Power-of-2-Arms for Adversarial Bandit Learning With Switching Costs

Ming Shi, *Member, IEEE,* Xiaojun Lin, *Fellow, IEEE,* and Lei Jiao, *Member, IEEE*

***Abstract*—Motivated by edge computing with artificial intelligence, in this paper we study an adversarial bandit-learning problem with switching costs. Existing results in the literature either incur $\Theta(T^{\frac{2}{3}})$ regret with bandit feedback, or rely on free full-feedback in order to reduce the regret to $O(\sqrt{T})$. In contrast, we expand our study to incorporate two new factors. First, full feedback could incur a cost. Second, the player may choose 2 (or more) arms at a time and observe their feedback, even though switching costs are still incurred when she changes the set of chosen arms. For the setting where the player pulls only one arm at a time, our new regret lower-bound shows that, even when costly full-feedback is added, the $\Theta(T^{\frac{2}{3}})$ regret still cannot be improved. However, the dependence on the number of arms may be improved when the full-feedback cost is small. In contrast, for the setting where the player can choose 2 (or more) arms at a time, we provide a novel online learning algorithm that achieves a significantly lower regret equal to $O(\sqrt{T})$. Further, our new algorithm does not need any full feedback at all. This sharp difference therefore reveals the surprising power of choosing 2 (or more) arms for this type of bandit learning problems with switching costs. Both our new algorithm and regret analysis involve several new ideas in choosing the primary and secondary arms, tuning the weight-decay parameters within and across episodes, and using the loss differences in the weight updates, which may be of independent interest.**

***Index Terms*—Bandit learning, switching costs, regret analysis, edge computing with artificial intelligence.**

I. Introduction

In this paper, we are interested in bandit learning with switching costs, which can be used to model many practical decision-making problems that not only face significant uncertainty, but also incur costs for changing decisions. Consider edge computing with artificial intelligence (Edge AI) [2], [3] as an example, where an edge server close to the end users downloads machine learning (ML) models from the cloud to process incoming inference requests. As the underlying ground-truth model of the data changes in uncertain ways (which is often referred to as concept drift [4]), the best ML model also changes in time. However, because of the limited capability of the edge server, it can often only accommodate a small number of ML models. Thus, the edge server needs to learn which subset of ML models should be used, based on the feedback (e.g., inference losses) observed. Further, downloading an ML model (which is not currently on the edge server) from the cloud incurs communication overhead, which can be modelled by a switching cost β_1. Hence, the edge server has to carefully select the ML models to reduce the total inference losses and switching costs in the long run, which thus corresponds to a bandit learning problem with switching costs. Other examples of such problems can be found in transportation networks [5], wireless communication [6], recommendation systems [7], and robotics [8], etc.

In the online learning literature, it is well-known that the existence of switching costs significantly changes the nature of the regret. Specifically, in an adversarial setting (which we will focus on in this paper), for bandit learning *without* switching costs, the Exp3 algorithm can attain $O(\sqrt{T})$ regret over a time-horizon T [9]. However, once the switching cost is added, the regret (for the setting where only one arm can be pulled at each time) increases substantially to $\Theta(T^{\frac{2}{3}})$ [10]. A matching lower bound in [11] suggests that such an increased regret is unavoidable. While this result may be somewhat discouraging, it leaves many important open questions, as we explained below. Note that since ML models in Edge AI corresponds to arms in bandit learning, we use the word "model" and "arm" interchangeably in the rest of the paper.

First, in practice, in addition to pulling one arm, there are often other ways to obtain feedback. For example, in Edge AI, the edge server could send the data to the cloud for analysis. In this case, the feedback from all ML models can be obtained, beyond the model already deployed on the edge server. This is somewhat analogous to the full-feedback setting studied in [12]. Reference [12] shows that, if the full feedback can be obtained with zero costs, the regret for bandit learning with switching costs will remain at $O(\sqrt{T})$, which would have been much lower than that of [10] where only bandit feedback is available. However, in practice, feedback from the cloud also incurs non-negative costs due to multiple reasons, e.g., communication costs, latency and privacy issues [2], [3]. *Thus, the regret for bandit learning with both switching costs and full-feedback costs remains an open problem.*

Second, instead of holding only one ML model at each time, in Edge AI, the edge server can usually accommodate $M \geq 2$ ML models at each time. In this setting, at each time, the feedback from all M models currently on the edge server can be observed. This setting is thus most similar to a bandit learning problem with limited advice [13], where $M \geq 2$ arms can be chosen at each time. However, [13] only

This work has been partially supported by the U.S. National Science Foundation (through Grants CNS-2113893, CNS-2047719, CNS-2225950, and CNS-2225949). A preliminary version of this work has appeared in the Twenty-third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (ACM MobiHoc), October 17-20, 2022 [1].

Ming Shi is with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, USA (e-mail: shi.1796@osu.edu).

Xiaojun Lin is with Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA (e-mail: linx@ecn.purdue.edu). Part of this work was completed when he moved to The Chinese University of Hong Kong.

Lei Jiao is with the Center for Cyber Security and Privacy, University of Oregon, Eugene, OR 97403, USA (e-mail: ljiao2@uoregon.edu).

studied the case without switching costs, where the regret is $O(\sqrt{T})$ regardless of whether one ($M=1$) or more ($M\geq 2$) arms are chosen at each time. Our setting is also related to bandit learning problems with semi-bandit feedback [14] and side information [5]. The studies for semi-bandit feedback [14] typically do not consider switching costs either. Although the side-information setting [5] has been studied with switching costs, it is somewhat different from ours because the source of the side information is not controlled by the algorithm there. Partly due to this difference, the regret [5] remains at $\Theta(T^{\frac{2}{3}})$. *In summary, it remains an open problem whether in our setting, choosing $M\geq 2$ arms can improve the regret.*

In this paper, we provide new answers to the aforementioned two important open problems. First, we study the case when $M=1$, i.e., only one arm can be pulled at each time, and there is a switching cost β_1 to change the arm and a full-feedback cost β_2 to obtain feedback from all arms. As we discussed earlier, the latter action corresponds to the edge server sending data to the cloud for analysis. We provide a lower bound of the regret, which grows as $\Theta(T^{\frac{2}{3}})$. In other words, when only one arm can be pulled ($M=1$), adding costly full-feedback will not fundamentally change how regret depends on T. However, our lower bound does suggest that utilizing costly full-feedback may change the multiplication factor in front of $T^{\frac{2}{3}}$. In some settings, this factor can be reduced from $O(K^{\frac{1}{3}})$ to $O((\ln K)^{\frac{1}{3}})$, where K is the total number of arms. This lower bound is obtained by constructing two new adversaries (please see Sec. III-B) that force any online learning algorithm to either switch arms or use costly full-feedback for at least $\Omega(T^{\frac{2}{3}})$ number of times, in order to obtain a loss no greater than the optimal static loss plus $O(T^{\frac{2}{3}})$. The proof of the lower bound involves a non-trivial analysis of the Kullback-Leibler (KL) divergence (i.e., relative entropy) [15, p. 23] on a hidden Markov model, which is of independent interest. Moreover, we provide an algorithm that achieves a regret that matches the lower bound.

Second, we study the setting when $M\geq 2$, i.e., more than one arm can be chosen at each time and one of them is used to incur loss. The feedback of all M arms are then observed, while there are still switching costs and full-feedback costs. Specifically, in Sec. IV, we first start from the case where the switching cost is only for changing the set of the M chosen arms. In other words, there is no switching cost for picking the arm (from these M arms) that is used to incur loss. Surprisingly, here we provide a new online learning algorithm, called Randomized Online Learning With Working Groups (ROW), that can achieve a regret of $O(\sqrt{T})$ without even using full feedback (see Theorem 2), which significantly improves the $\Theta(T^{\frac{2}{3}})$ regret for $M=1$. In other words, having the flexibility to accommodate one additional model (i.e., $M=2$) almost brings comparable benefit as having free full-feedback [12]. *To the best of our knowledge, this sharp transition from $M=1$ to $M\geq 2$ has never be reported in the literature for bandit learning with switching costs*[1]. This may be seen as somewhat analogous to the "power-of-2" routing in load balancing [16] (where sampling two queues can attain comparable reduction to delay as sampling all queues), which is why we refer to it as the "power-of-2-arms". Moreover, as M increases, the regret of ROW further decreases. Using a trivial lower bound for bandit learning with free full-feedback [12], [17], we conclude that the dependence of the regret of our ROW algorithm on T must be optimal.

To achieve the improved $O(\sqrt{T})$ regret, ROW employs several new ideas. In order to fully utilize the flexibility of choosing $M\geq 2$ arms and minimize switching costs, the first idea of ROW is to fix a primary arm over $O(\sqrt{T})$ time-slots (which we refer to as an episode), and switch the secondary arms $\left\lceil\frac{K-1}{M-1}\right\rceil$ times during an episode, each time to a new subset of secondary arms that have not yet been chosen in this episode. In this way, ROW only makes a constant number of switches within each episode (and $\Theta(\sqrt{T})$ switches for all the time), but it can obtain not only the feedback of the primary arm for the entire episode, but also the feedback of every other arms for $\frac{1}{\left\lceil\frac{K-1}{M-1}\right\rceil}$ fraction of the episode. Intuitively, this way of obtaining feedback incurs much lower costs than using costly full-feedback to obtain the same amount of feedback (for any positive K and β_2 independent of T), which is also the reason that ROW does not use costly full-feedback at all. Note that such a saving is only possible when $M\geq 2$. As we discussed earlier, for $M=1$, either the switching cost or the full-feedback cost has to be $\Omega(T^{\frac{2}{3}})$ to attain a low loss.

However, just using the above idea alone is insufficient to produce the $O(\sqrt{T})$ regret. The reason is that the feedback obtained is highly correlated in time. This is because each subset of secondary arms is retained for the whole sub-episode (whose length is also $O(\sqrt{T})$). It is known that such correlation tends to increase the regret. Indeed, we construct two counter-examples in Sec. IV-A to show that, if we merely use episodic versions of existing bandit-learning algorithms, e.g., Exp3 [9], the regret will still be very high. To resolve this challenge, ROW utilizes a second crucial idea. Our key observation is that, whenever such a sub-episode with highly-correlated feedback occurs, one of arms in the current working group (either the primary arm or a secondary arm) will likely be consistently better than other arms. Then, ROW will try to switch to the better arm more quickly within the sub-episode, and thus improve the regret. To accomplish this faster switching within a sub-episode, our proposed ROW algorithm will use a larger weight-decay parameter η_2 within each sub-episode, while using a smaller parameter η_1 across episodes. In Sec. IV-B2, we give a sufficient condition on how much η_2 should be larger than η_1 to strike the right balance. We note that this idea of using two different weight-decay parameters is new and may be of independent interest.

Furthermore, since in each episode the primary arm will receive much more feedback than the secondary arms, this creates a bias in the overall quality of feedback at the end of each episode. This bias issue is resolved by using instead the loss differences between the primary and secondary arms (please see our Idea 3 in Sec. IV-A). Our proof for the $O(\sqrt{T})$ regret carefully captures the effect of the above ideas by utilizing several new techniques (please see Sec. IV-B for

[1] Note that for bandit learning *without* switching costs, choosing $M\geq 2$ arms will improve the regret, but it cannot alter the dependence on T [13].

details). Then, in Sec. V, we extend our results in Sec. IV to a more general case where there is an additional switching cost for changing the arm (even among the M chosen arms) that is used to incur loss.

Finally, using both a generic setting and a more realistic Edge-AI setting, our simulation results (in Sec. VI) demonstrate that our algorithms can significantly reduce the regret.

II. Problem Formulation

In this section, we provide the problem formulation for the bandit learning problem with switching costs and full-feedback costs that we consider. Moreover, we present a motivating example based on edge computing with artificial intelligence (Edge AI), which has received extensive attention recently [2], [3]. Finally, we introduce the performance metric.

A. Bandit Learning With Switching Costs and Full-Feedback Costs

A player interacts with the adversary/environment sequentially in time. We use $\mathcal{K} \triangleq \{1, 2, ..., K\}$ to denote the set of all arms and let M be an integer, $1 \leq M < K$. In each time-slot $t = 1, ..., T$, first the player chooses M arms among all K arms. Let $\hat{\mathbf{k}}(t)$ denote the set of the M arms chosen at time t. The player uses one of the arms in $\hat{\mathbf{k}}(t)$ as the active arm, which is denoted by $\mathrm{k}(t)$. The loss of this arm, $l_{\mathrm{k}(t)}(t)$, will be used to calculate the loss and regret later. In addition, the losses $l_k(t)$ of all arms $k \in \hat{\mathbf{k}}(t)$ are observed by the player. The loss $l_k(t)$ can be any arbitrary value in $[0, 1]$. In this paper, we study both the cases when $M = 1$ and $2 \leq M < K$. When $M = 1$, $\hat{\mathbf{k}}(t)$ only contains the active arm $\mathrm{k}(t)$ and only the loss of $\mathrm{k}(t)$ is observed. In this case, we simply say that the player "pulls" the single arm $\mathrm{k}(t)$ at time t. On the other hand, when $2 \leq M < K$, in addition to the loss of the active arm $\mathrm{k}(t)$, the losses of other $M-1$ arms in $\hat{\mathbf{k}}(t)$ are also observed.

We next present our model for the switching costs. We will first focus on the case where there is no switching cost changing the active arm from the current set $\hat{\mathbf{k}}(t)$. However, every time a new arm is added to the set $\hat{\mathbf{k}}(t)$, a switching cost $\beta_1 > 0$ will be incurred. (In Sec. V, we will generalize our results to the case with additional switching costs for changing the active arm $\mathrm{k}(t)$ in $\hat{\mathbf{k}}(t)$.) Thus, the switching cost at time t is $\beta_1 \sum_{k \in \hat{\mathbf{k}}(t)} \mathbf{1}_{\{k \notin \hat{\mathbf{k}}(t-1)\}}$, where $\mathbf{1}_E$ is an indicator function (i.e., $\mathbf{1}_E = 1$ if the event E is true, and $\mathbf{1}_E = 0$ otherwise). As typically assumed in bandit learning problems [5], [9], [11], [12], [18], we assume that $\hat{\mathbf{k}}(0) = \Phi$ is empty. In addition to the feedback from the M arms in $\hat{\mathbf{k}}(t)$, at each time t, the player can choose to obtain full feedback of time t for all the arms (including those not in $\hat{\mathbf{k}}(t)$) at a cost β_2. Let $z(t) = 1$ if the player chooses to obtain the full feedback at time t, and $z(t) = 0$ otherwise. Therefore, the total cost is

$$\begin{aligned}&\mathrm{Cost}(1:T)\\&\triangleq \sum_{t=1}^{T} \Big\{ l_{\mathrm{k}(t)}(t) + \beta_1 \sum_{k \in \hat{\mathbf{k}}(t)} \mathbf{1}_{\{k \notin \hat{\mathbf{k}}(t-1)\}} + \beta_2 z(t) \Big\}. \qquad (1)\end{aligned}$$

B. An Example Motivated by Edge AI

We consider an Edge AI setting where an edge server collaborates with a remote cloud. The edge server runs machine learning (ML) models on an online stream of input data to predict their labels. (For example, in an E-commerce recommendation system, the input data at each time contains the customer data, item data and web shop transactions, etc. The input data will be used by the edge server to return the recommendations, i.e., the predicted labels of what the customer is interested in.) We assume that K ML models are already trained and available in the remote cloud. However, due to the limited capability of the edge server, only M models can be deployed at the edge server at each time. Since it is unknown which ML model works best, the edge server needs to use the feedback (e.g., the actual product picked by the customer) to learn which subset of ML models it should deploy. (In practice, both the underlying distribution of the input data and the mapping from data to labels change in time due to the so-called concept drift [4]. Therefore, the best model(s) also changes in time. As a result, this learning process may be performed again after a concept drift.)

This learning process can be modelled as the bandit learning problem described in Sec. II-A. Each arm corresponds to one of the K ML models. At each time t, the edge server chooses the subset $\hat{\mathbf{k}}(t)$ of M models, which correspond to the M arms chosen in bandit learning. This subset $\hat{\mathbf{k}}(t)$ may be the same as the subset $\hat{\mathbf{k}}(t-1)$ chosen at last time $t-1$, or it may differ, in which case a switching cost $\beta_1 \sum_{k \in \hat{\mathbf{k}}(t)} \mathbf{1}_{\{k \notin \hat{\mathbf{k}}(t-1)\}}$ for downloading the ML models that are not currently on the edge server will be incurred. Note that this switching cost is assumed to be proportional to the number of ML models (which are not currently on the edge server) downloaded at time t. Then, the input data $\vec{X}(t)$ is revealed. The edge server will use the models in $\hat{\mathbf{k}}(t)$ to infer the label of $\vec{X}(t)$. Further, it will use the result $\vec{Y}'_{\mathrm{k}(t)}(t)$ of one of the models $\mathrm{k}(t) \in \hat{\mathbf{k}}(t)$, to return to the end user. This model $\mathrm{k}(t)$ then corresponds to the active arm in bandit learning. Next, the true label $\vec{Y}(t)$ of $\vec{X}(t)$ is revealed. The edge server can then calculate the inference loss $l_k(t)$ for each ML model $k \in \hat{\mathbf{k}}(t)$, based on the difference between the inferred label $\vec{Y}'_k(t)$ and the true label $\vec{Y}(t)$, e.g., using the squared loss (i.e., $l_k(t) = \|\vec{Y}(t) - \vec{Y}'_k(t)\|^2$) [19].

At the end of time t, the edge server may also choose to consult the cloud for the quality of all ML models. In that case, it sends the data $\vec{X}(t)$ to the cloud. After the cloud processes this data with all ML models $k \in \mathcal{K}$, the edge server can retrieve the inference-loss $l_k(t)$ of all the ML models. Clearly, it incurs additional computation/communication overhead to obtain such feedback from the cloud, which we model by the full-feedback cost β_2.

C. Performance Metric

We use the regret [9]–[12] as the performance metric. For an online learning algorithm π, its total cost $\mathrm{Cost}^{\pi}(1:T)$ is given by (1), which includes both switching costs and full-feedback costs. For the optimal static solution OPT, it knows the future losses in advance, and hence can choose only one arm/model throughout the time-horizon. The cost of OPT is then given by

$\text{Cost}^{\text{OPT}}(1:T) = \min_{k\in\mathcal{K}} \sum_{t=1}^{T} l_k(t) + \beta_1$, where there is only one switching cost β_1 at the beginning of the time-horizon, and there is no full-feedback cost. The regret of algorithm π is defined to be the worst-case difference between the expected total cost of algorithm π and the total cost of OPT, i.e.,

$$R^{\pi}(T) \triangleq \sup_{l_{1:K}(1:T)} \left\{ \mathbb{E}_{\pi}\left[\text{Cost}^{\pi}(1:T)\right] - \text{Cost}^{\text{OPT}}(1:T) \right\}, \tag{2}$$

where the expectation is taken over the possible randomness of the algorithm π, and $l_{1:K}(1:T)$ denotes the losses $l_k(t)$ of all arms $k \in [1, K]$ for all time $t \in [1, T]$. Our goal is to design an online learning algorithm with a regret as low as possible.

III. The Case of $M = 1$

In this section, we focus on the case when $M = 1$, i.e., the player (e.g., edge server) can pull only one arm (e.g., model) at each time. We are interested in studying whether adding full feedback with a cost β_2 can alter the regret of bandit learning with switching costs. Recall that in this case, the active arm $\mathrm{k}(t)$ is the only arm in $\hat{\mathbf{k}}(t)$. As we mentioned in the introduction, when full feedback is free, it has been shown in [12] that using full feedback will improve the regret from $\Theta(T^{\frac{2}{3}})$ to $O(\sqrt{T})$. However, since in our model the full feedback incurs a cost, it is no longer clear whether the regret can still be improved.

A. A Lower Bound on the Regret (When $M = 1$)

Our first main result shows that adding costly full-feedback will not change the dependence of the regret on T, but may change the multiplication factor as a function of K.

Theorem 1. *Consider bandit learning with switching costs and full-feedback costs introduced in Sec. II-A. When $M = 1$, the regret of any online algorithm π must be lower-bounded as follows,*

$$R^{\pi}(T) \geq \underline{R}^{\pi}(T) \triangleq \max\left\{ C_1 \beta_a^{\frac{1}{3}} \left(\log_2 K\right)^{\frac{1}{3}} T^{\frac{2}{3}}, C_2 \beta_b^{\frac{1}{3}} T^{\frac{2}{3}} \right\}, \tag{3}$$

where

$$\beta_a = \min\left\{\frac{3}{2}\beta_1, \beta_2\right\},\ \beta_b = \min\left\{\frac{3}{4}K\beta_1, \beta_2\right\},$$

$$C_1 = \sqrt[3]{\frac{2}{9\ln 2}} \cdot \frac{1}{144\left(\log_2 T - \log_2 \log_2 K\right)},\ \textit{and}$$

$$C_2 = \sqrt[3]{\frac{2}{9\ln 2}} \cdot \frac{1}{144 \log_2 T}.$$

We can see from Theorem 1 that, even when the costly full-feedback is added, as long as $M = 1$, $\Theta(T^{\frac{2}{3}})$ is still the optimal regret for bandit learning with switching costs. This is in sharp contrast to the case of free full-feedback [12], where the regret can be improved to $O(\sqrt{T})$. While this result may be somewhat discouraging, the costly full-feedback does play some role in the multiplication factor in front of $T^{\frac{2}{3}}$, which depends on the relative magnitude of β_1 and β_2. Intuitively, when the full-feedback cost β_2 is large, the online learning algorithm would rather switch to obtain feedback than using costly full-feedback. On the other hand, when β_2 is small, the online learning algorithm should avoid switching and obtain feedback from costly full-feedback. Thus, we expect that costly full-feedback will be more useful in the latter case than in the former case. The conclusion of Theorem 1 shows this difference precisely. Specifically, we can make the following observations.

(i) When $\beta_2 \geq \frac{3}{4}K\beta_1$, the lower bound $\underline{R}^{\pi}(T)$ in (3) is equal to

$$\max\left\{ C_1 \left(\frac{3}{2}\beta_1\right)^{\frac{1}{3}} \left(\log_2 K\right)^{\frac{1}{3}} T^{\frac{2}{3}}, C_2 \left(\frac{3}{4}\beta_1\right)^{\frac{1}{3}} K^{\frac{1}{3}} T^{\frac{2}{3}} \right\}. \tag{4}$$

As K increases, the second term in (4) quickly dominates. This means that, when the full-feedback cost β_2 is high, the regret of any online learning algorithm π will at least increase as $\beta_1^{\frac{1}{3}} K^{\frac{1}{3}} T^{\frac{2}{3}}$. Note that this expression is the same as the regret (for bandit learning with switching costs) when there is no full feedback at all [11]. This observation is consistent with our intuition that, when β_2 is large, the online algorithm cannot benefit from costly full-feedback.

(ii) When $\beta_2 < \frac{3}{4}K\beta_1$, the lower bound $\underline{R}^{\pi}(T)$ in (3) is equal to

$$\max\left\{ C_1 \beta_a^{\frac{1}{3}} \left(\log_2 K\right)^{\frac{1}{3}} T^{\frac{2}{3}}, C_2 \beta_2^{\frac{1}{3}} T^{\frac{2}{3}} \right\}. \tag{5}$$

As K increases, the first term in (5) quickly dominates. This means that, when the full-feedback cost β_2 is not high, the regret of any online algorithm π will at least increase as $\beta_a^{\frac{1}{3}} (\ln K)^{\frac{1}{3}} T^{\frac{2}{3}}$. If in addition $\beta_2 \leq \frac{3}{2}\beta_1$, we have $\beta_a^{\frac{1}{3}} (\ln K)^{\frac{1}{3}} T^{\frac{2}{3}} = \beta_2^{\frac{1}{3}} (\ln K)^{\frac{1}{3}} T^{\frac{2}{3}}$, which is smaller than $\beta_1^{\frac{1}{3}} K^{\frac{1}{3}} T^{\frac{2}{3}}$. Compared with the earlier case (with large β_2), our regret expression here has the same dependence on T, but now increases more slowly as a function of the total number K of arms. This observation is also consistent with our intuition that, when β_2 is small, the online algorithm can benefit from costly full-feedback more.

Finally, we note that the division of the two cases depends on the value of $K\beta_1$ and β_2. The intuition is that, with K switches, an online algorithm may also attain the feedback from all K arms. Thus, it makes sense to compare $K\beta_1$ with β_2 to determine which type of feedback is more effective.

B. Lower Bound Analysis

To prove Theorem 1, we design two important adversaries, The first adversary captures the dependence of the regret on T. The second adversary uses the first adversary as a building block, which allows us to refine the dependence of the regret on K. For both adversaries, we make use of Yao's principle [20] that the worst-case expected regret $R^{\pi}(T)$ of a randomized online algorithm π is lower-bounded by the expected regret of the best deterministic online algorithm against a randomized adversary. Thus, in the following we focus on designing randomized adversaries, and studying the regret of deterministic online algorithms. Recall that $\mathcal{K} = \{1, ..., K\}$.

Algorithm 1 The Multivariate Hidden Markov (MHM) adversary

```
Parameters: Choose $\epsilon$ and $\sigma$ according to (8).
Initialization: Choose $k^*$ uniformly from $\mathcal{K}$.
for $t = 1 : T$ do
    Step 1: Generate the value of $G(t)$ according to (7).
    Step 2: Generate the losses of each arm $k \in \mathcal{K}$ as follows,
        $l_k(t) = G(t) + \frac{1}{2} - \epsilon \cdot \mathbf{1}_{\{k=k^*\}} + \gamma_k(t),$   (6)
    where $\gamma_k(t) \sim \mathcal{N}(0, \sigma^2)$ are i.i.d. Gaussian random variables with zero-mean and $\sigma^2$-variance.
end for
```

1) Multivariate Hidden Markov (MHM) Adversary: In this section, we provide the first randomized adversary, called Multivariate Hidden Markov (MHM) adversary, which generalizes the idea in [11]. Please see Algorithm 1.

Specifically, Step 1 in Algorithm 1 is the same as that used by the adversary introduced in [11]. That is, for each time t, define the parent time of t as $\rho(t) \triangleq t - 2^{\delta(t)}$, where $\delta(t) \triangleq \max\{\delta \mid t \equiv 0 \pmod{2^\delta}\}$. The main reason that the parent time $\rho(t)$ is $2^{\delta(t)}$ time-slot ahead of time t is to guarantee that with high probability, the generated losses $l_k(t)$ are in $[0, 1]$. Please see our technical report [21] for the concrete proof of this guarantee. Then, Step 1 of MHM generates a Gaussian process $G(t)$ in the following way,

$$G(t) = G(\rho(t)) + \xi(t), \text{ for all time } t \in [1, T], \tag{7}$$

where $G(0) = 0$, and $\xi(t) \sim \mathcal{N}(0, \sigma^2)$ are *i.i.d.* Gaussian random variables with zero-mean and σ^2-variance. As in [11], this process $G(t)$ creates a common uncertainty across all arms. As a result, if an online algorithm does not switch arms, it will have a difficult time figuring out whether the losses experienced on the chosen arms are due to this common process $G(t)$, or due to the chosen arms being inferior to other arms. In Step 2, the first three terms[2] in (6) are also the same as that used in [11].

However, (6) differs from the adversary of [11] in the fourth term. This additional term adds a Gaussian noise $\gamma_k(t)$ to the loss $l_k(t)$ of each arm at each time. This additional noise is critical because our online algorithm π can use costly full-feedback, which is not considered in [11]. Intuitively, without this noise $\gamma_k(t)$, by using one round of costly full-feedback, the online algorithm can know the losses of all arms in the same time-slot. Then, the online algorithm will immediately know which arm is the optimal one (i.e., the arm with a loss that is ϵ lower). In contrast, the additional noise in (6) eliminates the possibility for such a trivial solution. As we explain soon, this additional noise $\gamma_k(t)$ leads to new difficulties in the proof of the lower bound. We refer to this adversary as Multivariate Hidden Markov (MHM) because the hidden loss $l^{\text{hi}}(t) \triangleq l_{\text{k}(t)}(t) - \gamma_{\text{k}(t)}(t)$ satisfies the Markov property [22, p. 235].

Lemma 1. *Consider bandit learning with switching costs and full-feedback costs introduced in Sec. II-A. When $M = 1$, by choosing*

$$\epsilon = \sqrt[3]{\frac{2}{9\ln 2}} \cdot \frac{1}{9\log_2 T} \cdot \beta_b^{\frac{1}{3}} T^{-\frac{1}{3}} \text{ and } \sigma = \frac{1}{9\log_2 T}, \tag{8}$$

the regret of any online learning algorithm π against the MHM adversary is lower-bounded as follows: for $T \geq \max\{\beta_b, 6K\}$,

$$R^\pi(T) \geq \sqrt[3]{\frac{2}{9\ln 2}} \cdot \frac{1}{144\log_2 T} \cdot \beta_b^{\frac{1}{3}} T^{\frac{2}{3}}, \tag{9}$$

where $\beta_b = \min\left\{\frac{3}{4}K\beta_1, \beta_2\right\}$.

Please see our technical report [21] for the complete proof of Lemma 1. From Lemma 1, we can see that the regret lower-bound produced by MHM corresponds to the second term in (3). Note that it correctly captures the dependence of the regret on T.

Below, we briefly sketch the main ideas of our proof and the new difficulties. We follow the approach in [11] to derive the regret lower-bound of any deterministic online algorithm π against the MHM adversary. Specifically, let $\mathcal{P}_{k^*}(\cdot)$ denote the probability measure under the setting where one optimal arm k^* incurs ϵ lower cost than other arms, as in (6). Let $\mathcal{P}_0(\cdot)$ denotes the probability measure when $\epsilon = 0$, i.e., the arm k^* is statistically the same as other arms. In addition, let $l^{\text{ob}}(\cdot)$ denote the observed losses of the online learning algorithm. Then, the analysis in [11] focuses on estimating the Kullback-Leibler (KL) divergence $D_{\text{KL}}(\mathcal{P}_{k^*}(l^{\text{ob}}(1:T)) \| \mathcal{P}_0(l^{\text{ob}}(1:T)))$, which then leads to the lower bound on the regret. However, for our MHM adversary, the additional noise $\gamma_k(t)$ causes several new difficulties in the proof of the lower bound.

Difficulty 1: The observed loss $l^{\text{ob}}(t)$ does not satisfy the Markov property [22, p. 235] any more. Recall that $\rho(t)$ is the parent (time) of t, and thus t is the child (time) of $\rho(t)$. Let $\bar{\rho}(t)$ denote the set of the predecessors of time t, i.e. its parent, parent's parent, etc. Similarly, let $\underline{\rho}(t)$ denote the set of the descendants of time t. Note that without $\gamma_k(t)$, the observed loss $l^{\text{ob}}(t)$ would have been a Gaussian process $G(t)$ plus a fixed constant $\frac{1}{2}$ or $\frac{1}{2} - \epsilon$. Thus, $l^{\text{ob}}(t)$ would have satisfied a form of the Markov property, i.e., conditioned on the current observed losses, the conditional probability distribution of future losses at a descendant time in $\underline{\rho}(t)$ is independent of past losses at any predecessor time in $\bar{\rho}(t)$. Then, the proof could use the chain rule of KL divergence [15, p. 23]. In contrast, with the additional noise $\gamma_k(t)$, the observed loss $l^{\text{ob}}(t)$ does not satisfy the Markov property any more. This is because, conditioned on the observed losses at time t, past observed losses still provide information for the statistics of the future losses. For example, by taking the average of the losses observed at all predecessors in $\bar{\rho}(t)$, we can average out $\gamma_k(t)$ across time, and thus estimate the mean value of the loss at a descendant time in $\underline{\rho}(t)$ with a higher accuracy. Therefore, we cannot use the chain rule directly, and must find a new way to bound the KL divergence.

To overcome this new difficulty, we develop a result on the KL divergence of hidden Markov models [15, p. 69]. Specifically, notice that the hidden loss $l^{\text{hi}}(t) \triangleq l_{\text{k}(t)}(t) - \gamma_{\text{k}(t)}(t)$,

[2] The first three terms in (6) guarantees that the expected values of the losses are $\frac{1}{2}$ and $\frac{1}{2} - \epsilon$ for the sub-optimal arms $k \neq k^*$ and the optimal arm k^*, respectively.

i.e., the loss in (6) but with $\gamma_{\mathrm{k}(t)}(t)$ removed, satisfies the Markov property. Then, using the chain rule of probability, we can show that

$$\begin{aligned} &D_{\mathrm{KL}}\left(\mathcal{P}_{k^*}(l^{\mathrm{ob}}(1:T))\|\mathcal{P}_0(l^{\mathrm{ob}}(1:T))\right) \\ &\leq D_{\mathrm{KL}}\left(\mathcal{P}_{k^*}(l^{\mathrm{ob}}(1:T)|l^{\mathrm{hi}}(1:T))\|\mathcal{P}_0(l^{\mathrm{ob}}(1:T)|l^{\mathrm{hi}}(1:T))\right) \\ &\quad + D_{\mathrm{KL}}\left(\mathcal{P}_{k^*}(l^{\mathrm{hi}}(1:T))\|\mathcal{P}_0(l^{\mathrm{hi}}(1:T))\right). \end{aligned} \tag{10}$$

The first term on the right-hand-side of (10) can be easily calculated at each time, since conditioned on the hidden loss $l^{\mathrm{hi}}(t)$, the observed loss $l^{\mathrm{ob}}(t)$ is only due to *i.i.d.* Gaussian variables $\gamma_k(t)$. The second term on the right-hand-side of (10) can be calculated by using the chain rule of the KL divergence, since the hidden loss $l^{\mathrm{hi}}(t)$ satisfies the Markov property. Bounding the right-hand-side of (10) then leads to a lower bound on the regret as in [11].

Difficulty 2: The losses generated in (6) may go out of the range $[0,1]$. In [11], the authors resolve this problem by clipping any losses $l_k(t)$ to the range $[0,1]$. Since there is no additional noise $\gamma_k(t)$ in their case, after clipping the loss gap between the sub-optimal arm $k \neq k^*$ and the optimal arm k^* is at most ϵ. In contrast, in our case, due to the additional noises $\gamma_k(t)$ that are independent for all k, the loss gap between the sub-optimal and optimal arms could become arbitrarily large both before and after clipping. As a result, it becomes more difficult to establish the regret lower-bound for the clipped losses.

To overcome this new difficulty, we leverage coupling and stochastic dominance [23]. Specifically, since the conditional probability distribution of $\gamma_k(t)$ is symmetric for all arms k, even though the worst-case loss gap could become larger after clipping, the *average* loss gap can still be upper-bounded, which eventually leads to the result in Lemma 1. Please see our technical report [21] for details.

2) Dividing Set (DS) adversary and Randomized Online Learning With Costly Full-Feedback (ROCF): Note that the dependence on K provided in Lemma 1 still needs to be refined. To further refine such dependence, we provide a second adversary, called Dividing Set (DS) adversary. Please see our technical report [21] for details. Finally, we design an online learning algorithm, called Randomized Online Learning With Costly Full-Feedback (ROCF), that attains the following regret for large T,

$$R^{\mathrm{ROCF}}(T) \leq \begin{cases} 4\beta_1^{\frac{1}{3}}\left(K\ln K\right)^{\frac{1}{3}}T^{\frac{2}{3}}, & \text{if } \beta_2 \geq \frac{3K}{4}\beta_1, \\ \frac{7}{2}\beta_2^{\frac{1}{3}}(\ln K)^{\frac{1}{3}}T^{\frac{2}{3}} + O(1), & \text{if } \beta_2 < \frac{3K}{4}\beta_1, \end{cases} \tag{11}$$

which matches the lower bound in Theorem 1. ROCF essentially uses episodic versions of either Exp3 [9] (when β_2 is large) or the shrinking dartboard algorithm [12] (when β_2 is small). Due to page limits, we refer the readers to our technical report [21].

IV. The Power-of-2-Arms (When $M \geq 2$)

In this section, we proceed to the case when $M \geq 2$. In contrast to the previous section where we show that adding costly full-feedback does not change the $\Theta(T^{\frac{2}{3}})$ regret, here we provide a new algorithm that utilizes the flexibility of having 2 (or more) arms and successfully improves the regret to $O(\sqrt{T})$. In this section, we focus on the case when the switching cost is only for changing the set $\hat{\mathbf{k}}(t)$ of the M chosen arms. In other words, there is no switching cost for picking the arm (from these M arms) that is used to incur loss. The results in this section serve as a basis for a more general case in Sec. V, where there are additional switching costs for changing the active arm $\mathrm{k}(t)$ even inside $\hat{\mathbf{k}}(t)$.

Algorithm 2 Randomized Online Learning With Working Groups (ROW)

```
**Parameters:** Choose $\eta_2$, $\tau_2$, $\eta_1$ and $\tau_1$ according to (31).
**Initialization:** $w_k^{\mathrm{ROW}}[1] = 1$ and $p_k^{\mathrm{ROW}}[1] = \frac{1}{K}$, for all $k \in \mathcal{K}$.
**for** $u = 1 : \left\lceil \frac{T}{\tau_1} \right\rceil$ (The $u$-th episode starts from $t_u = (u-1)\tau_1 + 1$ to $t_u + \tau_1 - 1$.) **do**
    *Step 1:* At the beginning of the first time-slot $t_u$, according to probability $p_k^{\mathrm{ROW}}[u]$ calculated in (12), choose a primary arm $k_0^{\mathrm{ROW}}[u]$ from all arms $k \in \mathcal{K}$ for the entire episode.
    **for** $v = 1 : \frac{\tau_1}{\tau_2}$ (The $v$-th sub-episode starts from $t_{u,v} = (u-1)\tau_1 + (v-1)\tau_2 + 1$ to $t_{u,v} + \tau_2 - 1$.) **do**
        *Step 2:* At the beginning of the first time-slot $t_{u,v}$, uniformly choose the set $\hat{\mathbf{k}}_{M-1}^{\mathrm{ROW}}[u,v]$ of $M-1$ secondary arms from the not-yet-been-chosen arms in $\mathcal{K} - \left( \bigcup_{v'=1}^{v-1} \hat{\mathbf{k}}_{M-1}^{\mathrm{ROW}}[u,v'] \bigcup \{k_0^{\mathrm{ROW}}[u]\} \right)$. Then, form the working group by the primary arm and secondary arms, i.e., $\hat{\mathbf{k}}^{\mathrm{ROW}}[u,v] = \left\{k_0^{\mathrm{ROW}}[u]\right\} \bigcup \hat{\mathbf{k}}_{M-1}^{\mathrm{ROW}}[u,v]$.
        *Step 3:* Initialize the weights $\hat{w}_k^{\mathrm{ROW}}(t_{u,v})$ and probabilities $\hat{p}_k^{\mathrm{ROW}}(t_{u,v})$ of all arms $k \in \hat{\mathbf{k}}^{\mathrm{ROW}}[u,v]$ according to (13) and (14), respectively.
        **for** $t = t_{u,v} : t_{u,v} + \tau_2 - 1$ **do**
            *Step 4:* Use an arm $k \in \hat{\mathbf{k}}^{\mathrm{ROW}}[u,v]$ as the active arm according to the updated probability $\hat{p}_k^{\mathrm{ROW}}(t)$.
            *Step 5:* Update the weights $\hat{w}_k^{\mathrm{ROW}}(t)$ and probabilities $\hat{p}_k^{\mathrm{ROW}}(t)$ of all arms $k \in \hat{\mathbf{k}}^{\mathrm{ROW}}[u,v]$ according to (15) and (14), respectively.
        **end for**
    **end for**
    *Step 6:* At the end of the last time-slot of the $u$-th episode, update the weights $w_k^{\mathrm{ROW}}[u+1]$ and probabilities $p_k^{\mathrm{ROW}}[u+1]$ of all arms $k \in \mathcal{K}$ according to (17) and (12), respectively.
**end for**
```

A. Randomized Online Learning With Working Groups (ROW)

We call our new algorithm Randomized Online Learning With Working Groups (ROW). Please see Algorithm 2. We start with describing the high-level skeleton of ROW. Recall that $\mathcal{K} = \{1, ..., K\}$.

Idea 1: Note that in order to obtain the $O(\sqrt{T})$ regret, we can switch or use costly full-feedback at most $O(\sqrt{T})$ number of times. The first idea of ROW is thus to design an effective way to rotate a working group (of M arms) through all K arms, so that plenty of feedback can be obtained for all the arms, while incurring $O(\sqrt{T})$ switching costs and zero full-feedback costs. Specifically, ROW divides the entire time-horizon into $U = \left\lceil \frac{T}{\tau_1} \right\rceil$ episodes, each with $\tau_1 = \Theta(\sqrt{T})$ time-slots. In the first time-slot $t_u = (u-1)\tau_1 + 1$ of the u-th ($u = 1, ..., U$) episode, ROW chooses a primary arm $k_0^{\text{ROW}}[u]$. This primary arm $k_0^{\text{ROW}}[u]$ will be fixed for all τ_1 time-slots in the u-th episode. In addition, ROW divides each episode into $V = \left\lceil \frac{K-1}{M-1} \right\rceil$ sub-episodes, each with $\tau_2 = \frac{\tau_1}{V}$ time-slots. In the rest of this paper, we refer to the v-th sub-episode in the u-th episode as sub-episode (u, v). At the beginning of the first time-slot $t_{u,v} = (u-1)\tau_1 + (v-1)\tau_2 + 1$ of sub-episode (u, v), ROW uniformly chooses $M - 1$ secondary arms from the arms that have not yet been chosen in the u-th episode[3] (i.e., Step 2 in Algorithm 2). We let $\hat{\mathbf{k}}_{M-1}^{\text{ROW}}[u, v]$ denote the set of the $M-1$ secondary arms chosen in sub-episode (u, v). Let $\hat{\mathbf{k}}^{\text{ROW}}[u, v] = \left\{k_0^{\text{ROW}}[u]\right\} \bigcup \hat{\mathbf{k}}_{M-1}^{\text{ROW}}[u, v]$ denote the working group formed by the primary arm and secondary arms. The working group $\hat{\mathbf{k}}^{\text{ROW}}[u, v]$ will be fixed for the whole sub-episode (u, v).

Notice that by using this idea, ROW only switches at the boundaries of sub-episodes and never uses full feedback. Therefore, by tuning τ_2 to be $\Theta(\sqrt{T})$, the total switching cost is guaranteed to be $\Theta(\sqrt{T})$, and the total full-feedback cost is 0. More importantly, with this idea, we not only have the feedback for the primary arm for the entire episode, but also have the feedback for each secondary arm for $\frac{1}{V}$ fraction of each episode. Intuitively, this way of obtaining feedback incurs much lower costs than using costly full-feedback. For example, if we want to obtain the same amount of feedback by using costly full-feedback alone, we would have to incur a full-feedback cost equal to $\Theta(\sqrt{T})$ in every episode! This is also the reason that ROW does not use full feedback at all.

We now describe the rest of the details of ROW. At the beginning of the first time-slot of the u-th ($u = 1, ..., U$) episode, each arm $k \in \mathcal{K}$ is associated with a weight $w_k^{\text{ROW}}[u]$, which is initialized to be $w_k^{\text{ROW}}[1] = 1$ (we will describe how to update $w_k^{\text{ROW}}[u]$ from $w_k^{\text{ROW}}[u-1]$ shortly). Then, from all arms $k \in \mathcal{K}$, ROW chooses a primary arm $k_0^{\text{ROW}}[u]$ with probability (i.e., Step 1 in Algorithm 2)

$$p_k^{\text{ROW}}[u] = \frac{w_k^{\text{ROW}}[u]}{\sum_{k=1}^{K} w_k^{\text{ROW}}[u]}. \tag{12}$$

Then, at the beginning of the first time-slot of each sub-episode (u, v), the $M - 1$ secondary arms $\hat{\mathbf{k}}_{M-1}^{\text{ROW}}[u, v]$ are chosen uniformly and rotated through all of the rest $K - 1$ arms as we described earlier (i.e., Step 2 in Algorithm 2).

Further, within each sub-episode (u, v) we solve a bandit-learning problem with the set of arms restricted to the chosen working group. Note that this restricted version of the bandit-learning problem has no switching cost (since any arm $k \in \hat{\mathbf{k}}^{\text{ROW}}[u, v]$ can be used as the active arm without incurring switching costs), and also has full feedback (from all the arms $k \in \hat{\mathbf{k}}^{\text{ROW}}[u, v]$). Thus, we can directly use the full-feedback version of the Exp3 algorithm inside each sub-episode (u, v). Specifically, in the first time-slot $t_{u,v}$ of sub-episode (u, v), ROW initializes the weights of all the arms $k \in \mathcal{K}$ as follows (i.e., Step 3 in Algorithm 2),

$$\hat{w}_k^{\text{ROW}}(t_{u,v}) = w_k^{\text{ROW}}[u], \tag{13}$$

i.e., to be the values of the weights at the beginning of the entire episode u. Then, for each time $t = t_{u,v}, ..., t_{u,v} + \tau_2 - 1$, each arm $k \in \hat{\mathbf{k}}^{\text{ROW}}[u, v]$ is used as the active arm $\mathrm{k}^{\text{ROW}}(t)$ with probability (i.e., Step 4 and Step 5 in Algorithm 2)

$$\hat{p}_k^{\text{ROW}}(t) = \frac{\hat{w}_k^{\text{ROW}}(t)}{\sum_{k \in \hat{\mathbf{k}}^{\text{ROW}}[u,v]} \hat{w}_k^{\text{ROW}}(t)}. \tag{14}$$

After the losses $l_k(t)$ of all the arms $k \in \hat{\mathbf{k}}^{\text{ROW}}[u, v]$ are obtained for time t, ROW updates their weights with a tunable parameter η_2 as follows (i.e., Step 5 in Algorithm 2),

$$\hat{w}_k^{\text{ROW}}(t+1) = \hat{w}_k^{\text{ROW}}(t) \cdot e^{-\eta_2 l_k(t)}, \tag{15}$$

and then proceeds to the next time-slot $t + 1$. Note that the weights $\hat{w}_k^{\text{ROW}}(t)$ are reset by (13) in the first time-slot $t = t_{u,v}$ of each sub-episode (u, v).

Finally, at the end of the last time-slot of the entire episode u, ROW collects all the feedback received during the episode. Next, during the sub-episodes that arm k was chosen for the working group, ROW subtracts the loss of the primary arm from the corresponding loss of this arm k. Then, the resulting value is divided by the conditional probability that k is chosen as a secondary arm (conditioned on k not being the primary arm), i.e., $\frac{M-1}{K-1}$. Precisely, we let $v_u(k) \triangleq \left\{v \mid v = 1, ..., V, k \in \hat{\mathbf{k}}^{\text{ROW}}[u, v]\right\}$ denote the sub-episodes (u, v) when the arm k was chosen in the working group. Let $L_k[u, v_u(k)] \triangleq \sum\limits_{v \in v_u(k)} \sum_{t=t_{u,v}}^{t_{u,v}+\tau_2-2} l_k(t)$ denote the sum of the losses of arm k in sub-episodes (u, v) (except the last time-slot $t = t_{u,v} + \tau_2 - 1$) for all $v \in v_u(k)$. Then, ROW computes the loss difference of each arm $k \in \mathcal{K}$ as follows,

$$\tilde{L}_k^{\text{ROW}}[u] = \frac{L_k[u, v_u(k)] - L_{k_0^{\text{ROW}}[u]}[u, v_u(k)]}{\frac{M-1}{K-1}}. \tag{16}$$

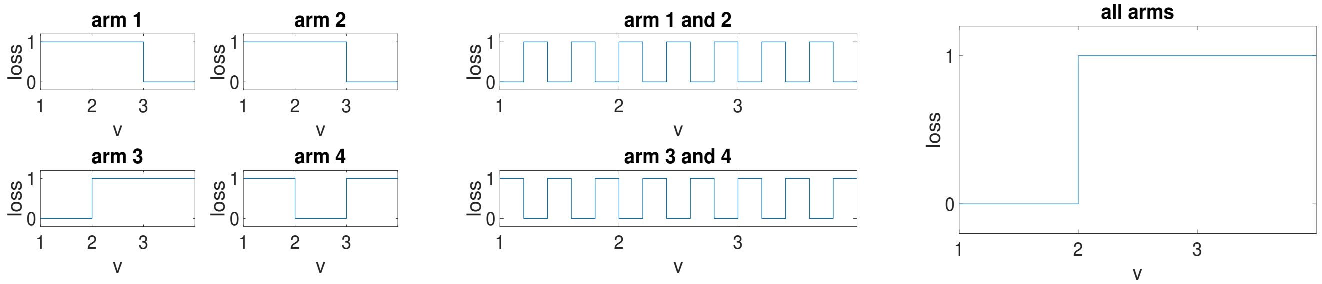
Note that for the primary arm $k_0^{\text{ROW}}[u]$, the loss difference is $\tilde{L}_{k_0^{\text{ROW}}[u]}^{\text{ROW}}[u] = 0$, which is also consistent with (16). Then, ROW updates the weights for all the arms $k \in \mathcal{K}$ with a tunable parameter η_1 as follows (i.e., Step 8 in Algorithm 2),

$$w_k^{\text{ROW}}[u+1] = w_k^{\text{ROW}}[u] \cdot e^{-\eta_1 \tilde{L}_k[u]}, \tag{17}$$

which becomes the initial weights for the next episode $u + 1$. In (31), we give the values of all parameters of ROW, i.e., η_1, η_2, τ_1 and τ_2.

Readers familiar with bandit-learning algorithms may have already noticed two other crucial differences in ROW. First, a different weight-decay parameter η_2 is used to update weights in (15) within the episode, compared with the parameter η_1 that is used in (17) across episodes. Second, when updating the weights across episodes in (17), we use the difference between

[3] When $K - 1$ is not divisible by $M - 1$, the number of the remaining unchosen arms in the last (i.e., V-th) sub-episode may be less than $M - 1$. In this case, after choosing all those unchosen arms, ROW uniformly chooses the secondary arms from the arms that have not yet been chosen for the V-th sub-episode.



(a) Trace in counter-example 1 (*i.i.d.* across arms k and sub-episodes $[u, v]$).

(b) Trace in counter-example 2 (repeats every 2 time-slots).

(c) Trace in counter-example 3 (repeats every episode u).

Fig. 1: One realization of the counter-example traces in one episode.

the loss of an arm and that of the primary arm, instead of using the absolute loss of the arm directly. In the following, we explain why these two differences (i.e., our idea 2 and idea 3) are crucial for achieving the $O(\sqrt{T})$ regret. We emphasize that these design choices are very different from the standard full-feedback algorithm. As we will demonstrate through three counter-examples below, our power-of-2-arms setting is very different from the full-feedback setting, and thus requires these crucial changes for attaining a low regret.

Idea 2: Use different weight-decay parameters η_2 and η_1. Recall that in every episode, ROW can obtain at least $\frac{1}{V}$ fraction of feedback from every arm. We would have hoped that this amount of feedback is sufficient for attaining a low $O(\sqrt{T})$ regret. Indeed, consider an alternate bandit-learning problem where the feedback of each arm is obtained independently with probability $\frac{1}{V}$ in every time-slot. It is not difficult to show that Exp3 [9] using this amount of feedback will attain the $O(\sqrt{T})$ regret.

However, compared with the above alternate setting when feedback is collected with probability $\frac{1}{V}$, the difficulty that we are facing here is that in ROW the feedback of different arms are not collected simultaneously. Indeed, the secondary arms are fixed during the whole sub-episode. Thus, we either have all feedback of an arm, or have none for the whole sub-episode. However, online decisions still need to be made after the feedback of only a subset of the arms are collected. This, combined with the possible correlation of the loss feedback across arms and time, leads to a large regret for existing full-feedback algorithm as we demonstrate below. Specifically, we construct two counter-examples to illustrate the difficulties in dealing with such correlation. For ease of exposition, we use $l(t_1 : t_2) \triangleq [l(t), \text{ for all } t = t_1, t_1 + 1, ..., t_2]$ to collect $l(t)$ from $t = t_1$ to $t = t_2$.

Counter-example 1: Consider $K = 4$ arms and $M = 2$. For each arm k, in each sub-episode (u, v), $l_k(t_{u,v} : t_{u,v} + \tau_2 - 1) = 0$ with probability $\frac{1}{2}$, and $l_k(t_{u,v} : t_{u,v} + \tau_2 - 1) = 1$ with probability $\frac{1}{2}$. The losses are independent across arms k and across sub-episodes $[u, v]$. Please see Fig. 1a for this loss trace in one episode. Using this counter-example, we show why existing bandit-learning method, Exp3 [10], could lead to a poor regret. Let us consider the optimal static loss. First, the expected total loss of each arm is trivially $\mathbb{E}[L] = \frac{T}{2}$. Second, let us estimate the variance of the total loss of each arm. Since the loss is a constant within a sub-episode, the higher correlation in time leads to a higher variance in the total loss of each arm. Specifically, for each arm, the variance of its total loss in a sub-episode[4] is $\Theta(\tau_2^2)$. Thus the variance of its total loss across T time-slots is $\text{Var}(L) = \frac{T}{\tau_2} \cdot \Theta(\tau_2^2) = \Theta(T^{\frac{3}{2}})$. Thus, one of the K arms may incur a total loss that is smaller than the average by $\Theta(\sqrt{\text{Var}(L)})$. As a result, the total loss of the optimal static decision OPT is $\mathbb{E}[L] - \Theta(\sqrt{\text{Var}(L)}) = \frac{T}{2} - \Theta(T^{\frac{3}{4}})$. (This estimate can also be obtained by applying the random walk analysis [18, p. 111].) Next, we consider the total loss of the episodic version of Exp3 [10]. Such version of Exp3 picks an arm k_0 at the beginning of an episode, and use it as the active arm for the entire episode. Since the loss in each episode is independent, the total loss of such Exp3 will be the average loss of each arm in this counter-example, i.e., $\frac{T}{2}$. Therefore, the regret would be $\Theta(T^{\frac{3}{4}})$.

Counter-example 1 clearly illustrates why the higher correlation in time leads to a higher regret for the episodic version of Exp3. To overcome this difficulty, we make an important observation. In this setting with highly correlated losses, we observe that one arm (with losses 0) will be consistently better than the other arms (with losses 1) in each sub-episode. We may then beat the average loss by switching to the better arm within a sub-episode. Indeed, with $M = 2$, the chance that one of the two arms incurs zero loss is $\frac{3}{4}$. Thus, if we can switch to the better arm (with losses 0) quickly within a sub-episode, we may attain a total loss approximately equals to $\frac{T}{4}$, which would have beaten the optimal static decision OPT. This counter-example thus suggests why it is important to use Exp3 [9] inside each sub-episode (in addition to across episodes).

However, it is still highly non-trivial to choose the parameter η of Exp3 within each sub-episode. One possible thought is that, we can think of each sub-episode as a bandit-learning problem with $\tau_2 = \Theta(\sqrt{T})$ time-slots. Then, if we view the better arm within the sub-episode as the static optimal arm, we would have to use $\eta = \Theta(T^{-\frac{1}{4}})$ in order to attain the minimal regret against the better arm. However, this choice of η would have been too large, as can be seen in the counter-example below.

[4] In contrast, if the losses were *i.i.d.* in time, the variance should have been $\Theta(\tau_2)$.

Counter-example 2*:* Consider $K = 4$ arms and $M = 2$. For arms $k = 1, 2$, $l_k(t) = 0$ for all odd time-slots t, and $l_k(t) = 1$ for all even time-slots t. For arms $k = 3, 4$, $l_k(t) = 1$ for all odd time-slots t, and $l_k(t) = 0$ for all even time-slots t. Please see Fig. 1b for this loss trace in one episode. Using this counter-example, we can see why using Exp3 [9] with a parameter $\eta = \Theta(T^{-\frac{1}{4}})$ could lead to a poor regret. Let us consider the optimal static loss. Since the total loss of every arm is $\frac{T}{2}$, the optimal static loss is $\frac{T}{2}$. Next, we consider the total loss of Exp3. Notice that the probability of each arm is initialized to be the same, i.e., $\frac{1}{K}$, at time $t = 1$. Then, at each time, suppose that all arms have been observed almost the same number of times. Thus, the probabilities of all arms would be about the same. However, whenever an arm with loss $l_{k_1}(t) = 0$ and an arm with loss $l_{k_2}(t) = 1$ are observed simultaneously, at the next time $t+1$ Exp3 will use the arm k_1 as the active arm with a probability higher by approximately $\Theta(\eta)$. According to counter-example 2, $l_{k_1}(t+1) = 1$. Thus, Exp3 will suffer an additional loss $\Theta(\eta)$ approximately at each time. Hence, the total loss of Exp3 will be $\frac{T}{2} + \Theta(\eta T) = \frac{T}{2} + \Theta(T^{\frac{3}{4}})$. Therefore, the regret would be $\Theta(T^{\frac{3}{4}})$.

Counter-example 2 clearly indicates that, in order to attain the $O(\sqrt{T})$ regret, the parameter η_2 should be no larger than $O(T^{-\frac{1}{2}})$. However, since a sub-episode is of length much smaller than T, we conjecture that η_2 still needs to be larger than η_1 (the latter is used across episodes), so that ROW converges fast to the better arm inside the chosen working group. Lemma 4 in Sec. IV-B2 will provide the exact condition on how η_2 and η_1 should be tuned to obtain the $O(\sqrt{T})$ regret.

Idea 3**:** Use the loss difference from the primary arm to update weights across episodes. We next describe why it is also crucial to use the loss difference in (16) instead of the absolute loss of each arm. Recall that at the end of each episode, we receive τ_1 feedback from the primary arm, but only $\tau_2 = \frac{\tau_1}{V}$ feedback from each secondary arm. Intuitively, this bias will also increase the variance of the total losses accumulated in the past, which again leads to a higher regret. The following counter-example illustrates this difficulty.

Counter-example 3*:* Consider $K = 4$ arms and $M = 2$. In the first sub-episode of each episode, the loss of each arm at each time is 0. For all subsequent sub-episodes of each episode, the loss of each arm at each time is 1. Please see Fig. 1c for this loss trace in one episode. In the literature, the standard way to deal with this bias in the amount of feedback is to divide the observed loss by the probability that the arm is observed [9], [10], [17]. For each arm, this probability is $p_k[u] + (1 - p_k[u])\frac{M-1}{K-1}$, where $p_k[u]$ is the probability that arm k is chosen as the primary arm, and $(1 - p_k[u])\frac{M-1}{K-1}$ is the probability that arm k is chosen as the secondary arm in a sub-episode. With this mechanism, the estimated losses will be $\tilde{L}_k[u] = \frac{2\tau_2}{p_k[u]+(1-p_k[u])\frac{M-1}{K-1}}$ when k is the primary arm, $\tilde{L}_k[u] = 0$ when k is a secondary arm that is chosen in the first ($v = 1$) sub-episode, and $\tilde{L}_k[u] = \frac{\tau_2}{p_k[u]+(1-p_k[u])\frac{M-1}{K-1}}$ when k is a secondary arms that is chosen in the subsequent ($v = 2, 3$) sub-episodes. Suppose that $p_k[u] = \frac{1}{K}$ is the same across all arms. Then, the denominator is actually the same across all arms, but the numerator will still lead to a significant variance. Indeed. since the primary arm is chosen randomly with probability $p_k[u] = \frac{1}{K}$, it is not hard to verify that the total estimated loss of each arm over an episode will have a variance of $\Theta(\tau_2^2)$. In contrast, if full feedback was available, all arms would have a total loss equal to $2\tau_2$ in an episode, and the variance would have been zero. It is easy to show that, with this additional $\Theta(\tau_2^2)$ gap in the variance, the regret of Exp3 [10] is still $O(T^{\frac{2}{3}})$, which is much larger than $O(\sqrt{T})$.

Counter-example 3 thus suggests that, instead of dividing the loss by the probability of observing an arm, we need some new ways to deal with the above bias issue. Precisely, in (16), ROW updates the estimated loss by the difference of the loss of each secondary arm and that of the primary arm. In addition, the loss difference of the primary arm is simply 0. Returning to counter-example 3, the new estimated loss will then be $\tilde{L}_k[u] = 0$ for all the arms $k \in \mathcal{K}$. Thus, the additional variance $\Theta(\tau_2^2)$ of the estimated losses has been eliminated, which is also crucial for attaining the $O(\sqrt{T})$ regret.

B. Regret Analysis

In Theorem 2 below, we show the upper bound of the regret attained by ROW. For ease of exposition, we focus on the case when $K-1$ is divisible by $M-1$. (It is not difficult to extend to the case when $K-1$ is not divisible by $M-1$. Please see our technical report [21] for details.)

Theorem 2. *Consider bandit learning with switching costs and full-feedback costs introduced in Sec. II-A. When $M \geq 2$, the regret of ROW can be upper-bounded as follows, for $T \geq \frac{448(K-1)^2 \ln K}{\frac{5}{2}+2\beta_1}$,*

$$R^{ROW}(T) \leq 8b_1 \frac{K-1}{M-1}\sqrt{\ln K}\sqrt{T} + b_2, \tag{18}$$

where $b_1 = \sqrt{\frac{5}{2} + 2b_3\beta_1}$, $b_2 = b_3\beta_1 + 1$ and $b_3 = \min\{M, K-M\}$.

In Sec. III when $M = 1$, the optimal regret is $\Theta(T^{\frac{2}{3}})$ for bandit learning with switching costs and full-feedback costs. In sharp contrast, now with $M \geq 2$, ROW achieves a significantly lower regret equals to $O(\sqrt{T})$. Moreover, ROW never uses full feedback. Further, as M increases, the regret of ROW can be further reduced. *To the best of our knowledge, this is the first result in the literature to utilize the flexibility of choosing $M \geq 2$ arms to improved the regret to $O(\sqrt{T})$ for bandit learning with switching costs.* Furthermore, using a trivial lower bound for bandit learning with free full-feedback [12], [17], we can conclude that the $O(\sqrt{T})$ regret cannot be further improved.

The rest of this section is devoted to the proof of Theorem 2. Due to the three new ideas in ROW, new analytical techniques are needed to capture the evolution of the weights, which are also of independent interest. In order to relate the loss of ROW to that of the optimal static loss, our analysis below is carried out in three steps. First, inside each sub-episode, we relate the total loss of ROW in each sub-episode to a log-sum-exp function $g_2[u, v]$ of the parameter η_2 and the feedback from the chosen working group. Second, at the end of each episode, we relate $g_2[u, v]$ of all sub-episodes to another log-sum-exp function $g_1[u]$ of the parameter η_1 and the loss difference $\tilde{L}_k^{\text{ROW}}[u]$. Third, across all episodes, we relate $g_1[u]$ to the optimal static loss. Combining these three steps, the total loss of ROW will then be related to the optimal static loss. In the following, we let $\mathcal{H}[u-1]$ denote the σ-algebra generated by the observation of ROW from time $t=1$ to $t=(u-1)\tau_1$. Let $L_k[u,v] \triangleq \sum_{t=t_{u,v}}^{t_{u,v}+\tau_2-2} l_k(t)$.

1) Inside each sub-episode: We start by relating the expected loss of ROW inside each sub-episode (u,v) to a log-sum-exp function $g_2[u,v]$ (see Lemma 2). This function $g_2[u,v]$ will then be further related to the variance of the feedback from the chosen working group $\hat{\mathbf{k}}^{\text{ROW}}[u,v]$ in the sub-episode (see Lemma 3). Recall that in (13), the weights $\hat{w}_k^{\text{ROW}}(t_{u,v})$ in the first time-slots of all sub-episodes are initialized to be the weights $w_k^{\text{ROW}}[u]$ at the beginning of the episode u. Thus, given a same working group, the probabilities $\hat{p}_k^{\text{ROW}}(t_{u,v})$ are also the same at the beginning of all sub-episode v in an episode u. We let

$$\hat{p}_k^{\text{ROW}}[u] \triangleq \frac{w_k^{\text{ROW}}[u]}{\sum_{k\in\hat{\mathbf{k}}^{\text{ROW}}[u,v]} w_k^{\text{ROW}}[u]} \tag{19}$$

denote this common probability.

Lemma 2. *For each sub-episode (u,v), given the history $\mathcal{H}[u-1]$ and the chosen working group $\hat{\mathbf{k}}[u,v]$, we have*

$$\sum_{t=t_{u,v}}^{t_{u,v}+\tau_2-1} \sum_{k\in\hat{\mathbf{k}}^{ROW}[u,v]} \hat{p}_k^{ROW}(t) l_k(t) \leq g_2[u,v] + \frac{1}{2}\eta_2\tau_2 + 1, \tag{20}$$

where

$$g_2[u,v] \triangleq -\frac{1}{\eta_2}\ln\left(\sum_{k\in\hat{\mathbf{k}}^{ROW}[u,v]} \hat{p}_k^{ROW}[u] e^{-\eta_2 L_k[u,v]}\right). \tag{21}$$

On the left-hand-side of (20), the probability $\hat{p}_k^{\text{ROW}}(t)$ is the probability of using arm k as the active arm. Thus, the left-hand-side of (20) represents the conditional (conditioned on the working group $\hat{\mathbf{k}}^{\text{ROW}}[u,v]$ and history $\mathcal{H}[u-1]$) expected loss of ROW in sub-episode (u,v). Hence, (20) upper-bounds the conditional expected loss of ROW by a log-sum-exp function $g_2[u,v]$ and the term $\frac{1}{2}\eta_2\tau_2+1$. We make two important comments. First, the value of $g_2[u,v]$ is approximated dominated by the arm with the smallest loss $L_k[u,v]$ (whenever the corresponding probability $\hat{p}_k^{\text{ROW}}[u]$ is non-zero). (20) thus confirms that ROW is trying to switch to the "better" arm in the working group. Second, the gap $\frac{1}{2}\eta_2\tau_2$ is much smaller than the gap $\frac{1}{2}\eta\tau_2^2$ incurred by the episodic version of Exp3 [10]. Note that the above-mentioned two conclusions precisely capture our ideas 1 and 2, which together allow ROW to converge quickly to the better arm in the working group. Please see our technical report [21] for the complete proof of Lemma 2.

The following lemma then relates $g_2[u,v]$ to the expectation and variance of the feedback from the chosen working group in the sub-episode, which will be useful when we move to the second-step of studying the weight updates at the end of each episode.

Lemma 3. *For each sub-episode (u,v), given the history $\mathcal{H}[u-1]$ and the chosen working group $\hat{\mathbf{k}}^{ROW}[u,v]$, if $\eta_2\tau_2 \leq \ln 2$, we have*

$$\begin{aligned} g_2[u,v] \leq\ & \mathbb{E}\left[L[u,v]\middle|\mathcal{H}[u-1], \hat{\mathbf{k}}^{ROW}[u,v]\right] \\ & - \frac{\eta_2}{8}\cdot Var\left(L[u,v]\middle|\mathcal{H}[u-1], \hat{\mathbf{k}}^{ROW}[u,v]\right), \end{aligned} \tag{22}$$

where the expectation is taken with regard to the randomness in $\hat{p}_k^{ROW}[u]$, i.e.,

$$\mathbb{E}\left[L[u,v]\middle|\mathcal{H}[u-1], \hat{\mathbf{k}}^{ROW}[u,v]\right] \triangleq \sum_{k\in\hat{\mathbf{k}}^{ROW}[u,v]} \hat{p}_k^{ROW}[u] L_k[u,v],$$

$$\begin{aligned} Var\left(L[u,v]\middle|\mathcal{H}[u-1], \hat{\mathbf{k}}^{ROW}[u,v]\right) \triangleq & \sum_{k\in\hat{\mathbf{k}}^{ROW}[u,v]} \hat{p}_k^{ROW}[u] \\ & \cdot\left(L_k[u,v] - \mathbb{E}\left[L[u,v]\middle|\mathcal{H}[u-1], \hat{\mathbf{k}}^{ROW}[u,v]\right]\right)^2. \end{aligned}$$

Notice that the expectation and variance on the right-hand-side of (22) are for the feedback from the working group $\hat{\mathbf{k}}^{\text{ROW}}[u,v]$. Thus, Lemma 3 shows that the log-sum-exp function $g_2[u,v]$ can be related to the expectation and variance of the feedback from the chosen working group. Given the working group $\hat{\mathbf{k}}^{\text{ROW}}[u,v]$, Lemma 3 is proved by applying the Taylor expansion on the e^{-x} function in $g_2[u,v]$. Please see our technical report [21] for the complete proof of Lemma 3.

2) Relating the loss upper-bound at the end of a sub-episode to the weights across episodes: Lemma 2 provides an upper bound on the loss of ROW at the end of each sub-episode (u,v). Note that this upper bound depends on η_2. On the other hand, at the end of each episode u, we calculate the weights according to (17). Notice that not only is $\tilde{L}_k^{\text{ROW}}[u]$ in (17) different from $L_k[u,v]$ in (21), the parameter η_2 is also different from η_1. Thus, we need a way to convert the loss upper-bound in Lemma 2 for each sub-episode to a form that depends on the weights calculated by (17). This is accomplished by Lemma 4 below. Further, this lemma gives a sufficient condition on how to tune the parameters η_2 and η_1.

Specifically, notice that the loss difference $\tilde{L}_k^{\text{ROW}}[u]$ calculated in (16) is a difference from the loss of the primary arm $k_0^{\text{ROW}}[u]$. We let $g_2[u]$ denote the sum of $g_2[u,v]$ for all sub-episodes v, minus a term that corresponds to the loss of the primary arm, i.e.,

$$\begin{aligned} g_2[u] &\triangleq \sum_{v=1}^{V} g_2[u,v] - \sum_{v=1}^{V} L_{k_0^{\text{ROW}}[u]}[u,v] \\ &= -\frac{1}{\eta_2}\sum_{v=1}^{V}\ln\left(\sum_{k\in\hat{\mathbf{k}}^{\text{ROW}}[u,v]} \hat{p}_k^{\text{ROW}}[u] e^{-\eta_2\mathcal{L}_k^{\text{ROW}}[u,v]}\right), \end{aligned} \tag{23}$$

where $\mathcal{L}_k^{\text{ROW}}[u,v] = L_k[u,v] - L_{k_0^{\text{ROW}}[u]}[u,v]$.

Lemma 4. *If the parameters η_2, τ_2, η_1 and τ_1 satisfy that*

$$\eta_2 \geq 16\left(\frac{K-1}{M-1}\right)^2\cdot\eta_1,\ \eta_2\tau_2 \leq \ln 2 \text{ and } \eta_1\tau_1 \leq \ln 2, \tag{24}$$

we have

$$\begin{aligned} &\mathbb{E}_{\hat{\mathbf{k}}^{ROW}[u,1:V]}\left[g_2[u]\middle|\mathcal{H}[u-1]\right] \\ &\qquad \leq \mathbb{E}_{\hat{\mathbf{k}}^{ROW}[u,1:V]}\left[g_1[u]\middle|\mathcal{H}[u-1]\right], \end{aligned} \tag{25}$$

where the expectation is taken with respect to the randomness in the working groups, and

$$g_1[u] \triangleq -\frac{1}{\eta_1} \ln \left(\sum_{k=1}^{K} p_k^{ROW}[u] e^{-\eta_1 \tilde{L}_k^{ROW}[u]} \right). \tag{26}$$

The log-sum-exp function $g_2[u]$ on the left-hand-side of (25) is related to $g_2[u,v]$ through (23), which is then related to the loss of ROW in each sub-episode through (20). The log-sum-exp function $g_1[u]$ on the right-hand-side of (25) is related to the weights calculated at the end of the episode. Thus, Lemma 4 relates the loss upper-bound at the end of each sub-episode to the weights across episodes, and (24) confirms our conjecture that η_2 should be larger than η_1.

The proof of Lemma 4 first relates the function $g_1[u]$ and $g_2[u]$ to the variances of the working-group feedback and the loss differences, respectively, and then bounds these variances. Please see Appendix A for the proof sketch of Lemma 4.

Up to now, by combining (20), (23) and (25) for all sub-episode v and episode u, we can relate the total loss of ROW to $g_1[u]$ as follows,

$$\begin{aligned} &\sum_{u=1}^{U} \mathbb{E}\Bigg\{ \mathbb{E}\Bigg[\sum_{v=1}^{V} \sum_{t=t_{u,v}}^{t_{u,v}+\tau_2-1} \sum_{k \in \hat{\mathbf{k}}^{\text{ROW}}[u,v]} \hat{p}_k^{\text{ROW}}(t) \\ &\qquad \cdot l_k(t) - \sum_{v=1}^{V} L_{k_0^{\text{ROW}}[u]}[u,v] \Big| \mathcal{H}[u-1] \Bigg] \Bigg\} \\ &\leq \sum_{u=1}^{U} \mathbb{E}\left\{ \mathbb{E}\left[g_1[u] \middle| \mathcal{H}[u-1] \right] \right\} + \frac{1}{2}\eta_2 T + VU, \end{aligned} \tag{27}$$

where the outer and inner expectations on both sides are taken with respect to $\mathcal{H}[u-1]$ and $\hat{\mathbf{k}}^{\text{ROW}}[u, 1:V]$, respectively. In the next subsubsection, we show how to relate the first term on the right-hand-side of (27) to the optimal static loss.

3) Relating the upper-bound of the total loss of ROW to the optimal static loss: Lemma 5 below relates the sum of $g_1[u]$ on the right-hand-side of (27) to the optimal static loss of OPT.

Lemma 5. *We have the following inequality,*

$$\begin{aligned} &\sum_{u=1}^{U} \mathbb{E}_{\mathcal{H}[u-1]} \left\{ \mathbb{E}_{\hat{\mathbf{k}}^{ROW}[u,1:V]} \left[g_1[u] \middle| \mathcal{H}[u-1] \right] \right\} \\ &\leq \mathit{Cost}^{OPT}(1:T) + \frac{\ln K}{\eta_1} \\ &\quad - \sum_{u=1}^{U} \mathbb{E}_{\mathcal{H}[u-1]} \left\{ \mathbb{E}\left[\sum_{v=1}^{V} L_{k_0^{ROW}[u]}[u,v] \middle| \mathcal{H}[u-1] \right] \right\}. \end{aligned} \tag{28}$$

In (28), the term on the left-hand-side is one of the terms in the upper bound of the total loss of ROW, i.e., the first term on the right-hand-side of (27). The first term on the right-hand-side is the optimal static loss. The second term on the right-hand-side of (28) can be obtained by following the Exp3 analysis [9]. The third term on the right-hand-side of (28) is because the loss of the primary arm is subtracted in $g_2[u]$ (see (23)). This term also appears on the left-hand-side of (27), which will eventually be cancelled. Please see our technical report [21] for the complete proof of Lemma 5.

4) The final regret: Since ROW only switches the chosen working group $\hat{\mathbf{k}}(t)$ at the boundaries of the sub-episodes, and the expected number of switching the active arm is at most $\ln 2$ in each sub-episode, the total switching cost of ROW can be upper-bounded as follows,

$$\sum_{t=1}^{T} \sum_{k \in \hat{\mathbf{k}}^{\text{ROW}}(t)} \beta_1 \mathbf{1}_{\{k \notin \hat{\mathbf{k}}^{\text{ROW}}(t-1)\}} \leq \min\{M, K-M\} \cdot \beta_1 \left\lceil \frac{T}{\tau_2} \right\rceil. \tag{29}$$

Next, since ROW never asks for full feedback, the total full-feedback cost of ROW is 0. Combining (27), (28) and (29), we can see that the regret of ROW is upper-bounded as follows,

$$\begin{aligned} &R^{\text{ROW}}(T) \\ &\leq \frac{\ln K}{\eta_1} + \frac{1}{2}\eta_2 T + \min\{M, K-M\} \cdot \beta_1 \left\lceil \frac{T}{\tau_2} \right\rceil + \left\lceil \frac{T}{\tau_2} \right\rceil. \end{aligned} \tag{30}$$

Then, by choosing

$$\begin{cases} \eta_2 = \frac{c_1 c_2}{\sqrt{T}}, & \tau_2 = \left\lfloor \frac{\ln 2}{c_1 c_2} \sqrt{T} \right\rfloor, \\ \eta_1 = \frac{c_1}{c_2 \sqrt{T}}, & \tau_1 = \left\lceil \frac{K-1}{M-1} \right\rceil \left\lfloor \frac{\ln 2}{c_1 c_2} \sqrt{T} \right\rfloor, \end{cases} \tag{31}$$

where $c_1 = \sqrt{\frac{\ln K}{\frac{5}{2} + \min\{M, K-M\} \cdot 2\beta_1}}$ and $c_2 = \frac{4(K-1)}{M-1}$, we have

$$\begin{aligned} R^{\text{ROW}} \leq \frac{8(K-1)}{M-1} &\sqrt{\frac{5}{2} + \min\{M, K-M\} \cdot 2\beta_1} \sqrt{\ln K} \sqrt{T} \\ &+ \min\{M, K-M\} \cdot \beta_1 + 1, \end{aligned} \tag{32}$$

for $T \geq \frac{448(K-1)^2 \ln K}{\frac{5}{2} + 2\beta_1}$. The result of Theorem 2 then follows. Please see our technical report [21] for the complete proof of Theorem 2.

V. Power-of-2-Arms for a More General Case

In the model studied in Sec. IV, we assume the switching cost is only incurred for changing the set $\hat{\mathbf{k}}(t)$ of that M chosen arms, but there is no switching cost when changing the active arm within this set. Readers may ask whether the power-of-2-arms improvement attained by the ROW algorithm is only because there is no switching cost for changing the active arm. To answer this question, in this section we consider a more general case where switching costs are also incurred for changing the active arm $\mathrm{k}(t)$ chosen from $\hat{\mathbf{k}}(t)$. We will propose another algorithm that also attains $O(\sqrt{T})$ regret, which then confirms that the power-of-2-arms improvement is precisely due to our intelligent use of 2 or more arms, regardless of the presence of switching costs for changing the active arm.

Specifically, in addition to the switching cost β_1 for changing the arm in $\hat{\mathbf{k}}(t)$, if the active arm $\mathrm{k}(t)$ used at time t is different from the one used at time $t-1$, another switching cost $\beta_3 > 0$ will also be incurred. Therefore, the total cost becomes

$$\begin{aligned} \text{Cost}(1:T) \triangleq \sum_{t=1}^{T} \Big\{ & l_{\mathrm{k}(t)}(t) + \beta_1 \sum_{k \in \hat{\mathbf{k}}(t)} \mathbf{1}_{\{k \notin \hat{\mathbf{k}}(t-1)\}} \\ & \beta_3 \mathbf{1}_{\{\mathrm{k}(t) \neq \mathrm{k}(t-1)\}} + \beta_2 z(t) \Big\}, \end{aligned} \tag{33}$$

where the third term on the right-hand-side represents the additional switching cost for changing the active arm $\mathrm{k}(t)$. Note that due to such an additional switching cost, the regret of ROW from Sec. IV could be linear in the time horizon T. This is because within each sub-episode, ROW could use any arm in the chosen working group $\hat{\mathbf{k}}^{\text{ROW}}(t)$ as the active arm (please see Step-4 of Algorithm 2). By doing so, ROW could change the active arm $\mathrm{k}^{\text{ROW}}(t)$ almost every time, which results in a total switching cost that is linear in T.

To address this new issue, we present a new algorithm "Switching Reduced Randomized Online Learning With Working Groups" (SR-ROW), whose total switching cost can be upper-bounded by $O(\sqrt{T})$. The difference between SR-ROW and ROW is in how to choose the active arm at each time within a sub-episode. Specifically, as in Algorithm 2, SR-ROW follows Step-1 to choose the primary arm $k_0^{\text{SR-ROW}}[u]$ for each episode u, follows Step-2 to choose the secondary arms $\hat{\mathbf{k}}_{M-1}^{\text{SR-ROW}}[u,v]$ for each sub-episode (u,v), and follows Step-3 to initialize the weights $\hat{w}_k^{\text{SR-ROW}}(t_{u,v})$ and probabilities $\hat{p}_k^{\text{SR-ROW}}(t_{u,v})$. However, differently from Step-4 of Algorithm 2, SR-ROW chooses the active arm $\mathrm{k}^{\text{SR-ROW}}(t)$ according to the shrinking-dartboard method in [12]. That is, at the beginning of the sub-episode (u,v), i.e., at time $t_{u,v}$, SR-ROW chooses the initial active arm $\mathrm{k}^{\text{SR-ROW}}(t_{u,v})$ with probability $\hat{p}_k^{\text{SR-ROW}}(t_{u,v})$. Then, at each time $t = t_{u,v} + 1, ..., t_{u,v} + \tau_2 - 1$, in order to reduce the switching costs for changing the active arm, SR-ROW will reuse the previous active arm with probability $\frac{\hat{w}^{\text{SR-ROW}}_{\mathrm{k}^{\text{SR-ROW}}(t-1)}(t)}{\hat{w}^{\text{SR-ROW}}_{\mathrm{k}^{\text{SR-ROW}}(t-1)}(t-1)}$. Only with probability $1 - \frac{\hat{w}^{\text{SR-ROW}}_{\mathrm{k}^{\text{SR-ROW}}(t-1)}(t)}{\hat{w}^{\text{SR-ROW}}_{\mathrm{k}^{\text{SR-ROW}}(t-1)}(t-1)}$, SR-ROW will *change* the active arm, in which case she picks an arm k in $\hat{\mathbf{k}}^{\text{SR-ROW}}[u,v]$ as the active arm $\mathrm{k}^{\text{SR-ROW}}(t)$ according to the updated probability $\hat{p}_k^{\text{SR-ROW}}(t)$.

Intuitively, this way of switching will significantly lower the switching costs. Interestingly, using the techniques in [12], we can show that the probability of choosing a given active arm at each time is exactly the same as that of Algorithm 2. Thus, the losses incurred would also be the same. It then only remains to bound the switching costs due to β_3. Note that

$$\begin{aligned}\frac{\hat{w}^{\text{SR-ROW}}_{\mathrm{k}^{\text{SR-ROW}}(t-1)}(t)}{\hat{w}^{\text{SR-ROW}}_{\mathrm{k}^{\text{SR-ROW}}(t-1)}(t-1)} &= e^{-\eta_2 l_{\mathrm{k}^{\text{SR-ROW}}(t-1)}(t-1)} \\ &\approx 1 - \eta_2 l_{\mathrm{k}^{\text{SR-ROW}}(t-1)}(t-1),\end{aligned}$$

where the approximation is because $e^{-x} \approx 1 - x$ when x is small. Then, since $\eta_2 = \Theta(\frac{1}{\sqrt{T}})$ and $0 \leq l_{\mathrm{k}^{\text{SR-ROW}}(t-1)}(t-1) \leq 1$, within each sub-episode of length $O(\sqrt{T})$, SR-ROW changes the active arm only a constant number of times on average. Indeed, we can upper-bound this constant by $\ln 2$. (Please see our technical report [21] for details.) Finally, by using the same values of parameters τ_1, τ_2, η_1 and η_2 in (31), the regret of SR-ROW can be upper-bounded as follows.

Theorem 3. *Consider bandit learning with switching costs and full-feedback costs. When $M \geq 2$ and when there exists a switching cost β_3 for changing the active arm among the chosen M arms, the regret of SR-ROW (with parameters τ_1, τ_2, η_1 and η_2 in (31)) can be upper-bounded as follows, for $T \geq \frac{448(K-1)^2 \ln K}{\frac{5}{2}+2\beta_1}$,*

$$R^{\textit{SR-ROW}}(T) \leq 8b_1 \frac{K-1}{M-1}\sqrt{\ln K}\sqrt{T} + b_2 + \ln 2 \cdot \beta_3 \left\lceil \frac{T}{\tau_2} \right\rceil, \tag{34}$$

where $b_1 = \sqrt{\frac{5}{2} + 2b_3\beta_1}$, $b_2 = b_3\beta_1 + 1$ and $b_3 = \min\{M, K-M\}$.

Notice that the only difference between the regret of SR-ROW in Theorem 3 and the regret of ROW in Theorem 2 is the last term $\ln 2 \cdot \beta_3 \left\lceil \frac{T}{\tau_2} \right\rceil$, which captures the switching costs for changing the active arm. Please see our technical report [21] for the complete proof of Theorem 3.

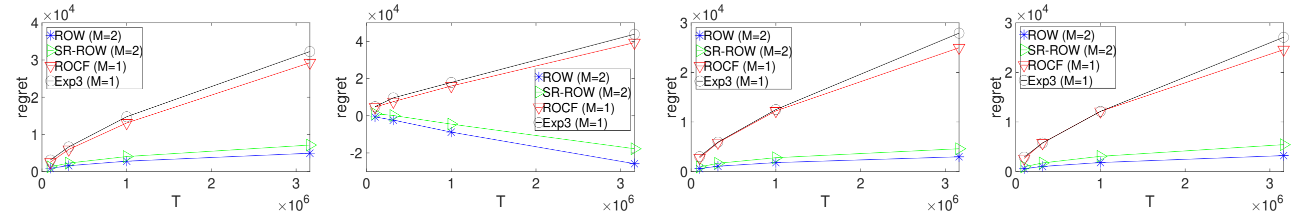
VI. Numerical Results

In this section, using both a generic setting and a more realistic Edge-AI setting, we perform numerical experiments comparing the regrets of our algorithms ROW and SR-ROW for $M \geq 2$ (and ROCF for $M = 1$), and the episodic version of Exp3 proposed in [10]. (Please see our technical report [21] for more numerical results for ROW, SR-ROW and ROCF.) According to [10], the theoretical regret of the episodic version of Exp3 is $\Theta(K^{\frac{1}{3}}T^{\frac{2}{3}})$.

A. Regret Comparisons for a Generic Bandit Setting

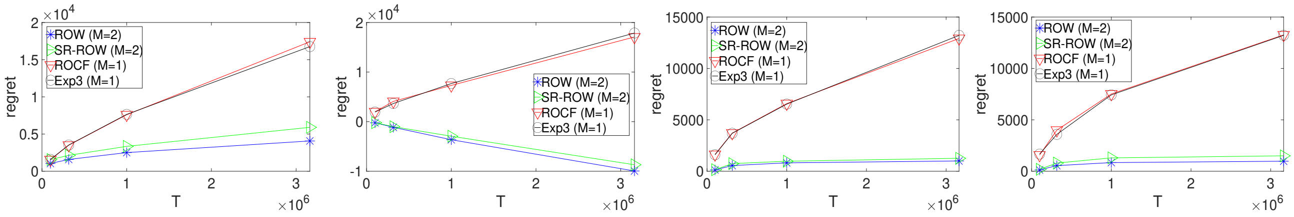
In Fig. 2 and Fig. 3, we use both the lower-bound trace that we designed in Sec. III-B and the three counter-example traces that we designed in Sec. IV-A. We consider $K = 4$ arms, and $M = 2$ for ROW and SR-ROW. (When M increases, the gap between the regret of ROW/SR-ROW and that of Exp3 will further increase.) Note that the switching-cost coefficients β's could be affected by various practical factors, e.g., how much the service provider and the customer dislike the delay, service interruption, and/or communication overhead, etc. As a result, the values of β's could vary significantly across different scenarios. Below, guided by the condition on the relation between β_1 and β_2 in Theorem 1 (i.e., $\frac{3}{4}K\beta_1$ v.s. β_2), we focus on two cases: $\beta_1 = 1$ and $\beta_2 = 1$ in Fig. 2 (which corresponds to the setting when the user views model switching as costly as consulting the cloud), and $\beta_1 = 0.1$ and $\beta_2 = 1$ in Fig. 3 (which corresponds to the setting that the user views model switching to be less costly than consulting the cloud). We find that our main conclusions below hold across these different settings and are robust to the β values.

Specifically, we compare how the regret increases with the time length T. From Fig. 2 and Fig. 3, we can see that for all 4 traces, the regret of ROW (with $M = 2$) is much smaller than that of Exp3 (and ROCF). For example, when using counter-example 3 and $T = \sqrt{10} \times 10^6$ in Fig. 2, the regret of Exp3 is around 2.61×10^4. In contrast, the regret of ROW is only about 3.22×10^3, confirming the power of using 2 arms. For $M = 1$, the regret of ROCF is also smaller than that of Exp3. This is because the choice of β_1 and β_2 here satisfies $\beta_2 \leq \frac{3}{4}K\beta_1$. As we show in (3) and (11), this is the range where costly full-feedback is helpful for reducing the regret when $M = 1$. When



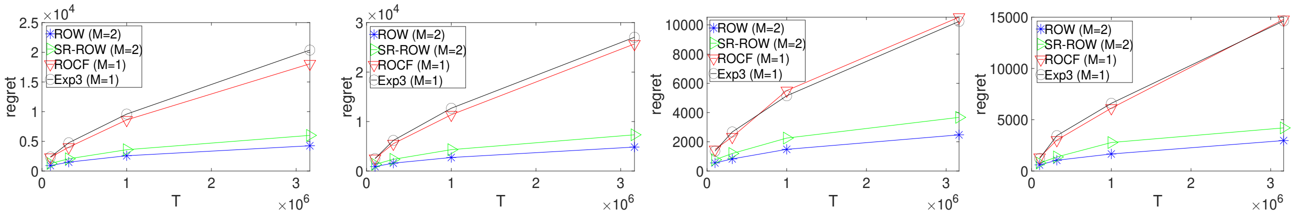
(a) Using the lower-bound trace. (b) Using counter-example 1. (c) Using counter-example 2. (d) Using counter-example 3.

Fig. 2: Compare ROW, SR-ROW, ROCF and the episodic version of Exp3. ($\beta_1 = 1$, $\beta_2 = 1$, synthetic data.)



(a) Using the lower-bound trace. (b) Using counter-example 1. (c) Using counter-example 2. (d) Using counter-example 3.

Fig. 3: Compare ROW, SR-ROW, ROCF and the episodic version of Exp3. ($\beta_1 = 0.1$, $\beta_2 = 1$, synthetic data.)



(a) With more easy-to-analyze images. ($\beta_1 = 1$, $\beta_2 = 1$.) (b) With more hard-to-analyze images. ($\beta_1 = 1$, $\beta_2 = 1$.) (c) With more easy-to-analyze images. ($\beta_1 = 0.1$, $\beta_2 = 1$.) (d) With more hard-to-analyze images. ($\beta_1 = 0.1$, $\beta_2 = 1$.)

Fig. 4: Compare the regrets of ROW, SR-ROW, ROCF and the episodic version of Exp3. (Real-world data.)

β_2 increases to values larger than $\frac{3}{4}K\beta_1$, the gap between the regret of ROCF and that of Exp3 will diminish (see Fig. 3). Note that all the above results do not consider the switching cost for changing the active arm (i.e., $\beta_3 = 0$). To evaluate SR-ROW, we further use $\beta_3 = 1$. In Fig. 2 and Fig. 3, we can see that the regrets of SR-ROW are close to that of ROW and are still much smaller than that of Exp3 (and ROCF).

B. Regret Comparisons for a More Realistic Edge-AI Setting

We then consider a more realistic Edge-AI setting, where an edge server uses various types of ML models to analyze incoming images. For such tasks, the inference accuracy and latency have been shown to be two important factors that affect the performance of Edge AI systems [24]. Specifically, if the edge server uses a simple ML model with low inference latency, the performance will be good when the incoming image is clear and easy to be analyzed (see Fig. 1a in [24] as an example). However, when the incoming image is obscure and hard to be classified (see Fig. 1c in [24] as an example), the accuracy could be very bad. On the other hand, if a more sophisticated ML model with high inference accuracy is used, while the accuracy will be better for hard-to-analyze images, the inference latency will be unnecessarily high for easy-to-analyze images. Thus, there is a need to *adaptively select* the ML models based on the incoming images.

Towards this end, we use the MS COCO dataset [25]. We first train three ML models based on YOLOv7 [26]: YOLOv7-W6, YOLOv7-E6, and YOLOv7-D6. YOLOv7-W6 is with the worst accuracy but the lowest inference latency, while YOLOv7-D6 is with the best accuracy but the highest inference latency. Then, we use these $K = 3$ models as the arms in an online bandit-learning problem, and apply ROW (and SR-ROW) with $M = 2$. For the feedback $l_k(t)$, we use the sum of the confidence level reported by the ML model k and its inference latency as the loss, both of which are scaled back to $[0, 1]$. These losses are averaged every 10 images and fed back to the online algorithm. We let the switching cost and full-feedback cost be $\beta_1 = \beta_2 = 1$. To evaluate SR-ROW, we further use $\beta_3 = 1$. In Fig. 4, we compare how the regret increases with the time length T. In Fig. 4a, we plot the regrets for the case when around 80% images are easy to be analyzed, while Fig. 4b is for the case when around 80% images are hard to be analyzed. Each incoming image is chosen *i.i.d.* between the easy-to-analyze and hard-to-analyze images. From Fig. 4, we can see that for both of these two traces, the regret of ROW and SR-ROW with $M = 2$ is much smaller than that of Exp3 (and ROCF). The results thus suggest that our

ROW and SR-ROW algorithms are more efficient than existing algorithms in balancing inference accuracy, inference latency and switching cost.

VII. Conclusion

In this paper, we investigate adversarial bandit-learning problems with switching costs and full-feedback costs. First, when only $M = 1$ arm is pulled at each time, we provide a lower bound (and a matching upper bound) of the regret. Our new bounds show that adding costly full-feedback will not alter the $\Theta(T^{\frac{2}{3}})$ regret for $M = 1$, while the dependence on K could be improved when the full-feedback cost β_2 is small. Second, when $M \geq 2$ arms can be chosen at each time, we provide a novel online learning algorithm ROW that improves the regret to $O(\sqrt{T})$ without even using full feedback. Our result thus reveals that having 2 (or more) arms is surprisingly as powerful as having free full-feedback, for obtaining a low regret in bandit-learning problems with switching costs. Our algorithm ROW and regret analysis involve several new ideas, e.g., using different weight-decay parameters inside and across episodes. Our numerical results confirm that the regret of our algorithm ROW is much smaller than that of the episodic version of Exp3.

There are several interesting directions of future work. First, notice that we study the static regret. It would be interesting to extend our study to the dynamic regret, where the optimal arm changes in time. Second, ROW assumes the knowledge of the time length T. It would be useful to extend ROW to the setting where T is not known in advance. Third, it remains open whether the benefit of power-of-2-arms holds in contextual bandits.

References

[1] M. Shi, X. Lin, and L. Jiao, "Power-of-2-arms for bandit learning with switching costs," in *Proceedings of the Twenty-Third International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2022, pp. 131–140.

[2] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[3] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[4] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys*, vol. 46, no. 4, pp. 1–37, 2014.

[5] R. Arora, T. V. Marinov, and M. Mohri, "Bandits with feedback graphs and switching costs," in *Advances in Neural Information Processing Systems*, 2019, pp. 10 397–10 407.

[6] J. Steiger, B. Li, B. Ji, and N. Lu, "Constrained bandit learning with switching costs for wireless networks," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[7] G. Theocharous, P. S. Thomas, and M. Ghavamzadeh, "Ad recommendation systems for life-time value optimization," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1305–1310.

[8] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[9] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM journal on computing*, vol. 32, no. 1, pp. 48–77, 2002.

[10] R. Arora, O. Dekel, and A. Tewari, "Online bandit learning against an adaptive adversary: from regret to policy regret," in *Proceedings of 29th International Conference on Machine Learning*, 2012, pp. 1747–1754.

[11] O. Dekel, J. Ding, T. Koren, and Y. Peres, "Bandits with switching costs: $T^{2/3}$ regret," in *Proceedings of 46th annual ACM symposium on Theory of computing*, 2014, pp. 459–467.

[12] S. Geulen, B. Vöcking, and M. Winkler, "Regret minimization for online buffering problems using the weighted majority algorithm," in *Conference on Learning Theory*. Citeseer, 2010, pp. 132–143.

[13] Y. Seldin, P. Bartlett, K. Crammer, and Y. Abbasi-Yadkori, "Prediction with limited advice and multiarmed bandits with paid observations," in *Proceedings of 31st International Conference on Machine Learning*, 2014, pp. 280–287.

[14] R. Combes, M. S. Talebi Mazraeh Shahi, and A. Proutiere, "Combinatorial bandits revisited," in *Advances in Neural Information Processing Systems*, 2015, pp. 2116–2124.

[15] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[16] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1094–1104, 2001.

[17] A. Blum and Y. Monsour, "Learning, regret minimization, and equilibria," *Algorithmic Game Theory*, 2007.

[18] A. Slivkins, "Introduction to multi-armed bandits," *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.

[19] P. Domingos, "A unified bias-variance decomposition," in *Proceedings of 17th International Conference on Machine Learning*, 2000, pp. 231–238.

[20] A. C.-C. Yao, "Probabilistic computations: Toward a unified measure of complexity," in *18th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 1977, pp. 222–227.

[21] M. Shi, X. Lin, and L. Jiao, "Power-of-2-arms for bandit learning with switching costs," Purdue University, Tech. Rep., 2023. Available at https://engineering.purdue.edu/\%7elinx/papers.html.

[22] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2019.

[23] H. Levy, "Stochastic dominance and expected utility: Survey and analysis," *Management science*, vol. 38, no. 4, pp. 555–593, 1992.

[24] B. Taylor, V. S. Marco, W. Wolff, Y. Elkhatib, and Z. Wang, "Adaptive deep learning model selection on embedded systems," *ACM SIGPLAN Notices*, vol. 53, no. 6, pp. 31–43, 2018.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[26] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 7464–7475.

Appendix A
Sketch of Proof of Lemma 4

Please see our technical report [21] for the complete proof of Lemma 4. In the following, we sketch the key steps (Step 1 - Step 3 below) for proving Lemma 4, which may also be of independent interest.

Sketch of proof of Lemma 4:

Step-1: Similar to Lemma 3, we can derive a lower bound of $g_1[u]$ by relating it to the expectation and variance of the loss differences.

Lemma 6. *For each episode u, given the history $\mathcal{H}[u-1]$ and the chosen working groups $\hat{\mathbf{k}}^{ROW}[u, 1:V]$, if $\eta_1 \tau_1 \leq \ln 2$, we have*

$$g_1[u] \geq \mathbb{E}\left[\tilde{L}^{ROW}[u] \middle| \mathcal{H}[u-1], \hat{\mathbf{k}}^{ROW}[u, 1:V]\right] - \eta_1 \cdot \mathit{Var}\left(\tilde{L}^{ROW}[u] \middle| \mathcal{H}[u-1], \hat{\mathbf{k}}^{ROW}[u, 1:V]\right), \quad (35)$$

where the expectation is taken with regard to the randomness in $p_k^{ROW}[u]$, i.e.,

$$\mathbb{E}\left[\tilde{L}^{ROW}[u]\middle|\mathcal{H}[u-1],\hat{\mathbf{k}}^{ROW}[u,1:V]\right] \triangleq \sum_{k=1}^{K} p_k^{ROW}[u]\tilde{L}^{ROW}[u],$$

$$Var\left(\tilde{L}^{ROW}[u]\middle|\mathcal{H}[u-1],\hat{\mathbf{k}}^{ROW}[u,1:V]\right) \triangleq \sum_{k=1}^{K} p_k^{ROW}[u] \cdot \left(\tilde{L}^{ROW}[u]-\mathbb{E}\left[\tilde{L}^{ROW}[u]\middle|\mathcal{H}[u-1],\hat{\mathbf{k}}^{ROW}[u,1:V]\right]\right)^2.$$

Please see our technical report [21] for the complete proof of Lemma 6.

Step-2: Lemma 4 is then proved by mainly comparing the expectations of (22) and (35) with regard to the randomness in the working groups. Here, we use the help of a fictitious "full feedback" system, where we assume that there is an oracle who knows the losses from all arms in all time-slots during the episode. Further, this oracle assigns the probability distribution $p_k^{\text{ROW}}[u]$ on the arms.

It is easy to show that the expectations of both working-group feedback and the loss differences are related to the expectation of the fictitious "full feedback". Further, Lemma 7 and Lemma 8 below show that the variances of both the working-group feedback and loss differences can also be related to the variance of full feedback, given by $\text{Var}\left(L[u,v]|\mathcal{H}[u-1]\right)$ in the lemma below.

Lemma 7. *For each sub-episode (u,v), given the history $\mathcal{H}[u-1]$, we have*

$$\mathbb{E}_{\hat{\mathbf{k}}^{ROW}[u,v]}\left[Var\left(L[u,v]\middle|\hat{\mathbf{k}}^{ROW}[u,v]\right)\middle|\mathcal{H}[u-1]\right] \geq \frac{M-1}{K-1}\cdot Var\left(L[u,v]|\mathcal{H}[u-1]\right), \tag{36}$$

where

$$Var\left(L[u,v]|\mathcal{H}[u-1]\right) \triangleq \sum_{k=1}^{K} p_k^{ROW}[u]\left(L[u,v]-\sum_{k=1}^{K}p_k^{ROW}[u]L[u,v]\right)^2.$$

The variance on the left-hand-side of (36) is for the losses from the feedback in the working group $\hat{\mathbf{k}}^{\text{ROW}}[u,v]$. The outside expectation is taken over all possible working groups. The variance on the right-hand-side of (36) is for the fictitious "full feedback". Intuitively, if the right-hand-side of (36) is strictly positive, there must be some difference among the losses of the arms. Then, even when a random subset of arms is chosen into the working group, we should still see some variance. That is the intuition why the left-hand-side of (36) must also be strictly positive, which is the conclusion in Lemma 7. Moreover, as M increases, the constant factor $\frac{M-1}{K-1}$ increases to be closer to 1. This is one of the reasons that the regret of ROW decreases with M. In sharp contrast, when $M=1$, we have $\frac{M-1}{K-1}=0$. Indeed, in this case, no matter how large the variance of full feedback is, the variance on the left-hand-side of (36) will always be equal to 0. This is one of the reasons for the sharp transition from the $O(T^{\frac{2}{3}})$ regret when $M=1$ to the $O(\sqrt{T})$ regret when $M\geq 2$. Please see our technical report [21] for the complete proof of Lemma 7.

Step-3: However, the fictitious "full feedback" is not available to the online learning algorithm. Hence, Lemma 7 is not very useful unless we can related the full feedback to the loss difference that we design in (16). This is exactly the purpose of Lemma 8 below.

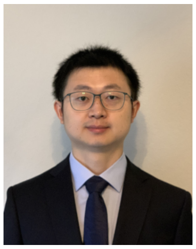
Lemma 8. *For each episode u, given the history $\mathcal{H}[u-1]$, we have*

$$\sum_{v=1}^{V} Var\left(L[u,v]|\mathcal{H}[u-1]\right) \geq \frac{M-1}{2(K-1)} \cdot \mathbb{E}_{\hat{\mathbf{k}}^{ROW}[u,1:V]}\left[Var\left(\tilde{L}^{ROW}[u]\middle|\hat{\mathbf{k}}^{ROW}[u,1:V]\right)\middle|\mathcal{H}[u-1]\right]. \tag{37}$$

Different from Lemma 7, Lemma 8 focuses on the variance of the loss differences $\tilde{L}^{\text{ROW}}[u]$, as in the right-hand-side of (37). Moreover, the expectation is taken over all possible sequences of the working groups for the whole episode. Thus, the variance of full feedback on the left-hand-side of (37) is also summed over all sub-episodes v. Intuitively, if the right-hand-side of (37) is strictly positive, there must exist some difference across the secondary arms when comparing with the common primary arm. Then, the differences among the secondary arms cannot all be 0. This means there must be some variance of the full feedback. This is the intuition why the left-hand-side of (37) must also be strictly positive, which is the conclusion in Lemma 8. Similar to that in (36), as M increases, the constant factor $\frac{M-1}{2(K-1)}$ increases to be closer to 1. This is another reason that the regret of ROW decreases with M. In sharp contrast, when $M=1$, we have $\frac{M-1}{2(K-1)}=0$, which again implies a sharp transition from $M=1$ to $M\geq 2$. By comparing the constant factors in (22) and (36) with that in (35) and (37), we can see that to obtain (25), η_2 needs to be larger than $16\left(\frac{K-1}{M-1}\right)^2\cdot\eta_1$. Please see our technical report [21] for the complete proof of Lemma 8.

Remark 1. *Lemma 8 is the result of using our idea 3. In other words, without our idea 3 for constructing the loss differences $\tilde{L}_k^{ROW}[u]$ in (16), Lemma 8 may not hold. For example, in the counter-example 3 that we introduced in Sec. IV-A, the variance of full feedback is 0. Without our idea 3, the variance of the absolute loss from the feedback in all sub-episodes will be $\Theta(\tau_2^2)$, which would have made Lemma 8 invalid. In contrast, with our idea 3, the loss difference is the difference from the loss of the primary arm, which will be 0 for all arms. Thus, the variance of the loss differences of ROW in each episode is 0, which is the same as the variance of full feedback.*

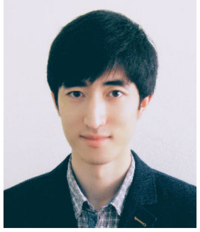
Combining Lemma 3, Lemma 6, Lemma 7, and Lemma 8, we can then prove Lemma 4.



Ming Shi is a Post-Doctoral Scholar at the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH, USA. He received his B.S. degree from Tianjin University, Tianjin, China, in 2015, and his Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 2022. Dr. Shi's research interests are in the theoretical analysis of machine learning and online optimization algorithms, with applications in networking, wireless communication, and Edge AI.

Xiaojun Lin (S'02 M'05 SM'12 F'17) received his B.S. from Zhongshan University, Guangzhou, China, in 1994, and his M.S. and Ph.D. degrees from Purdue University, West Lafayette, IN, in 2000 and 2005, respectively. He joined the School of Electrical and Computer Engineering at Purdue University in 2005, and became a Professor of ECE in 2017. Since June 2023, he joined the Department of Information Engineering, The Chinese University of Hong Kong, as a Professor and Global STEM Scholar.

Dr. Lin's research interests are in the analysis, control and optimization of large and complex networked systems, including both communication networks and power grid. He received the NSF CAREER award in 2007. He received 2005 best paper of the year award from Journal of Communications and Networks, IEEE INFOCOM 2008 best paper award, ACM MobiHoc 2021 best paper award, and ACM e-Energy 2022 best paper award. He was the Workshop co-chair for IEEE GLOBECOM 2007, the Panel co-chair for WICON 2008, the TPC co-chair for ACM MobiHoc 2009, the Mini-Conference co-chair for IEEE INFOCOM 2012, and the General co-chair for ACM e-Energy 2019. He has served as an Area Editor for (Elsevier) Computer Networks Journal, an Associate Editor for IEEE/ACM Transactions on Networking, and a Guest Editor for (Elsevier) Ad Hoc Networks journal.



Lei Jiao received the Ph.D. degree in computer science from the University of Göttingen, Germany. He is currently a faculty member at the University of Oregon, USA. Previously he worked at Nokia Bell Labs in Ireland. He is interested in the mathematics of optimization, control, learning, and mechanism design applied to computer and telecommunication systems, networks, and services. He is a recipient of the NSF CAREER award. He publishes papers in journals such as JSAC, ToN, TPDS, and TMC and in conferences such as INFOCOM, MOBIHOC, ICNP, ICDCS, SECON, and IPDPS. He also received the Best Paper Awards of IEEE LANMAN 2013 and IEEE CNS 2019, and the 2016 Alcatel-Lucent Bell Labs UK and Ireland Recognition Award. He has been on the program committees of INFOCOM, MOBIHOC, ICDCS, TheWebConf, and IWQoS, and served as the program chair of multiple workshops with INFOCOM and ICDCS.