Learning to Interact with Learning Agents

Adish Singla
MPI-SWS
Saarbrücken, Germany
adishs@mpi-sws.org

Hamed Hassani
University of Pennsylvania
Philadelphia, USA
hassani@seas.upenn.edu

Andreas Krause
ETH Zurich
Zurich, Switzerland
krausea@ethz.ch

Abstract

AI and machine learning methods are increasingly interacting with and seeking information from people, robots, and other learning agents. Consequently, the learning dynamics of these agents creates fundamentally new challenges for existing methods. Motivated by the application of learning to offer personalized deals to users, we highlight these challenges by studying a variant of the framework of "online learning using expert advice with bandit feedback". In our setting, we consider each expert as a learning agent, seeking to more accurately reflect real-world applications. The bandit feedback leads to additional challenges in this setting: at time t, only the expert i^t that has been selected by the central algorithm (forecaster) receives feedback from the environment and gets to learn at this time. A natural question to ask is whether it is possible to be competitive with the best expert j^* had it seen all the feedback, *i.e.*, competitive with the policy of always selecting expert j^*. We prove the following hardness result—without any coordination between the forecaster and the experts, it is impossible to design a forecaster achieving no-regret guarantees. We then consider a practical assumption allowing the forecaster to *guide* the learning process of the experts by blocking some of the feedback observed by them from the environment, *i.e.*, restricting the selected expert i^t to learn at time t for some time steps. With this additional coordination power, we design our forecaster LIL that achieves no-regret guarantees, and we provide regret bounds dependent on the learning dynamics of the best expert j^*.

Introduction

Many real-world applications involve repeatedly making decisions under uncertainty—for instance, choosing one of the several products to recommend to a user in an online recommendation service, or dynamically allocating resources among available stock options in a financial market. AI methods and machine learning techniques driving these applications are typically designed—or operate under the assumption—to be interacting with static components, *e.g.*, users' preferences are fixed, or domain experts / trading tools providing stock recommendations are static. This assumption is often violated in modern applications as these methods are increasingly interacting with and seeking information from people, robots, and other learning agents. In this paper, we highlight the fundamental challenges in designing algorithms that have to interact with learning agents, especially when algorithm's decisions directly affect the learning dynamics of these agents.

Motivated by the application of learning to offer personalized deals to users, we focus on a well-studied framework of "online learning using expert advice with bandit feedback" (Littlestone and Warmuth 1994; Cesa-Bianchi et al. 1997; Freund and Schapire 1995; Auer et al. 2002; Cesa-Bianchi and Lugosi 2006; Bubeck and Cesa-Bianchi 2012). This is a generic framework for sequential decision making under uncertainty and addresses the fundamental question of how a learning algorithm should trade-off exploration (the cost of acquiring new information) versus exploitation (acting greedily based on current information to minimize instantaneous losses). In this paper, we investigate this framework with an important practical consideration: *How do we use the advice of experts when they themselves are learning agents?*

Motivating Applications

Modeling experts as learning agents realistically captures many practical scenarios of how one would define/encounter these experts in real-world applications, such as seeking advice from fellow players or friends, aggregating prediction recommendations from trading agents or different marketplaces, product testing with human participants who might adapt over time, and information acquisition from crowdsourcing participants who might learn over time. A specific instance of this problem setting is that of meta-learning whereby different learning algorithms (*e.g.*, with different hyperparameters or loss functions) are treated as experts (Baram, El-Yaniv, and Luz 2004; Hsu and Lin 2015; Maillard and Munos 2011; Agarwal et al. 2017).

As a concrete running example, we consider the problem of learning to offer personalized deals / discount coupons to users enabling new businesses to incentivize and attract more customers (Edelman, Jaffe, and Kominers 2011; Singla, Tschiatschek, and Krause 2016; Hirnschall et al. 2018). An emerging trend is *deal-aggregator* sites like *Yipit*[1] providing personalized coupon recommendation services to their users by aggregating and selecting coupons from *daily-*

[1] `http://yipit.com/`; `http://www.groupon.com/`; `https://livingsocial.com/`

deal marketplaces like *Groupon* and *LivingSocial*[1]. One of the primary goals of these recommendation systems like *Yipit* (corresponding to the central algorithm / forecaster in our setting) is to design better selection strategies for choosing coupons from different marketplaces (corresponding to the experts in our setting). However, these marketplaces themselves would be learning to optimize the coupons to offer, for instance, the discount price or the coupon type based on historic interactions with users (Edelman, Jaffe, and Kominers 2011).

Overview of Our Approach and Main Results

Our goal is to design a central online algorithm (henceforth, called as forecaster) to seek the advice of the available experts—more specifically, at time t, the forecaster selects an expert i^t, performs an action $a^t_{i^t}$ recommended by the expert i^t, and observes/incurs a loss $l^t(a^t_{i^t})$ set by the adversary. Furthermore, given the bandit setting, only the selected expert i^t receives feedback from the environment and gets to learn at time t; all other experts that have not been selected at time t experience no change in their learning state at this time. A natural benchmark in our problem setting is to be competitive with the best expert j^* had it seen all the feedback, *i.e.*, competitive with the cumulative loss one would incur by following the policy of always selecting expert j^*.

Generic setting and the hardness result. The fundamental challenge in our setting arises from the fact that the forecaster's selection of experts affects which expert gets to learn at a particular time. In this paper, we establish the following hardness result—without any coordination between the forecaster and the experts, it is impossible to design a forecaster achieving no-regret guarantees when competing with the policy which always selects the expert j^*.

Additional coordination via blocking the feedback. In light of this hardness result, we next explore practically applicable approaches where it is possible to achieve no-regret for the forecaster. In order to make our results applicable to a wide range of real-world applications mentioned above, the focus of this paper is on a generic black-box approach in which the forecaster does not know the internal learning dynamics of the experts. The specific coordination protocol that we consider (alternatively, we can think of this as the additional power at the hands of the forecaster) is as follows: At a time t, the forecaster could decide to *block* the feedback from being observed by the selected expert i^t, thereby restricting the selected expert from learning at time t for some time steps. For instance, in the motivating application of offering personalized deals to users, the deal-aggregator site (forecaster) primarily interacts with users on behalf of the individual daily-deal marketplaces (experts) and hence could control the flow of feedback (*e.g.* users' bids or clicks denoting their purchase decisions) to these marketplaces. With this additional coordination, we design our forecaster LIL that achieves no-regret guarantees with regret bounds dependent on the learning dynamics of the best expert j^*.

Connections to existing results. To conclude the overview of our results, we would like to point out a few relevant papers. First, Maillard and Munos (2011) introduced the EXP4/EXP3 algorithm, *i.e.*, EXP4 meta-algorithm with experts executing EXP3 algorithms proving a regret bound of $\mathcal{O}(T^{\frac{2}{3}})$. Second, in a recent work contemporary to ours, Agarwal et al. (2017) provide improved regret bound of $\mathcal{O}(T^{\frac{1}{2}})$ (in comparison to the above-mentioned regret bound of $\mathcal{O}(T^{\frac{2}{3}})$) for the problem of designing meta-algorithm combining multiple bandit algorithms. Agarwal et al. (2017) also prove a hardness result similar in spirit to that of ours. However, all these existing meta-algorithms are based on the idea of feeding unbiased estimate of losses to the experts and are not directly applicable to our motivating applications where experts could be implementing learning algorithms with more complex feedback structure (*e.g.*, dynamic-pricing algorithm based on the partial monitoring framework (Cesa-Bianchi and Lugosi 2006; Bartók et al. 2014)), or experts being human agents who are learning over time. Below, we highlight two technical points of how our approach and main results different from these existing results:

- Our coordination approach of blocking the feedback observed by experts (*i.e.*, making a binary decision, instead of modifying losses as done in existing approaches) is more suitable for real-world application scenarios, especially in situations where experts' learning algorithms are not directly controlled by the forecaster and have complex feedback structure. Alternatively, when viewing this coordination in terms of communication between the forecaster and the selected expert i^t, our coordination can be achieved with a 1-bit of communication at time t, whereas the coordination in existing approaches requires communicating the probability of selecting expert i^t at time t.
- Our results apply to a rich class of no-regret online learning algorithms that experts might be implementing—the key ingredient of our results relies on proving a property, we termed as *smooth no-regret* learning dynamics. This property quantifies the robustness of an online learning algorithm w.r.t. the sparsity in the observed feedback and is of independent interest.

Generic Setting: The Model

In this section, we formally introduce the generic problem setting and discuss our objective. We have the following entities in our setting: (i) a central algorithm ALGO as the forecaster; (ii) an adversary ADV acting on behalf of the environment; and (iii) N experts $\text{EXP}_j\ \forall j \in \{1, \dots N\}$ (henceforth denoted as $[N]$). Protocol 1 provides a high-level specification of the interaction between these entities. In subsequent sections, we will introduce an additional coordination allowing the forecaster to block the feedback observed by the selected expert for some time steps (*i.e.*, modifying the specification in line 7 of the Protocol 1).

Specification of the Interaction

The sequential decision making process proceeds in rounds $t = 1, 2, \dots, T$ (henceforth denoted as $[T]$); for simplicity we assume that T is known in advance to the forecaster and the results in this paper can be extended to an unknown horizon via the usual doubling trick (Cesa-Bianchi and Lugosi 2006).

Protocol 1: The interaction between adversary ADV, forecaster ALGO, and experts

```
foreach t = 1, 2, ..., T do
    /* Adversary generates the following */
    a private loss vector l^t for the forecaster, i.e., l^t(a) ∀ a ∈ A
    a private feedback vector f^t for the experts, i.e., f^t(a) ∀ a ∈ A
    /* Selecting an expert and performing an action */
    ALGO selects an expert i^t ∈ [N] denoted as EXP_{i^t}
    ALGO performs the action a^t_{i^t} recommended by EXP_{i^t}
    /* Feedback and updates */
    ALGO incurs (and observes) loss l^t(a^t_{i^t}) and updates its selection strategy
    ∀j ∈ [N] : j ≠ i^t, EXP_j does not observe any feedback and makes no update
    EXP_{i^t} observes the feedback f^t(a^t_{i^t}) from the environment and updates its learning state
```

However, we do not assume that T is known to the experts. Each expert EXP_j where $j \in [N]$ is associated with a set of actions $\mathcal{A}_j$ and the action set of the forecaster ALGO is given by $\mathcal{A} = \cup_{j \in [N]} \mathcal{A}_j$.[2] For the clarity of presentation in defining the loss and feedback vectors, we will consider that the action sets of experts are disjoint.[3] The actions here could represent simple discrete actions (*e.g.*, offering a discount coupon of a particular type) or could also represent functional policies defined over a time-dependent context (*e.g.*, mapping user features to the value of a discount coupon).

At any time t, the adversary ADV generates a private loss vector l^t (*i.e.*, $l^t(a) \; \forall \; a \in \mathcal{A}$) for the forecaster and a private feedback vector f^t (*i.e.*, $f^t(a) \; \forall \; a \in \mathcal{A}$) for the experts—see examples below for the specific notion of feedback. Additionally, the adversary ADV generates a publicly available context that is accessible to all the experts at time t—this context essentially encodes any side information from the environment at time t (*e.g.*, user's features at time t).[4]

Simultaneously, the forecaster ALGO (possibly with some randomization) selects expert EXP_{i^t} to seek advice. The selected expert EXP_{i^t} recommends an action $a_{i^t} \in \mathcal{A}_{i^t} \subseteq \mathcal{A}$ (possibly with its internal randomization) which is then performed by the forecaster. The forecaster ALGO observes and incurs the loss $l^t(a^t_{i^t})$, and updates its strategy on how to select experts in the future. All the experts apart from the one selected, $\text{EXP}_j \; \forall \; j \neq i^t$, observe no feedback and make no update—these experts do not experience *any* change in their learning state at this time. The selected expert EXP_{i^t} observes a feedback from the environment denoted as $f^t(a^t_{i^t})$ and performs one learning step.

We assume that losses are bounded in the range $[0, l_{max}]$ for some known $l_{max} \in \mathbb{R}_+$; w.l.o.g. we will use $l_{max} = 1$ (Auer et al. 2002). We consider an oblivious (non-adaptive adversary) as is usual in the literature (Freund and Schapire 1995; Auer et al. 2002), *i.e.*, the loss vector l^t and the feedback vector f^t at any time t do not depend on the actions taken by the forecaster, and hence can be considered to be fixed in advance. Apart from that, no other restrictions are put on the adversary, and it has complete knowledge about the forecaster and the learning dynamics of the experts.

The notion of the feedback and concrete examples. So far, we have considered a generic notion of the feedback received by the selected expert—this feedback essentially depends on the application setting and is supposed to be "compatible" with the learning algorithm used by an expert. For instance, consider an expert EXP_j implementing the EXP3 algorithm and recommending an action a^t_j at time t, then the feedback $f^t(a^t_j)$ received by this expert (if selected at time t) is the loss $l^t(a^t_j)$; for the case of expert EXP_j implementing the HEDGE algorithm, the feedback $f^t(a^t_j)$ received by this expert (if selected at time t) is the set of losses $\{l^t(a) \mid \forall \; a \in \mathcal{A}_j\}$. The feedback could be more general, for instance, receiving a binary signal of acceptance/rejection of the offered deal when an expert is implementing a dynamic-pricing algorithm based on the partial monitoring framework (Cesa-Bianchi and Lugosi 2006; Bartók et al. 2014).

Specification of the Experts

Next, we provide a formal specification of the experts. The focus of this paper is on a black-box approach in which the forecaster ALGO does not know the internal dynamics of the experts. At time t, let us denote an instance of feedback received by EXP_{i^t} by a tuple $h = (a^t_{i^t}, f^t(a^t_{i^t}))$. For any expert EXP_j where $j \in [N]$, let $\mathcal{H}^t_j = (h^1, h^2, \ldots)$ denote the feedback history for EXP_j, *i.e.*, an ordered sequence of feedback instances observed by EXP_j in the time period $[1, t)$. The length $|\mathcal{H}^t_j|$ denotes the number of learning steps for EXP_j up to time t. At time t, the action a^t_j recommended by EXP_j to the forecaster, if this expert is selected, is given by $a^t_j = \pi_j(\mathcal{H}^t_j)$ where π_j is a (possibly randomized) function of EXP_j, taking as input a history of feedback sequence, and outputs an action $a \in \mathcal{A}_j$. Importantly, this history $\mathcal{H}^t_j$ is dependent on the execution of the forecaster ALGO—for clarify of presentation, we denote it as $\mathcal{H}^t_{j,\text{ALGO}}$.

No-regret learning dynamics. To be able to say anything meaningful in this setting, we introduce the constraint of *no-*

[2] The special case of standard multi-armed bandits is captured by the setting when $\mathcal{A}_j$ is a singleton $\forall j \in [N]$.

[3] Note that assuming the disjoint action sets across experts is w.l.o.g., as we can still simulate the shared actions by enforcing that losses/feedbacks for the shared actions are same at any given time.

[4] We have omitted a formal definition of the context as it doesn't directly play a role in the design of our forecaster.

regret learning dynamics on the experts. Let us consider any sequence of a loss vector l and a feedback vector f given by $\mathcal{D} = (l^\tau, f^\tau)_{\tau=\{1,2,\ldots\}}$ generated arbitrarily by the adversary $\textsc{Adv}$ and let $|\mathcal{D}|$ denotes its length. Consider a setting in which the forecaster executes a simple policy which always select a specific expert $\textsc{Exp}_j$ for a fixed $j \in [N]$. Hence, this expert $\textsc{Exp}_j$ gets to see all the feedback and has the *complete* feedback history at every time step—we denote this complete history at any time $\tau \in [|\mathcal{D}|]$ as $\mathcal{H}^\tau_{j,\textsc{Full}}$. Then, the no-regret learning dynamics of $\textsc{Exp}_j$ parameterized by $\beta_j \in [0,1]$ guarantees that the cumulative loss of the forecaster executing this policy satisfies the following:

$$\mathbb{E}\Big[\sum_{\tau=1}^{|\mathcal{D}|} l^\tau\big(\pi_j(\mathcal{H}^\tau_{j,\textsc{Full}})\big)\Big] - \min_{a\in\mathcal{A}_j}\sum_{\tau=1}^{|\mathcal{D}|} l^\tau(a) \leq \mathcal{O}(|\mathcal{D}|^{\beta_j}) \tag{1}$$

where the expectation is w.r.t. the randomization of π_j.

Our Objective: No-Regret Guarantees

As a first attempt in designing the forecaster, one might consider using one of the standard algorithms from the EXP family (*e.g.*, the EXP3 algorithm (Auer et al. 2002) or the NEXP algorithm (McMahan and Streeter 2009)) as the forecaster. This would guarantee that the forecaster has no-regret guarantees using the classical notion of *external* regret (*cf.* Equation 2). However, we argue that external regret is not a desirable objective for our problem setting. We then formally state the guarantees we seek for the forecaster (*cf.* Equation 3).

External regret and its limitations. We begin by formally defining the classical notion of external regret used in the literature (Auer et al. 2002; Cesa-Bianchi and Lugosi 2006; Bubeck and Cesa-Bianchi 2012). Let us consider a complete execution of the forecaster $\textsc{Algo}$ in the retrospect: (i) let $\big(\{a^t_j : j \in [N]\}\big)_{t\in[T]}$ denote actions recommended by the experts during this execution and (ii) let $(l^t)_{t\in[T]}$ denote loss vectors generated by the adversary. In order to define the external regret of the forecaster in this execution, we need to *fix* these actions and loss vectors. Then, the external regret of the forecaster is given by

$$\textsc{RegExt}(T,\textsc{Algo}) := \sum_{t=1}^{T} l^t(a^t_{i^t}) - \min_{j\in[N]}\sum_{t=1}^{T} l^t(a^t_j) \tag{2}$$

If we would have used, let's say, the EXP3 algorithm (Auer et al. 2002) as the forecaster, we would obtain a bound of $\mathcal{O}(T^{\frac{1}{2}})$ on the external regret defined in Equation 2. However, this regret bound is only w.r.t. the post hoc sequence of actions performed and losses observed during the execution of the forecaster—it is not informative of the actual performance of the forecaster when comparing against a policy which always selects the best expert j^*(*cf.* Equation 3 for a formal definition). The reason why this classical notion of regret is not informative in our setting can also be attributed to the fact that losses at any time t are indirectly dependent on which experts were selected by the forecaster in the past as that defines the current learning state of the experts (McMahan and Streeter 2009; Maillard and Munos 2011; Agarwal et al. 2017). This challenge of history-dependent losses also arises when playing against a non-oblivious/adaptive adversary and requires different notions of regret beyond the external regret (Arora, Dekel, and Tewari 2012).

Competing with the best expert. Intuitively, we want to be competitive with the best expert $\textsc{Exp}_{j^*}$ (for any $j^* \in [N]$) had it seen all the feedback, *i.e.*, competitive with the policy of always selecting this expert $\textsc{Exp}_{j^*}$. In fact, executing such a policy ensures that the expert $\textsc{Exp}_{j^*}$ gets full feedback to improve its learning state and perform well w.r.t. the single best action from the set $\mathcal{A}_{j^*}$. We can formally state this alternate notion of regret for the forecaster $\textsc{Algo}$ as follows:

$$\textsc{Reg}(T,\textsc{Algo}) :=$$
$$\sum_{t=1}^{T}\mathbb{E}\Big[l^t\Big(\pi_{i^t}(\mathcal{H}^t_{i^t,\textsc{Algo}})\Big)\Big] - \min_{j\in[N]}\min_{a\in\mathcal{A}_j}\sum_{t=1}^{T} l^t(a) \tag{3}$$

where the expectation is w.r.t. the randomization of the forecaster as well as any internal randomization of the experts.

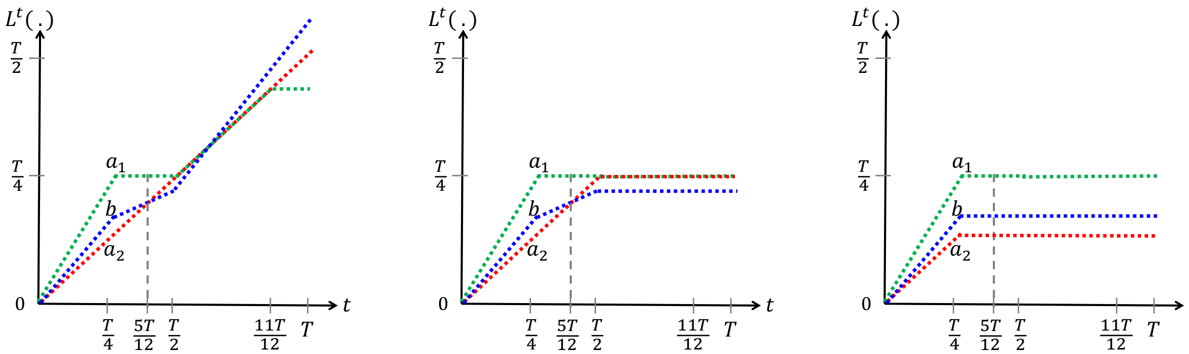
If we already knew who the best expert $\textsc{Exp}_{j^*}$ is at $t = 0$, we could always select this expert—the no-regret learning dynamics from Equation 1 dictates that the regret $\textsc{Reg}(T,\textsc{Algo})$ grows as $\mathcal{O}(T^{\beta_{j^*}})$. The main research question that we study in this paper is how to design an algorithm for the forecaster when we don't have this prior knowledge of j^*. It turns out that competing with this policy of always selecting $\textsc{Exp}_{j^*}$ is a challenging problem in the bandit feedback setting (*cf.* next section for the hardness result). For instance, what might go wrong is that the best expert could have a slow rate of learning/convergence thus incurring high loss in the beginning, misleading the forecaster to essentially "downweigh" this expert. This is turn further exacerbates the problem for the best expert in the bandit feedback setting as this expert will be selected even less and thus have fewer learning steps to improve its state. This adds new challenges to the classic trade-off between exploration and exploitation, suggesting the need to explore at a higher rate.

Generic Setting: Hardness Result

In this section, we highlight the fundamental challenges in designing algorithms that have to interact with learning agents by establishing the following hardness result: in the absence of any coordination between the forecaster and the experts, it is impossible to design a forecaster that achieves no-regret guarantees (*cf.* Equation 3) in the worst-case . Infact, we prove this hardness result when playing against an oblivious (non-adaptive) adversary and when restricting the experts to be implementing well-behaved learning algorithms (*e.g.*, the $\textsc{Hedge}$ algorithm (Freund and Schapire 1995)). We formally state this hardness result in Theorem 1 below.

Theorem 1. *There is a setting in which each of the experts has no-regret learning dynamics with parameter $\beta \leq \frac{1}{2}$; however, any forecaster* $\textsc{Algo}$ *will suffer a linear regret, i.e.,* $\textsc{Reg}(T,\textsc{Algo}) = \Omega(T)$.

The proof is given in the extended version of this paper (Singla, Hassani, and Krause 2018); we briefly outline the



(a) Cumulative losses (L_1) (b) Cumulative losses (L_2) (c) Cumulative losses (L_3)

Figure 1: We have two experts: EXP_1 plays HEDGE and has two actions $\mathcal{A}_1 = \{a_1, a_2\}$, and EXP_2 has only one action $\mathcal{A}_2 = \{b\}$. Figures 1(a), 1(b), and 1(c) shows the *cumulative* loss sequences L_1, L_2, and L_3 for three different scenarios—the adversary at $t = 0$ uniformly at random picks one of these scenarios and uses that loss sequence. These plots show the cumulative losses of the three actions $\mathcal{A} = \{a_1, a_2, b\}$ for three different sequences. The losses are illustrated with the following color scheme—a_1:*green*, a_2:*red*, and b:*blue*.

main ideas below. Our setting for proving this theorem consists of two experts EXP_1 and EXP_2. The first expert EXP_1 has two actions given by $\mathcal{A}_1 = \{a_1, a_2\}$, and the second expert EXP_2 has only one action given by $\mathcal{A}_2 = \{b\}$. The action set of the forecaster ALGO is given by $\mathcal{A} = \{a_1, a_2, b\}$. The expert EXP_1 plays the HEDGE algorithm (Freund and Schapire 1995), *i.e.*, the regret rate parameter is $\beta_1 = 0.5$; the expert EXP_2 has only one action to play, hence $\beta_2 = 0$. Figures 1(a), 1(b), and 1(c) show the *cumulative* loss sequences L_1, L_2, and L_3 for three different scenarios—the adversary at $t = 0$ uniformly at random picks one of these scenarios and uses that loss sequence.

Our main argument is that for any forecaster, one of the scenarios leads to linear regret.[5] In the proof, we consider the case where a forecaster is facing the sequence L_1. We then divide the time horizon T into different slots, and discuss the execution behavior of the forecaster and experts over these time slots. Specifically, our claim is that in the time slot $t \in (\frac{T}{4}, \frac{T}{2}]$, the expert EXP_1 would not be selected for $\frac{T}{12} - o(T)$ time steps. As a result, in the time slot $t \in (\frac{11T}{12}, T]$, the expert EXP_1 would end up recommending action a_2, and a_1 would only be recommended $o(T)$ number of times, leading to $\Omega(T)$ regret for the forecaster. Informally speaking, our negative example shows that the forecaster's selection strategy could add "blind spots" in the feedback history of the experts and that they might not be able to "recover" from this. The fundamental challenge leading to this hardness result is that the forecaster's selection strategy affects the feedback observed by the experts, which in turn alters the learning processes of these experts.

[5]In fact, this hardness result holds even when considering a powerful forecaster which knows exactly the learning algorithms used by the experts, and is able to see the losses $\{l^t(a_1), l^t(a_2), l^t(b)\}$ at every time $t \in [T]$.

Our Approach

In this section, we explore practically applicable approaches where it is possible to achieve no-regret guarantees for the forecaster. Our setting is similar to the problem of designing meta-algorithm for combining different bandit algorithms (Maillard and Munos 2011; Bubeck and Cesa-Bianchi 2012; Agarwal et al. 2017). However, existing meta-algorithms—for instance, the EXP4/EXP3 algorithm (Maillard and Munos 2011) or the CORRAL algorithm (Agarwal et al. 2017)—are based on the idea that the forecaster has the power to modify losses as seen by the experts, thereby, feeding unbiased estimate of losses to the experts. However, these existing meta-algorithms are not directly applicable to our motivating applications where experts could be implementing learning algorithms with more complex feedback structure (*e.g.*, dynamic-pricing algorithm based on the partial monitoring framework (Cesa-Bianchi and Lugosi 2006; Bartók et al. 2014)), or experts being human learning agents.

Additional Coordination and Forecaster LIL

Motivated by the application setting of deal-aggregator sites, we consider a new coordination approach in which the forecaster has the power to guide the experts' learning process by carefully blocking the feedback observed by them. For instance, in the motivating application of offering personalized deals to users, the deal-aggregator site (forecaster) primarily interacts with users on behalf of the individual daily-deal marketplaces (experts) and hence could control the flow of feedback (*e.g.* users' bids or clicks denoting their purchase decisions) to these marketplaces. Formally, at a time t, the forecaster could decide to block the feedback from being observed by the selected expert i^t, thereby restricting the selected expert from learning at time t for some time steps (*i.e.*, modifying the specification in line 7 of the Protocol 1). Compared to the existing meta-algorithms which require modifying losses, our coordination approach is more appli-

Algorithm 2: Forecaster LIL

1 **Parameters**: $\eta \in (0, 1]$
2 **Initialize**: time $t = 1$, weights $w^t_j = 1 \; \forall j \in [N]$
foreach $t = 1, 2, \ldots, T$ **do**
  `/* Selecting an expert */`
3   $\forall j \in [N]$, define probability
$$p^t_j = (1 - \eta) \cdot \frac{w^t_j}{\left(\sum_{k \in [N]} w^t_k\right)} + \frac{\eta}{N}$$
4   Draw i^t from the multinomial distribution $(p^t_j)_{j \in [N]}$
5   Perform action $a^t_{i^t}$ recommended by $\textsc{Exp}_{i^t}$
  `/* Making updates */`
6   Observe loss $l^t(a^t_{i^t})$
7   $\forall j \in [N]$, do the following:
8   Set $\widetilde{l}^t_j = \dfrac{l^t(a^t_{i^t})}{p^t_{i^t}}$ for $j = i^t$, else $\widetilde{l}^t_j = 0$
9   Update $w^{t+1}_j \leftarrow w^t_j \cdot \exp(-\dfrac{\eta \cdot \widetilde{l}^t_j}{N})$
  `/* Blocking the feedback */`
10   $\xi^t \sim Bernoulli(\dfrac{\eta}{N \cdot p^t_{i^t}})$
11   **if** $(\xi^t = 0)$ **then**
12     $\textsc{Exp}_{i^t}$ does **not** observe the feedback $f^t(a^t_{i^t})$ and has **no** change in the learning state

cable to generic black-box settings where experts' learning algorithms are not directly controlled by the forecaster and could have different feedback structure.

With this additional coordination power, we design our forecaster LIL (**L**earning to **I**nteract with **L**earners), presented in Algorithm 2. The selection strategy of the forecaster LIL is similar to the EXP3 algorithm (Auer et al. 2002). The core idea of guiding the experts' learning process is presented in lines 10,11, and 12 of the Algorithm 2. At time t, the forecaster LIL blocks the feedback observed by the selected expert when $\xi^t = 0$ where ξ^t is a $Bernoulli(\frac{\eta}{N \cdot p^t_{i^t}})$ random variable. By choosing this particular random variable, the forecaster LIL ensures that the probability that *any* expert $\textsc{Exp}_j \; \forall j \in [N]$ observes feedback is constant over time $t \in [T]$ and is given by η/N. Considering the negative example used in the proof of Theorem 1, this means that by carefully restricting the feedback, the forecaster LIL avoids any "blind spots" in the feedback history of the experts. However, in order to achieve this, LIL is required to explore at a higher rate, as is evident by the value of η in Theorem 2.

Performance Analysis

In this section, we analyze the performance of the forecaster LIL. We first introduce a novel property, we termed as *smooth no-regret* learning dynamics, quantifying the robustness of an online learning algorithm w.r.t. the sparsity in the observed feedback. Then, the key ingredient of our results (*cf.* Theorem 2) relies on showing that a rich class of no-regret online learning algorithms that experts might be implementing also satisfy this property of smoothness (*cf.* Proposition 1). This property is of independent interest towards designing learning algorithms that are robust against deletion of feedback.

Theoretical Guarantees

Smooth no-regret learning dynamics. In the bandit feedback setting, not all the experts get to observe feedback at a given time, and hence the feedback history would not be *complete* for the experts (*cf.* Equation 1). Consider the same setting as used in defining the no-regret learning dynamics in Equation 1 for a specific expert $\textsc{Exp}_j$. Again, the forecaster executes a simple policy which always select this expert $\textsc{Exp}_j$, however, let us consider a situation where the expert $\textsc{Exp}_j$ only gets to observe the feedback with a probability $\alpha \in (0, 1]$ at a given time. We denote this *sparse* feedback history at any time $\tau \in [|\mathcal{D}|]$ as $\mathcal{H}^\tau_{j,\alpha\text{-}\textsc{Full}}$. Then, the *smooth no-regret* learning dynamics of $\textsc{Exp}_j$ guarantees that the cumulative loss of the forecaster satisfies the following:

$$\mathbb{E}\Big[\sum_{\tau=1}^{|\mathcal{D}|} l^\tau\big(\pi_j(\mathcal{H}^\tau_{j,\alpha\text{-}\textsc{Full}})\big)\Big] - \min_{a \in \mathcal{A}_j} \sum_{\tau=1}^{|\mathcal{D}|} l^\tau(a) \leq \mathcal{O}\Big(\frac{(\alpha \cdot |\mathcal{D}|)^{\beta_j}}{\alpha}\Big) \tag{4}$$

where the expectation is w.r.t. the randomization of function π_j as well as w.r.t. the randomization in generating this sparse feedback history. The following proposition states that a rich class of online learning algorithms indeed have smooth no-regret learning dynamics that can be used by the experts—the proof is given in the extended version of this paper (Singla, Hassani, and Krause 2018).

Proposition 1. *A rich class of no-regret online learning algorithms based on gradient-descent style updates have smooth learning dynamics including the Online Mirror Descent family of algorithms with exact or estimated gradients (Shalev-Shwartz 2011) and Online Convex Programming via greedy projections (Zinkevich 2003).*

No-regret guarantees of LIL. Next, we prove the no-regret guarantees of our forecaster LIL when competing with the best expert $\textsc{Exp}_{j^*}$ (*cf.* Equation 3). The following theorem states the bounds, keeping only the leading terms of T. The proof is given in the extended version of this paper (Singla, Hassani, and Krause 2018).

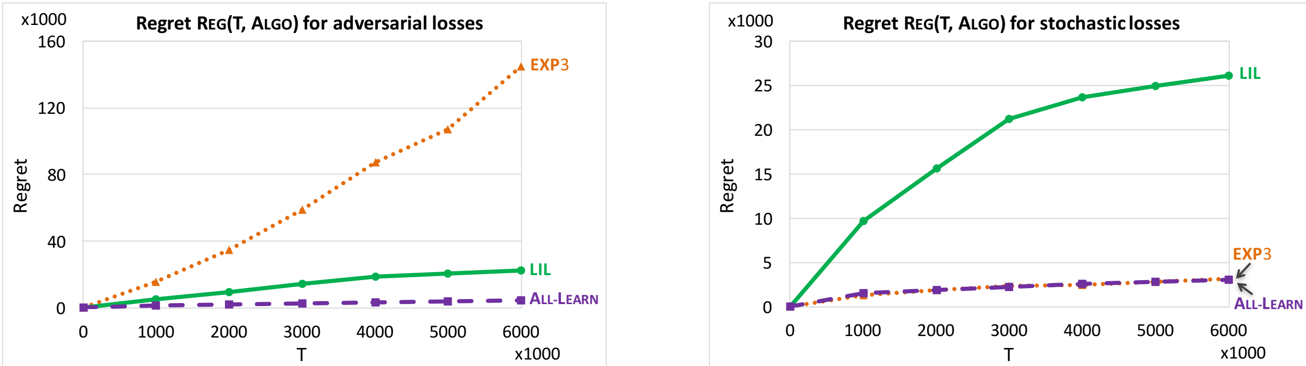
Theorem 2. *Let T be the fixed time horizon. Consider that the best expert $j^* \in [N]$ has no-regret smooth learning dynamics parameterized by $\beta_{j^*} \in [0, 1]$ and* LIL *is invoked with input $\beta \in [0, 1]$ such that $\beta \geq \beta_{j^*}$. Set parameters $\eta = \Theta\big(T^{-\frac{1-\beta}{2-\beta}} \cdot N^{\frac{1-\beta}{2-\beta}} \cdot (\log N)^{(\frac{1}{2} \cdot \mathbf{1}_{\{\beta=0\}})}\big)$. Then, for sufficiently large T, the worst-case expected cumulative regret of the forecaster* LIL *is:*

$$\textsc{Reg}(T, \text{LIL}) \leq \mathcal{O}\big(T^{\frac{1}{2-\beta}} \cdot N^{\frac{1}{2-\beta}} \cdot (\log N)^{(\frac{1}{2} \cdot \mathbf{1}_{\{\beta=0\}})}\big)$$

For the special case of multi-armed bandits (where $\beta = 0$), this regret bound matches the bound of $\Theta(T^{\frac{1}{2}})$—in fact, for this special case, our algorithm LIL is exactly equivalent to EXP3. For an important case when experts are implementing algorithms like $\textsc{Hedge}$ or EXP3 (where $\beta = \frac{1}{2}$), our algorithm LIL achieves the bound of $\mathcal{O}(T^{\frac{2}{3}})$.

Simulation Results

Next, we evaluate the performance of the forecaster LIL via simulations, and compare against the following benchmarks:



(a) Adversarial losses

(b) Stochastic losses

Figure 2: Simulation results showing the performance of LIL against different benchmarks for a setting with two experts (similar to what we used in the proof of Theorem 1). (a) considers adversarial losses from Figure 1(a); (b) considers stochastic losses. Results illustrate that LIL has provable no-regret guarantees in general, and also a fast convergence rate in the stochastic setting.

- **EXP3**: using EXP3 algorithm (Auer et al. 2002) as the forecaster for the specification in Protocol 1.
- **ALL-LEARN**: using EXP3 algorithm (Auer et al. 2002) as the forecaster for a relaxed/easier setting in which all experts $j \in [N]$ observe the feedback at any time t.

Adversarial losses. As our first simulation setting, we consider the same set up used in the proof of Theorem 1 and we use the loss sequence shown in Figure 1(a). For this loss sequence, the loss of actions $A = \{a_1, a_2, b\}$ averaged over $t \in [T]$ is given by $(0.4583, 0.5, 0.7487)$—hence the best expert is EXP_1 and the best action is a_1 (*cf.* Equation 3). Figure 2(a) shows the regret $\text{REG}(T, \text{ALGO})$ for LIL, EXP3, and ALL-LEARN, and illustrates the following points. First, EXP3 suffer a linear regret, as dictated by the hardness result in Theorem 1. Second, LIL has a sub-linear regret as proved in Theorem 2. Note that if we plot $\text{REG}(T, \text{ALGO})$ shown in Figure 2(a) on a `log-log` plot, the slope s of a linear fit on the resulting plot defines the rate T^s of the growth of regret. The slope is $s = 0.62$ for LIL—our results from Theorem 2 dictate an upper bound of 0.66 on LIL's slope (for $\beta = 0.5$).

Stochastic losses. To complement the above results, we next consider a stochastic version of the above setup where losses of actions $A = \{a_1, a_2, b\}$ are sampled i.i.d. from Bernoulli distributions with means given by $(0.45, 0.5, 0.475)$—as before, the best expert is EXP_1 and the best action is a_1 for this stochastic setting. Figure 2(b) shows the regret $\text{REG}(T, \text{ALGO})$ for LIL, EXP3, and ALL-LEARN, and illustrates the following points. First, EXP3 performs better than LIL: this is expected because in the stochastic setting, the strategy to block the feedback only slows down convergence; furthermore, LIL has a higher rate of exploration η compared to EXP3. Second, in the `log-log` plot, the slope is $s = 0.56$ for LIL, $s = 0.47$ for EXP3, and $s = 0.40$ for ALL-LEARN—this signifies the fast convergence rate of LIL. Third, the regret of LIL in Figure 2(a) (adversarial) and Figure 2(b) (stochastic) is about the same because LIL's coordination approach essentially adds stochasticity in the feedback observed by experts.

Further Related Work

Markovian, rested, and restless bandits. Our setting is similar in spirit to that of the *rested* Markovian bandits where each action/arm is associated with its own stochastic MDP and an arm changes its state only when it is pulled. In the seminal work, Gittins (1979) introduced the *Gittins index* to find an optimal sequential policy for these Markovian bandits problem. This work has been extended to settings where all arms change their reward distributions at every time step according to their associated stochastic MDPs, termed as *restless* bandits (Whittle 1988; Slivkins and Upfal 2008; Besbes, Gur, and Zeevi 2014). However, none of these frameworks could model learning dynamics of the experts in the adversarial setting we consider.

Online boosting and adaptive control. Another line of recent work similar in spirit to ours is online boosting—combining a set of "weak" online learning algorithms to form a "strong" online learning algorithm (Beygelzimer, Kale, and Luo 2015; Beygelzimer et al. 2015). However, there are substantial differences when compared to our problem setting: online boosting techniques have been studied in the context of classification/regression problems, (most of) the techniques are for the full-information setting, and one of the key challenges revolves around defining the weights for an online training example to be passed on to the "weak" learners. Our work is also similar to that of *switching adaptive control* (Fu 2015) which employs an array of simple candidate controllers; the goal is to design a meta-controller that can switch across controllers to search for the best candidate controller in real-time. Our result in Theorem 1 highlights the challenges in designing a meta-controller in an adversarial setting.

Learning in games. An orthogonal line of research studies the interaction of agents in multiplayer games where each agent uses a no-regret learning algorithm (Blum and Mansour 2007; Syrgkanis et al. 2015). The questions tackled in this line of research are very different as it focuses on interactions of the agents, their individual as well as social utilities, and the convergence of the game to an equilibrium. This orthogonal line of research reassures that the no-regret learning dynamics that we consider in this paper are indeed important and natural dynamics that are also prevalent in other application domains.

Conclusions and Future Work

In this paper, we investigated the framework of online learning using expert advice with bandit feedback when experts themselves are learning agents. Our hardness result highlights the fundamental challenges faced by traditional AI and machine learning methods when interacting with learning agents. In order to circumvent the hardness result, we introduced a new coordination approach allowing the forecaster to guide the experts' learning process by restricting the feedback received by them. In comparison to existing meta-algorithms that modify losses seen by the experts, our approach is more suitable for real-world applications where learning agents might have more complex feedback structure—for instance, a deal-aggregator site interacting with daily-deal marketplaces, or an AI system interacting with humans.

An important direction would be to study other practical ways of coordination and to understand the minimal communication required between the central algorithm and learning agents to achieve desired guarantees. For our problem setting, an interesting question to tackle is whether it is possible to design a forecaster using our coordination approach (which requires only 1-bit of communication at every time step) with a cumulative regret of $\Theta(T^{\frac{1}{2}})$ when the individual experts have no-regret learning dynamics with parameter $\beta = \frac{1}{2}$. Finally, we would like to point out that while we focused on the framework of online learning using expert advice, our results call for further studies of other frameworks and methods, *e.g.*, active learning methods when dealing with dynamic oracles.

Acknowledgments. This work was supported in part by the Swiss National Science Foundation, Nano-Tera.ch program as part of the Opensense II project, ERC StG 307036, and a Microsoft Research Faculty Fellowship. Adish Singla acknowledges support by a Facebook Graduate Fellowship.

References

Agarwal, A.; Luo, H.; Neyshabur, B.; and Schapire, R. E. 2017. Corralling a band of bandit algorithms. In *COLT*, 12–38.

Arora, R.; Dekel, O.; and Tewari, A. 2012. Online bandit learning against an adaptive adversary: from regret to policy regret. In *ICML*.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* 32(1):48–77.

Baram, Y.; El-Yaniv, R.; and Luz, K. 2004. Online choice of active learning algorithms. *Journal of Machine Learning Research*.

Bartók, G.; Foster, D. P.; Pál, D.; Rakhlin, A.; and Szepesvári, C. 2014. Partial monitoring – Classification, regret bounds, and algorithms. *Mathematics of Operations Research*.

Besbes, O.; Gur, Y.; and Zeevi, A. J. 2014. Optimal exploration-exploitation in a multi-armed-bandit problem with non-stationary rewards. In *NIPS*.

Beygelzimer, A.; Hazan, E.; Kale, S.; and Luo, H. 2015. Online gradient boosting. In *NIPS*.

Beygelzimer, A.; Kale, S.; and Luo, H. 2015. Optimal and adaptive algorithms for online boosting. In *ICML*, 2323–2331.

Blum, A., and Mansour, Y. 2007. Learning, regret minimization, and equilibria.

Bubeck, S., and Cesa-Bianchi, N. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning* 5(1):1–122.

Cesa-Bianchi, N., and Lugosi, G. 2006. *Prediction, learning, and games*. Cambridge university press.

Cesa-Bianchi, N.; Freund, Y.; Haussler, D.; Helmbold, D. P.; Schapire, R. E.; and Warmuth, M. K. 1997. How to use expert advice. *Journal of the ACM* 44(2):427–485.

Edelman, B.; Jaffe, S.; and Kominers, S. D. 2011. To groupon or not to groupon: The profitability of deep discounts. *Marketing Letters* 1–15.

Freund, Y., and Schapire, R. E. 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. In *COLT*, 23–37.

Fu, M. 2015. Switching adaptive control. In *Encyclopedia of Systems and Control*. Springer.

Gittins, J. C. 1979. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)* 148–177.

Hirnschall, C.; Singla, A.; Tschiatschek, S.; and Krause, A. 2018. Learning user preferences to incentivize exploration in the sharing economy. In *AAAI*.

Hsu, W.-N., and Lin, H.-T. 2015. Active learning by learning. In *AAAI*, 2659–2665.

Littlestone, N., and Warmuth, M. K. 1994. The weighted majority algorithm. *Info and Computation* 70(2):212–261.

Maillard, O.-A., and Munos, R. 2011. Adaptive bandits: Towards the best history-dependent strategy. In *AISTATS*.

McMahan, H. B., and Streeter, M. J. 2009. Tighter bounds for multi-armed bandits with expert advice. In *COLT*.

Shalev-Shwartz, S. 2011. Online learning and online convex optimization. *Foundations and Trends in Machine Learning* 4(2):107–194.

Singla, A.; Hassani, H.; and Krause, A. 2018. Learning to interact with learning agents (extended version).

Singla, A.; Tschiatschek, S.; and Krause, A. 2016. Actively learning hemimetrics with applications to eliciting user preferences. In *ICML*.

Slivkins, A., and Upfal, E. 2008. Adapting to a changing environment: the brownian restless bandits. In *COLT*.

Syrgkanis, V.; Agarwal, A.; Luo, H.; and Schapire, R. E. 2015. Fast convergence of regularized learning in games. In *NIPS*, 2971–2979.

Whittle, P. 1988. Restless bandits: Activity allocation in a changing world. *Journal of applied probability*.

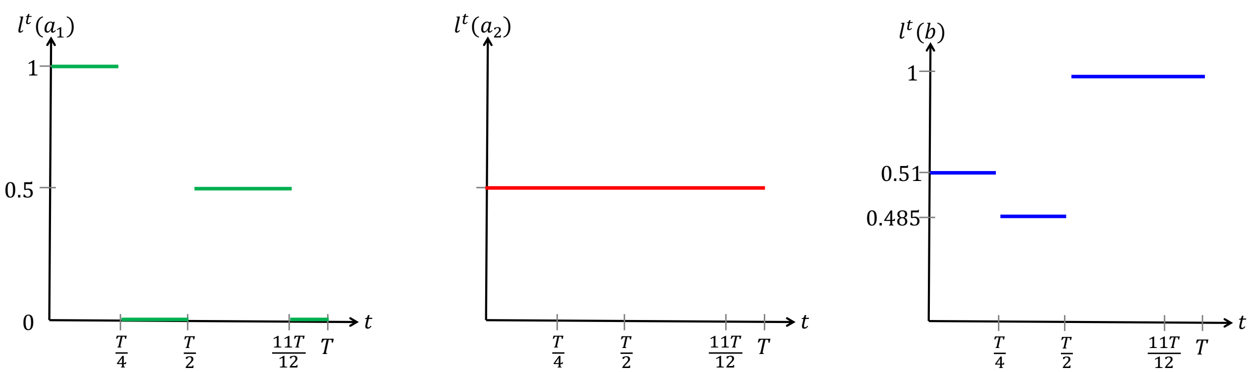
Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*.

Proof of Theorem 1

In this section, we give a proof of the hardness result by discussing a simple setting in which any forecaster would suffer a linear regret. We prove this hardness result when playing against an oblivious (non-adaptive) adversary and when restricting the experts to be implementing well-behaved learning algorithms (*e.g.*, the HEDGE algorithm (Freund and Schapire 1995)).

The setting. Our setting consists of two experts EXP_1 and EXP_2. The first expert EXP_1 has two actions given by $\mathcal{A}_1 = \{a_1, a_2\}$, and the second expert EXP_2 has only one action given by $\mathcal{A}_2 = \{b\}$. The action set of the forecaster ALGO is given by $\mathcal{A} = \{a_1, a_2, b\}$. The expert EXP_1 plays the HEDGE algorithm (Freund and Schapire 1995), *i.e.*, the regret rate parameter is $\beta_1 = 0.5$ (see Equation 1); the expert EXP_2 has only one action to play, hence $\beta_2 = 0$ (see Equation 1). We assume that the forecaster knows these parameters β_1 and β_2.[6]

Loss sequences. Figures 1(a), 1(b), and 1(c) show the *cumulative* loss sequences L_1, L_2, and L_3 for three different scenarios—the adversary at $t = 0$ uniformly at random picks one of these scenarios and uses that loss sequence. Our main argument is that for any forecaster, one of the scenarios leads to linear regret. In the proof below, we would consider the case where a forecaster is facing the first scenario with cumulative loss sequence L_1 and discuss the execution behavior. For the cumulative loss sequence L_1 shown in Figure 1(a), we show in Figure 3 the instantaneous losses of the actions $\mathcal{A} = \{a_1, a_2, b\}$.



(a) EXP_1: Action a_1 (L_1) (b) EXP_1: Action a_2 (L_1) (c) EXP_2: Action b (L_1)

Figure 3: For the first scenario with the cumulative loss sequence L_1 shown in Figure 1(a), here we show the instantaneous losses of the individual actions. 3(a) and 3(b) show the losses of actions for EXP_1; 3(c) shows the losses of action for EXP_2.

Model specification. To fully specify the model and Protocol 1, we describe now the feedback vector over time. The experts observe the following feedback when selected: the expert EXP_1 would observe the losses $\{l^t(a_1), l^t(a_2)\}$ when $i^t = 1$; and the expert EXP_2 would observe the loss $l^t(b)$ when $i^t = 2$.

Execution behavior on L_1. We now consider the case where a forecaster is facing the sequence L_1. We divide the time horizon T into different slots, and discuss the execution behavior of the forecaster and experts over these time slots. For a clarity of presentation, we shall use $\Delta = 0.01$ as a constant in rest of the proof below.

- $t \in [0, \frac{T}{4}]$: In this slot, the expert EXP_1 would be selected most of the times by the forecaster, and the number of times EXP_2 would be selected is $o(T)$. If this is not the case, then this forecaster would suffer linear regret on the loss sequence L_3 for the third scenario in Figure 1(c).[7] Note that the loss incurred by the forecaster at any time t in this slot is at least 0.5.
- $t \in (\frac{T}{4}, \frac{T}{2}]$: This is the first crucial slot whereby the forecaster's selection strategy would add "blind spots" in the feedback history of the expert EXP_1. The key argument is that in this slot, the forecaster cannot select expert EXP_1 for more than $\frac{T}{6} + o(T)$ time steps—if this happens, then this forecaster would have a linear regret on the loss sequence L_2 for the second scenario in Figure 1(b). In other words, in this slot, the expert EXP_1 has missed observing the feedback for $\frac{T}{12} - o(T)$ time steps. Note that the loss incurred by the forecaster at any time t in this time slot is at least $0.5 - \frac{3\Delta}{2}$.
- $t \in (\frac{T}{2}, \frac{11T}{12}]$: In this slot, the expert EXP_1 would be selected most of the times by the forecaster, and the number of times EXP_2 would be selected is $o(T)$. If this is not the case, then this forecaster would suffer a linear regret on the loss sequence L_1 as the loss associated with EXP_2 is $l^t(b) = 1$ in this slot. Note that the loss incurred by the forecaster at any time t in this time slot is at least 0.5.

[6]In fact, this hardness result holds even when considering a powerful forecaster which knows exactly the learning algorithms used by the experts, and is able to see the losses $\{l^t(a_1), l^t(a_2), l^t(b)\}$ at every time $t \in [T]$.

[7]Note here that the loss sequence L_3 is exactly equal to L_1 up to time $\frac{T}{4}$. Furthermore, although we have assumed that the setting chosen by the adversary is L_1, we should bear in mind that the forecaster (who can not distinguish between the losses at least up to time $\frac{T}{4}$) should play in a way that it does not suffer a linear regret on the loss sequence L_3.

- $t \in (\frac{11T}{12}, T]$: This is the second crucial slot which would lead to a linear regret for the forecaster. We consider that the forecaster still does the "right" thing in this slot, *i.e.*, the expert EXP_1 would be selected most of the times, and the number of times EXP_2 would be selected is $o(T)$. However, the expert EXP_1 has missed observing feedback for $\frac{T}{12} - o(T)$ time steps in the slot $(\frac{T}{4}, \frac{T}{2}]$. Note also that no coordination is permitted between the forecaster and the experts, and hence, EXP_1 is not aware of the time steps that it misses the feedback. As a result, at the start of this slot, the cumulative loss of the action a_1 (as perceived by EXP_1 based on observed feedback history) is at least $0.5 \cdot \frac{T}{12}$ more than the cumulative loss of the action a_2 (as perceived by EXP_1 based on observed feedback history). The expert EXP_1 who is playing HEDGE algorithm in our setting would recommend action a_2 most of the times, and a_1 would only be recommended $o(T)$ number of times.

Linear regret. Let us now compute the cumulative regret of the forecaster when experiencing loss sequence L_1 as discussed above. As per Equation 3, we want to be competitive with the action a_1 of the expert EXP_1. Based on Figure 3(a), the cumulative loss of the action a_1 is given by:

$$\sum_{t \in [T]} l^t(a_1) = 1 \cdot \frac{T}{4} + 0.5 \cdot (\frac{11T}{12} - \frac{T}{2}) = T \cdot (\frac{1}{2} - \frac{1}{24})$$

The cumulative loss of the forecaster as per the execution behavior discussed above can be lower bounded as follows:

$$\begin{aligned}\sum_{t \in [T]} l^t(a^t_{i^t}) &\geq 0.5 \cdot \frac{T}{4} + \big(0.5 - \frac{3\Delta}{2}\big) \cdot \frac{T}{4} + 0.5 \cdot (\frac{11T}{12} - \frac{T}{2}) + 0.5 \cdot \big(\frac{T}{12} - o(T)\big) \\ &= T \cdot (\frac{1}{2} - \frac{3\Delta}{8} - \frac{o(T)}{2 \cdot T})\end{aligned}$$

Hence, the cumulative regret of the forecaster is lower bounded by:

$$\text{REG}(\text{ALGO}, T) \geq T \cdot (\frac{1}{2} - \frac{3\Delta}{8} - \frac{o(T)}{2 \cdot T}) - T \cdot (\frac{1}{2} - \frac{1}{24}) = T \cdot (\frac{1}{24} - \frac{3\Delta}{8} - \frac{o(T)}{2 \cdot T})$$

Recall that the constant $\Delta = 0.01$, hence the cumulative regret of the forecaster is lower bounded by $\lim_{T \mapsto \infty} \text{REG}(\text{ALGO}, T) \geq \frac{91T}{2400}$. As we discussed above, any forecaster which doesn't have the above-mentioned execution behavior in the slot $t \in [0, \frac{T}{4}]$ or $t \in (\frac{T}{4}, \frac{T}{2}]$ would also suffer a linear regret on L_3 or L_2 loss sequences.

Proof of Proposition 1

In this section, we prove Proposition 1: (i) **OCP Algorithms** provides a proof for algorithms based on Online Convex Programming via greedy projections (Zinkevich 2003), and (ii) **OMD Algorithms** provides a proof for the Online Mirror Descent family of algorithms (Shalev-Shwartz 2011).

OCP Algorithms

Assume that an expert EXP_j is implementing an algorithm based on Online Convex Programming (OCP) via greedy projections. We will show that such an algorithm has smooth learning dynamics. Note that OCP algorithms have a cumulative regret of $O(\sqrt{T})$ (*i.e.*, $\beta_j = \frac{1}{2}$). Consider the α-OCP algorithm that proceeds according to Algorithm 3. Proving smooth learning dynamics for OCP is equivalent to showing that α-OCP suffers a cumulative regret of $O(\sqrt{\frac{T}{\alpha}})$. More precisely, we have the following lemma.

Lemma 1. *Let $\|S\|$ denote the diameter of the convex set S and L denotes an upper bound on the magnitude of the gradient at any time $t \in [T]$. Then, the expected regret of the α-OCP algorithm is given by*

$$\mathbb{E}\Big[\sum_{t=1}^{T} \big(f^t(w^t) - f^t(u)\big)\Big] \leq \frac{\|S\|^2}{2} \cdot \sqrt{\frac{T}{\alpha}} + L^2 \cdot \sqrt{\frac{T}{\alpha}} \tag{5}$$

where the expectation is w.r.t. the sequence of Bernoulli random variables B^t for $t \in [T]$.

Proof. We can equivalently write the updates in the α-OCP algorithm as follows:

$$\begin{aligned}w^{t+\frac{1}{2}} &= w^t - \eta^t B^t z^t \\ &= w^t - (\eta^t \cdot \alpha) \cdot \Big(\frac{B^t}{\alpha}\Big) z^t \\ &= w^t - \tilde{\eta}^t \cdot \tilde{z}^t\end{aligned}$$

Algorithm 3: α-OCP

Problem setting: Convex set S; sequence of convex loss functions $f^t : S \to \mathbb{R}_+$ for $t \in [T]$

Parameters: Learning rates η^t for $t \in [T]$

Initialize: $w_0 \in S$ arbitrarily

foreach $t = 1, 2, \ldots, T$ **do**

$w^{t+\frac{1}{2}} = w^t - \eta^t B^t z^t$ where

- $z^t \in \partial f^t(w^t)$,
- B^t is an independent Bernoulli random variable with parameter α (*i.e.*, $\Pr(B^t = 1) = 1 - \Pr(B^t = 0) = \alpha$), and
- $\eta_t = 1/\sqrt{1 + \sum_{\tau=1}^t B^\tau}$

$w^{t+1} = \mathrm{Proj}_S(w^{t+\frac{1}{2}})$

where $\tilde{\eta}^t = (\eta^t \cdot \alpha)$ and $\tilde{z}^t = (\frac{B^t}{\alpha})z^t$. Note that $\mathbb{E}\Big[\tilde{z}^t \,\Big|w^1 \ldots w^t\Big] = \mathbb{E}\Big[z^t\Big] \in \partial f^t(w^t)$. We have:

$$
\begin{aligned}
&\mathbb{E}\Big[\sum_{t=1}^T \big(f^t(w^t) - f^t(u)\big)\Big] \\
&= \mathbb{E}\Big[\sum_{t=1}^T \mathbb{E}\Big[\big(f^t(w^t) - f^t(u)\big) \,\Big|w^1 \ldots w^t\Big]\Big] \\
&\leq \mathbb{E}\Big[\sum_{t=1}^T \mathbb{E}\Big[< \partial f^t(w^t), w^t - u > \,\Big|w^1 \ldots w^t\Big]\Big] \\
&= \mathbb{E}\Big[\sum_{t=1}^T \mathbb{E}\Big[< \tilde{z}^t, w^t - u > \,\Big|w^1 \ldots w^t\Big]\Big] \\
&= \mathbb{E}\Big[\sum_{t=1}^T \frac{1}{2\alpha\eta^t}\mathbb{E}\Big[\left\|w^t - w\right\|^2 - \left\|w^{t+\frac{1}{2}} - w\right\|^2 + (\alpha\eta^t)^2 \left\|\tilde{z}^t\right\|^2 \,\Big|w^1 \ldots w^t\Big]\Big] \\
&\leq \mathbb{E}\Big[\sum_{t=1}^T \frac{1}{2\alpha\eta^t}\mathbb{E}\Big[\left\|w^t - w\right\|^2 - \left\|w^{t+1} - w\right\|^2 + (\alpha\eta^t)^2 \left\|\tilde{z}^t\right\|^2 \,\Big|w^1 \ldots w^t\Big]\Big] \\
&= \mathbb{E}\Big[\sum_{t=1}^T \frac{\left\|w^t - w\right\|^2}{2\alpha\eta^t} - \frac{\left\|w^{t+1} - w\right\|^2}{2\alpha\eta^t}\Big] + \frac{\alpha}{2}\mathbb{E}\Big[\sum_{t=1}^T \eta^t \left\|\tilde{z}^t\right\|^2\Big] \\
&\leq \mathbb{E}\Big[\frac{\left\|w^1 - w\right\|^2}{2\alpha\eta^1} - \frac{\left\|w^{T+1} - w\right\|^2}{2\alpha\eta^T} + \frac{1}{2}\sum_{t=2}^T \left\|w^t - w\right\|^2 \Big(\frac{1}{\alpha\eta^t} - \frac{1}{\alpha\eta^{t-1}}\Big)\Big] + \frac{\alpha}{2}\mathbb{E}\Big[\sum_{t=1}^T \eta^t \left\|\tilde{z}^t\right\|^2\Big] \\
&\leq \left\|S\right\|^2 \mathbb{E}\Big[\frac{1}{2\alpha\eta^T}\Big] + \frac{\alpha}{2}\mathbb{E}\Big[\sum_{t=1}^T \eta^t \left\|\tilde{z}^t\right\|^2\Big]
\end{aligned}
$$

Now note that $\mathbb{E}\Big[\eta^t \left\|\tilde{z}^t\right\|^2\Big] = \mathbb{E}\Big[\mathbb{E}\Big[\eta^t \left\|\tilde{z}^t\right\|^2 \,\Big|w^1 \ldots w^t\Big]\Big] = \mathbb{E}\Big[\frac{1}{\alpha}\eta^t \left\|z^t\right\|^2\Big] \leq \frac{1}{\alpha}L^2\mathbb{E}\Big[\eta^t\Big]$. We thus obtain

$$
\mathbb{E}\Big[\sum_{t=1}^T \big(f^t(w^t) - f^t(u)\big)\Big] \leq \left\|S\right\|^2 \mathbb{E}\Big[\frac{1}{2\alpha\eta^T}\Big] + \frac{L^2}{2}\mathbb{E}\Big[\sum_{t=1}^T \eta^t\Big].
$$

We next recall that $\eta^t = \frac{1}{\sqrt{1+\sum_{\tau=1}^t B^\tau}}$. By using the multiplicative Chernoff bound (as B^τ's are Bernoulli random variables), we obtain

$$
\Pr\big(\eta^t \geq \sqrt{\frac{2}{t\alpha}}\big) = \Pr\big(\sum_{\tau=1}^t B^\tau \leq \frac{t\alpha}{2}\big) \leq \exp(-\frac{t\alpha}{12}).
$$

Hence, we obtain $\mathbb{E}\Big[\eta_t\Big] \leq \sqrt{\frac{2}{t\alpha}} + \exp(-\frac{t\alpha}{12})$. Also, due to concavity of the function $h(x) = \sqrt{x}$, we have that $\mathbb{E}\Big[\frac{1}{\eta^T}\Big] \leq \sqrt{T\alpha}$.

We finally obtain

$$\mathbb{E}\Big[\sum_{t=1}^{T}\big(f^t(w^t)-f^t(u)\big)\Big] \le \|S\|^2\sqrt{\frac{T}{\alpha}} + L^2\sqrt{\frac{T}{\alpha}} + \sum_{t=1}^{T}\exp{(-\frac{t\alpha}{12})}$$
$$\le \|S\|^2\sqrt{\frac{T}{\alpha}} + L^2\sqrt{\frac{T}{\alpha}} + \frac{1}{(1-\exp{(-\frac{\alpha}{12})})}$$
$$\le \|S\|^2\sqrt{\frac{T}{\alpha}} + L^2\sqrt{\frac{T}{\alpha}} + \frac{24}{\alpha},$$

where the last line is because $\frac{1}{(1-\exp(-\frac{\alpha}{12}))} \le \frac{24}{\alpha}$ for $\alpha \le 1$. $\square$

OMD Algorithms

We now consider the case that an expert $\textsc{Exp}_j$ is implementing an algorithm from the Online Mirror Descent (OMD) family of algorithms. We assume that the algorithm has a cumulative regret of $O(\sqrt{T})$ (*i.e.*, $\beta_j = \frac{1}{2}$). We also assume that the algorithm uses the doubling trick and the regret of $O(\sqrt{T})$ holds for any time horizon T. The proof proceeds in three steps.

Step 1. α-OMD with a fixed time horizon

We first analyze the algorithm α-OMD that proceeds according to Algorithm 4 which is run for a fixed (deterministic) number of time steps T.

Algorithm 4: α-OMD

Problem setting: Convex set S; sequence of convex loss functions $f^t : S \to \mathbb{R}_+$ for $t \in [T]$
Parameters: a link function $g : \mathbb{R}^d \to S$; time horizon T
Initialize: time $t = 1$, auxiliary variable $\theta^t = \mathbf{0} \in \mathbb{R}^d$
foreach $t = 1, 2, \ldots, T$ **do**
  Predict vector $w^t = g(\theta^t)$
  Update $\theta^{t+1} = \theta^t - B^t z^t$ where
  - $z^t \in \partial f^t(w^t)$, and
  - B^t is an independent Bernoulli random variable with parameter α (*i.e.*, $\Pr(B^t = 1) = 1 - \Pr(B^t = 0) = \alpha$)

Lemma 2. *Let R be a $1/\eta$- strongly convex function over S with respect to a norm $\|\cdot\|$. Assume that α-OMD is run on the sequence of loss functions denoted as f^t for $t \in [T]$ with a link function*

$$g(\theta) = \arg\max_{w\in S}(\langle w, \theta\rangle - R(w))$$

Furthermore, assume that $\forall t \in [T], f^t$ is L-Lipshitz with respect to norm $\|\cdot\|$. Then

$$\mathbb{E}\Big[\sum_{t=1}^{T}\big(f^t(w^t)-f^t(u)\big)\Big] \le \frac{R(u)}{\alpha} + \eta T L^2. \tag{6}$$

Proof. For the sake of analysis, we introduce the following slightly modified algorithm:

1. Initialize $\tilde{\theta}^1 = \frac{\theta^1}{\alpha}$.
2. At time $t = 1, 2 \cdots T$, let $\tilde{w}^t = \tilde{g}(\tilde{\theta}^t)$ and $\tilde{\theta}^{t+1} = \tilde{\theta}^t - \tilde{z}^t$. Here, we have $\tilde{z}^t = \frac{B^t}{\alpha}z^t$, and the function $\tilde{g}$ is defined as $\tilde{g}(\theta) = \arg\max_{w\in S}(\langle w, \theta\rangle - \frac{R(w)}{\alpha})$.

It is straight forward to justify for any $t \in [T]$ that $\tilde{\theta}^t = \frac{\theta^t}{\alpha}$ and $\tilde{w}^t = w^t$. Also note that $\mathbb{E}\Big[\tilde{z}^t \,\Big|\, \tilde{z}^1 \ldots \tilde{z}^{t-1}\Big] = z^t \in \partial f^t(w^t)$. Hence, the modified algorithm using $(\tilde{\theta}^t, \tilde{w}^t)$ is precisely a stochastic OMD algorithm with the link function $\tilde{g}$. Next, we apply Theorem 4.1 from (Shalev-Shwartz 2011)—we have $\tilde{w}^t = w^t$ and the fact that $\frac{R(\cdot)}{\alpha}$ is a $1/(\eta\alpha)$-strongly convex function, we obtain

$$\mathbb{E}\Big[\sum_{t=1}^{T}\big(f^t(w^t)-f^t(u)\big)\Big] \le \sup_{u\in S}\frac{R(u)}{\alpha} + \eta\alpha\sum_{t=1}^{T}\mathbb{E}\Big[\big\|\tilde{z}^t\big\|^2\Big]$$

We finally note that

$$\mathbb{E}\Big[\left\|\tilde{z}^t\right\|^2\Big] = \mathbb{E}\Big[\mathbb{E}\Big[\left\|\tilde{z}^t\right\|^2 \ \Big|\tilde{z}^1 \dots \tilde{z}^{t-1}\Big]\Big] \leq \frac{L^2}{\alpha}$$

The result of the lemma is now immediate. □

From Lemma 2, if we set $\eta = \Theta(\frac{1}{\sqrt{\alpha T}})$, the algorithm α-OMD suffers a cumulative regret of $O(\sqrt{\frac{T}{\alpha}})$ for any fixed time horizon T. However, it is important to note that the algorithm doesn't know the value αT and hence the results from Lemma 2 are not directly applicable.

Step 2. α-OMD with a random time horizon

We are assuming that the algorithm used by the expert $\textsc{Exp}_j$ is performing the doubling trick, *i.e.*, it runs in blocks whose size get doubled consecutively and within each block the learning rate is fixed. As a result, after the algorithm receives sufficient number of feedback instances to finish a block, it restarts the algorithm and changes the learning rate for the next block (which has twice the size). The lemma below bounds the regret of the α-OMD when running for a block of size M, *i.e.*, the algorithm needs to be given M feedback instances (from the forecaster) until it switches to the next block. In order to analyze the regret suffered in this block, we need to consider a slightly different version of α-OMD which stops after a randomly chosen time horizon.

Lemma 3 (α-OMD with a random time horizon)**.** *Assume that we run the α-OMD algorithm until the time, call it T_{stop}, such that following stopping criterion has been fulfilled:*

$$\sum_{t=1}^{T_{\text{stop}}} B^t = M. \tag{7}$$

We then have

$$\mathbb{E}\Big[\sum_{t=1}^{T_{\text{stop}}} \big(f^t(w^t) - f^t(u)\big)\Big] \leq \frac{R(u)}{\alpha} + \frac{\eta M L^2}{\alpha} + 14 \cdot L \left\|S\right\| \sqrt{\frac{M}{\alpha^2}}, \tag{8}$$

where $\|S\|$ denotes the diameter of the convex set S and L denotes an upper bound on the Lipshitz parameter of all the functions f^t.

Proof. We can write

$$\begin{aligned}
&\mathbb{E}\Big[\sum_{t=1}^{T_{\text{stop}}} \big(f^t(w^t) - f^t(u)\big)\Big] \\
&= \mathbb{E}\Big[\sum_{t=1}^{M/\alpha} \big(f^t(w^t) - f^t(u)\big)\Big] - \Big(\mathbb{E}\Big[\sum_{t=1}^{M/\alpha} \big(f^t(w^t) - f^t(u)\big)\Big] - \mathbb{E}\Big[\sum_{t=1}^{T_{\text{stop}}} \big(f^t(w^t) - f^t(u)\big)\Big]\Big) \\
&\leq \mathbb{E}\Big[\sum_{t=1}^{M/\alpha} \big(f^t(w^t) - f^t(u)\big)\Big] + \Big(L \left\|S\right\| \cdot \mathbb{E}\Big[\big|T_{\text{stop}} - M/\alpha\big|\Big]\Big),
\end{aligned}$$

where the last step follows from the fact that for any two vectors $u, v \in S$, we have $|f(u) - f(v)| \leq L \|S\|$. The first term above can be bounded using Lemma 2.

We thus need to upper-bound the expected value of $\big|T_{\text{stop}} - M/\alpha\big|$. As B^t's are Bernoulli(α) random variables, we expect that T_{stop} concentrates around M/α. By using the multiplicative Chernoff bound, we have

$$\begin{aligned}
\Pr\Big(T_{\text{stop}} \geq M/\alpha + \beta\Big) &= \Pr\Big(\sum_{\tau=1}^{M/\alpha+\beta} B^\tau \leq M\Big) \\
&\leq \Pr\Big(\sum_{\tau=1}^{M/\alpha+\beta} B^\tau \leq (M + \alpha\beta)(1 - \frac{\alpha\beta}{M + \alpha\beta})\Big) \\
&\leq \exp(-\frac{(\alpha\beta)^2}{3(M + \alpha\beta)}).
\end{aligned}$$

Similarly, we can show that

$$\Pr\Big(T_{\text{stop}} \leq M/\alpha - \beta\Big) \leq \exp(-\frac{(\alpha\beta)^2}{3(M - \alpha\beta)}).$$

We thus obtain

$$\begin{aligned}
\mathbb{E}\Big[\Big|T_{\text{stop}} - M/\alpha\Big|\Big] &\leq \sum_{l=0}^{M/\alpha} \mathbf{Pr}\Big(T_{\text{stop}} \leq M/\alpha - l\Big) + \sum_{l=0}^{\infty} \mathbf{Pr}\Big(T_{\text{stop}} \geq M/\alpha + l\Big) \\
&\leq \sum_{l=0}^{M/\alpha} \exp(-\frac{(\alpha l)^2}{3(M-\alpha l)}) + \sum_{l=0}^{\infty} \exp(-\frac{(\alpha l)^2}{3(M+\alpha l)}) \\
&\leq 2\sum_{l=0}^{\infty} \exp(-\frac{(\alpha l)^2}{3(M+\alpha l)}) \\
&\leq 2\sum_{l=0}^{\infty} \sqrt{M/\alpha^2} \exp(-\frac{Ml^2}{3(M+l\sqrt{M})}) \\
&\leq 2\sqrt{M/\alpha^2} \sum_{l=0}^{\infty} \exp(-\frac{l}{6}) \\
&\leq 14\sqrt{M/\alpha^2}.
\end{aligned}$$

□

From Lemma 3, if we set $\eta = \Theta(\frac{1}{\sqrt{M}})$, the algorithm α-OMD suffers a cumulative regret of $O(\sqrt{\frac{M}{\alpha^2}})$ in a block of size M. We will use this bound to finish the proof.

Step 3. Putting it together

When the algorithm run by the expert $\textsc{Exp}_j$ is using the doubling trick, the regret suffered in a block of size M is upper-bounded as $O\big(\sqrt{\frac{M}{\alpha^2}}\big)$ (from Lemma 3). Now, assume that T is the total number of time steps for the forecaster. Then, the OMD algorithm (which is given feedback with probability α and plays according to the doubling trick) will be given feedback in a total of αT time steps (on average). Hence, αT is the effective time that needs to be split up into blocks and then we can apply the regret bounds from Lemma 3 to each of these blocks. As a result, it is not hard to see that the cumulative regret (after summing up over all the blocks played by the algorithm and using Jensen inequality) is bounded as $O\big(\sqrt{\frac{T}{\alpha}}\big)$.

Proof of Theorem 2

In this section, we provide a proof of Theorem 2, following a step by step approach.

Step 1: Bounds on the external regret of the forecaster LIL

We begin by noting that the selection strategy of the forecaster LIL is similar to the EXP3 algorithm (Auer et al. 2002), ignoring the fact that the experts are learning agents. It is important to note that the EXP3 algorithm in fact makes no assumptions about the behavior of arms/experts, and hence we can apply the bounds on the *external* regret of the EXP3 algorithm to our setting (*cf.* (McMahan and Streeter 2009) for a discussion of applying the EXP3 algorithm to the problem of online learning using experts advice with bandit feedback). By directly applying the results of the EXP3 algorithm, *cf.* Theorem 3.1 from (Auer et al. 2002), we can state the following guarantees on the *external* regret of the forecaster LIL when competing with an expert $\textsc{Exp}_k$ where $k \in [N]$ as:

$$\sum_{t=1}^{T} \mathbb{E}\Big[l^t\Big(\pi_{i^t}(\mathcal{H}^t_{i^t,\text{LIL}})\Big)\Big] - \mathbb{E}\Big[\sum_{t=1}^{T} l^t\Big(\pi_k(\mathcal{H}^t_{k,\text{LIL}})\Big)\Big] \leq c \cdot \eta \cdot T + \frac{(\log N)\cdot N}{\eta} \tag{9}$$

where c is a constant given by $c = e - 1$ and the expectation is w.r.t. the randomization of the forecaster as well as any internal randomization of the experts.

Step 2: Smooth no-regret learning dynamics of the experts

We note that during the execution of the forecaster LIL in Algorithm 2, the probability that *any* expert $\textsc{Exp}_j$ $\forall j \in [N]$ observes feedback is constant over time $t \in [T]$ and is given by $\alpha = \frac{\eta}{N}$. Hence, by definition, we have $\mathcal{H}^t_{j,\text{LIL}} \equiv \mathcal{H}^t_{j,\alpha\text{-}\textsc{Full}}$ $\forall j \in [N]$ with $\alpha = \frac{\eta}{N}$. By applying the smooth no-regret learning dynamics of the expert $\textsc{Exp}_k$ from Equation 4 guarantees the following:

$$\mathbb{E}\Big[\sum_{t=1}^{T} l^t\Big(\pi_k(\mathcal{H}^t_{k,\text{LIL}})\Big)\Big] - \min_{a\in\mathcal{A}_k} \sum_{t=1}^{T} l^t(a) \leq \mathcal{O}\Big(\frac{(\alpha\cdot T)^{\beta_k}}{\alpha}\Big)$$

$$= \mathcal{O}\Big(\frac{T^{\beta_k} \cdot N^{1-\beta_k}}{\eta^{1-\beta_k}}\Big), \tag{10}$$

where β_k is the regret rate parameter of the expert $\textsc{Exp}_k$.

Step 3: Putting it together

Combining Equations 9 and 10 from above, we get the following bounds on the cumulative loss of the forecaster LIL:

$$\sum_{t=1}^{T} \mathbb{E}\Big[l^t\Big(\pi_{i^t}(\mathcal{H}^t_{i^t,\mathrm{LIL}})\Big)\Big] - \min_{a \in \mathcal{A}_k} \sum_{t=1}^{T} l^t(a) \le \mathcal{O}\Big(\eta \cdot T + \frac{(\log N) \cdot N}{\eta} + \frac{T^{\beta_k} \cdot N^{1-\beta_k}}{\eta^{1-\beta_k}}\Big) \tag{11}$$

Note that the above bound holds for all $k \in [N]$. Hence, let us set $k = j^*$ where j^* corresponds to the best expert $\textsc{Exp}_{j^*}$ that we want to be competitive with in Equation 3. As per assumptions of the theorem, this best expert $\textsc{Exp}_{j^*}$ indeed has the smooth no-regret learning dynamics with parameter $\beta_{j^*} \in [0, 1]$. Stating the above bound in terms of $k = j^*$ and using the definition of the regret $\textsc{Reg}(T, \mathrm{LIL})$ from Equation 3, we can write down the regret of the forecaster LIL as follows:

$$\textsc{Reg}(T, \mathrm{LIL}) \le \mathcal{O}\Big(\eta \cdot T + \frac{(\log N) \cdot N}{\eta} + \frac{T^{\beta_{j^*}} \cdot N^{1-\beta_{j^*}}}{\eta^{1-\beta_{j^*}}}\Big) \tag{12}$$

Step 4: Optimizing η

Next, we will optimize the value of η in terms of T and N. However, note that the forecaster doesn't know β_{j^*} and hence cannot directly optimize the value of η. As per the theorem statement, the forecaster LIL is invoked with input $\beta \in [0, 1]$ such that $\beta \ge \beta_{j^*}$.

Step 4.1: Optimizing η for known β_{j^*}, *i.e.*, $\beta = \beta_{j^*}$

To begin with, let us first optimize η for case when $\beta = \beta_{j^*}$. In order to find the optimal dependency of η on T, we set $\eta \sim T^{-z}$, and we will then find the optimal value of z. By this choice of η, the following terms appear in Equation 12 when stated as the power of T :

$$\big\{T^{1-z}, T^{z}, T^{z+\beta_{j^*}\cdot(1-z)}\big\} \tag{13}$$

Solving for optimal value of z to minimize the power of T in the leading term, we get $z = \frac{1-\beta_{j^*}}{2-\beta_{j^*}}$.

Next, we find the optimal dependency of η on N. Note that, when $\beta = 0$, we have optimal dependency of η on N as $(N \cdot \log(N))^{\frac{1}{2}}$. In general, the optimal dependency of η on N can be found by setting $\eta \sim N^{z}$, which gives us from Equation 12 the following terms stated as the power of N (after keeping only the leading terms w.r.t. T):

$$\big\{N^{z}, N^{(1-\beta_{j^*})\cdot(1-z)}\big\} \tag{14}$$

Solving for optimal value of z to minimize the power of N, we get $z = \frac{1-\beta_{j^*}}{2-\beta_{j^*}}$.

For any $\beta_{j^*} \in [0, 1]$, we can thus write the optimal value of η as:

$$\eta = T^{-\frac{1-\beta_{j^*}}{2-\beta_{j^*}}} \cdot N^{\frac{1-\beta_{j^*}}{2-\beta_{j^*}}} \cdot (\log N)^{\left(\frac{1}{2}\cdot \mathbf{1}_{\{\beta_{j^*}=0\}}\right)} \tag{15}$$

By keeping only the leading terms of T, we can write the regret as follows:

$$\textsc{Reg}(T, \mathrm{LIL}) \le \mathcal{O}\big(T^{\frac{1}{2-\beta_{j^*}}} \cdot N^{\frac{1}{2-\beta_{j^*}}} \cdot (\log N)^{\left(\frac{1}{2}\cdot \mathbf{1}_{\{\beta_{j^*}=0\}}\right)}\big) \tag{16}$$

Step 4.2: Optimizing η for unknown β_{j^*}, *i.e.*, $\beta \ge \beta_{j^*}$

When β_{j^*} is not known exactly, and β only upper bounds β_{j^*}, we can still optimize η w.r.t. β to get the same η as stated above, replacing β_{j^*} by β_j (note that $1/(2-\beta)$ is increasing in β). By keeping only the leading terms of T, we can write the regret as follows:

$$\textsc{Reg}(T, \mathrm{LIL}) \le \mathcal{O}\big(T^{\frac{1}{2-\beta}} \cdot N^{\frac{1}{2-\beta}} \cdot (\log N)^{\left(\frac{1}{2}\cdot \mathbf{1}_{\{\beta=0\}}\right)}\big) \tag{17}$$

This gives us the desired bound stated in Theorem 2.