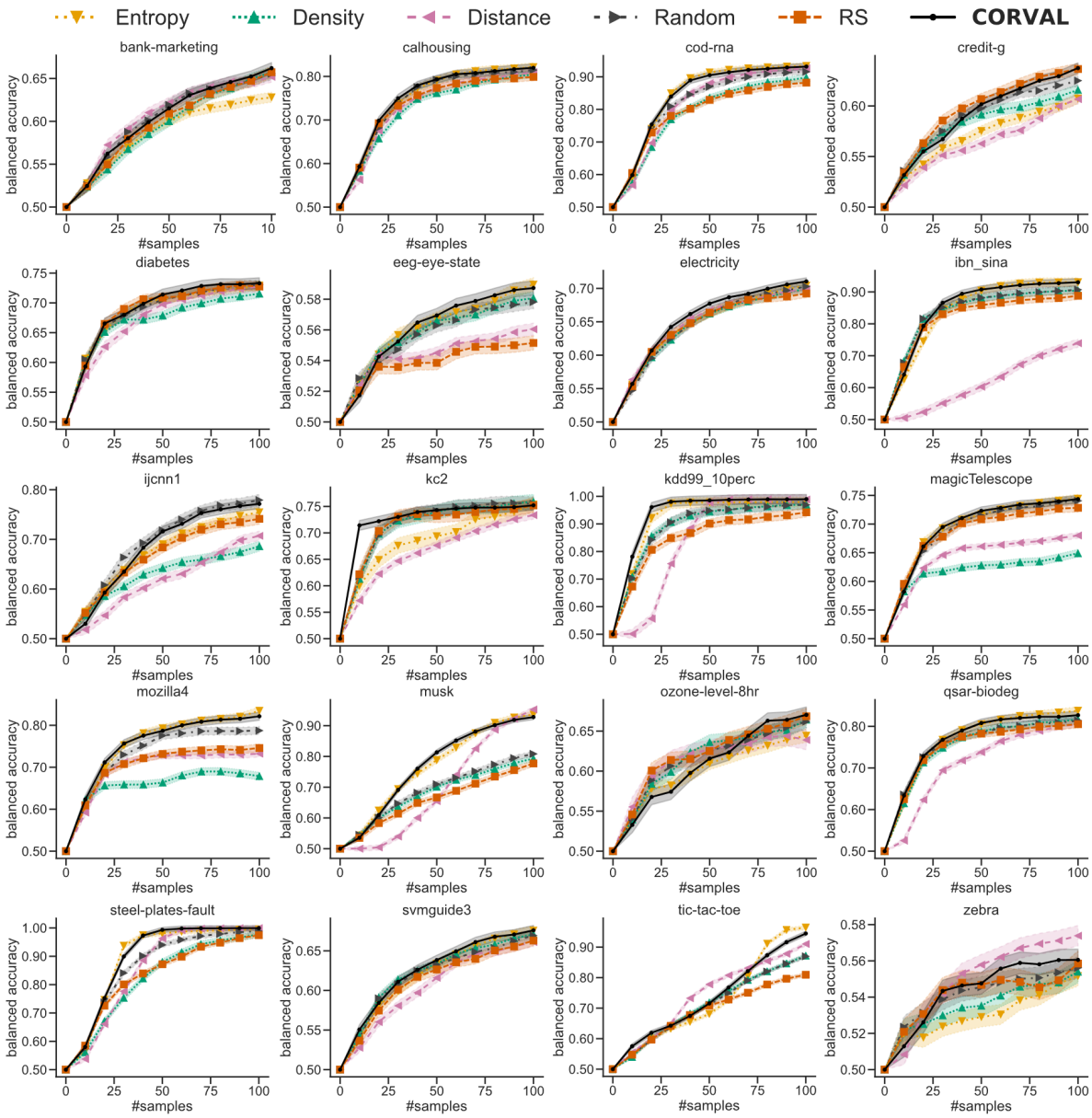

Fig. 12. Balanced accuracy as a function of the number of samples for each dataset.

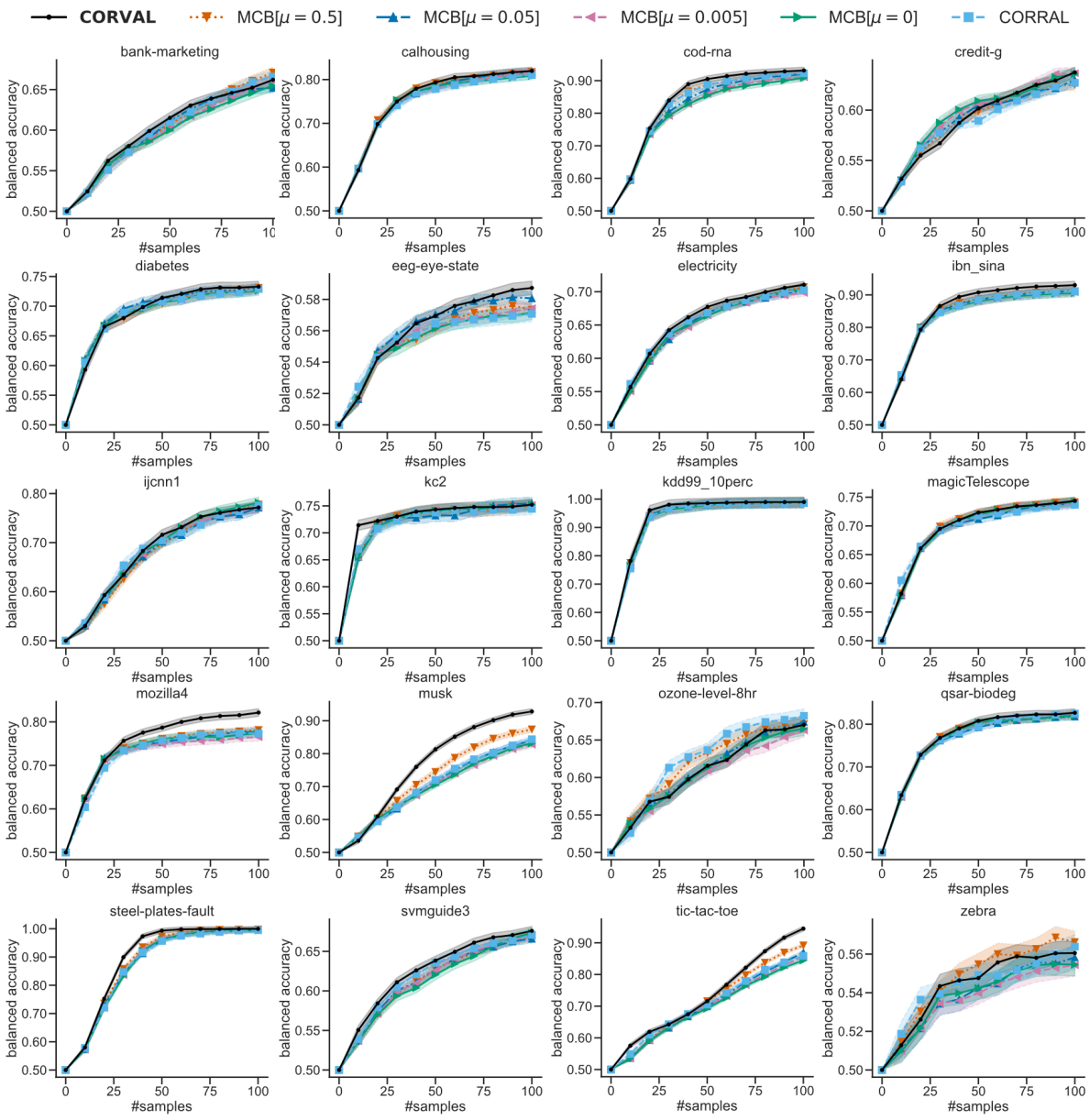

Fig. 13. Algorithms' balanced accuracy as a function of the number of samples for each dataset.

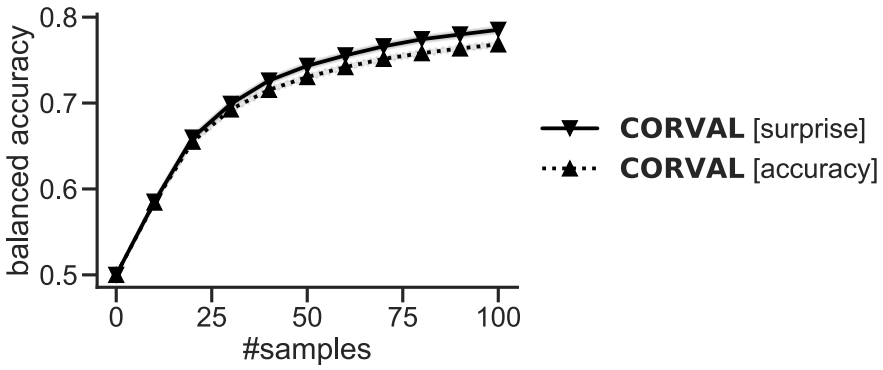

Fig. 14. Balanced accuracy as a function of the reward type and the number samples averaged over all datasets. Shaded areas contain the 95% simultaneous confidence bands.

Received 17 July 2024; revised 1 May 2025; accepted 9 May 2025