Online Learning with Stochastically Partitioning Experts

Puranjay Datta[1] **Sharayu Moharir**[1] **Jaya Prakash Champati**[2]

[1]Electrical Engineering Department, IIT Bombay, India
[2]Computer Science Department, University of Victoria, Canada

Abstract

We study a variant of the experts problem in which new experts are revealed over time according to a stochastic process. The experts are represented by partitions of a hypercube $\mathbb{B}$ in d-dimensional Euclidean space. In each round, a point is drawn from $\mathbb{B}$ in an independent and identically distributed manner using an unknown distribution. For each chosen point, we draw d orthogonal hyperplanes parallel to the d faces of $\mathbb{B}$ passing through the point. The set of experts available in a round is the set of partitions of $\mathbb{B}$ created by all the hyperplanes drawn up to that point. Losses are adversarial, and the performance metrics of interest include expected regret and high probability bounds on the sample-path regret. We propose a suitably adapted version of the Hedge algorithm called Hedge-G, which uses a constant learning rate and has $O(\sqrt{2^d T \log T})$ expected regret, which is order-optimal. Further, we show that for Hedge-G, there exists a trade-off between choosing a learning rate that has optimal expected regret and a learning rate that leads to a high probability sample-path regret bound. We address this limitation by proposing AdaHedge-G, a variant of Hedge-G that uses an adaptive learning rate by tracking the loss of the experts revealed up to that round. AdaHedge-G simultaneously achieves $O(\log(\log T)\sqrt{T \log T})$ expected regret and $O(\log T \sqrt{T \log T})$ sample-path regret, with probability at least $1 - T^{-c}$, where $c > 0$ is a constant dependent on d.

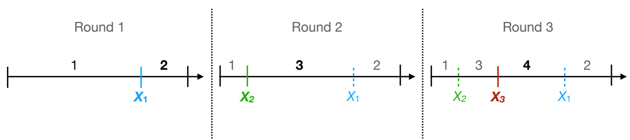
1 INTRODUCTION

In the standard decision-theoretic online learning studied by Freund and Schapire [1997], there are N experts (or actions) at the disposal of a learner. In round t, the learner chooses a probability mass function $\boldsymbol{p}_t$ over the set of experts $\{1, 2, \ldots, N\}$, an adversary reveals the loss vector $\boldsymbol{l}_t = (l_t(1), \ldots, l_t(N)) \in [0, 1]^N$, and the learner incurs an (expected) loss of $\langle \boldsymbol{p}_t, \boldsymbol{l}_t \rangle$. The total loss incurred by the learner after T rounds is $L_T = \sum_{t=1}^{T} \langle \boldsymbol{p}_t, \boldsymbol{l}_t \rangle$, and the total loss of choosing expert i in all the rounds is $L_T(i) = \sum_{r=1}^{T} l_t(i)$. The learner aims to minimize its cumulative regret up to round T, defined as $L_T - \min_i L_T(i)$.

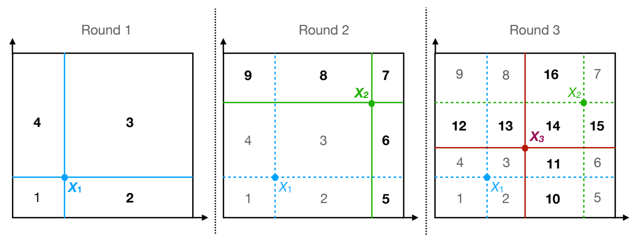
The celebrated Hedge algorithm by Freund and Schapire [1997] uses a parameter called the learning rate $\eta \geq 0$, assigns weight $w_t(i) = e^{-\eta L_{t-1}(i)}$ for each expert i based on the observed cumulative loss, and chooses expert i with probability $p_t(i) = w_t(i)/W_t$, where $W_t = \sum_{i=1}^{K} w_t(i)$ where K is the number of experts. For a suitable choice of η, Hedge has $O(\sqrt{T \log N})$ regret. Subsequent works explored improved algorithmic techniques seeking regret bounds where the dependency on T is replaced by metrics that capture the variability of the sequence of loss vectors $\boldsymbol{l}_t$ Cesa-Bianchi et al. [2007], Hazan and Kale [2010], Chiang et al. [2012]. In contrast to these works, Gofer et al. [2013] studied the dependency of the regret bound on the number of experts N. They introduced the *branching experts setting*, where new experts may be revealed in each round, and the cumulative loss of any new expert is either equal or close to the cumulative loss of one of the existing experts. They proposed an algorithm with $O(\sqrt{T N_T})$ regret, where N_T is the number of experts revealed in the first T rounds.

Motivated by learning problems that arise in out-of-distribution (OOD) detection Yang et al. [2024] and distributed Deep Learning (DL) inference Al-Atat et al. [2024], in this paper, we study a novel stochastically partitioning experts setting. This setting is a stochastic variant of the branching experts setting, where the experts revealed in each round are new sub-partitions of a hypercube $\mathbb{B}$ in d-dimensional Euclidean space, where $d < \infty$. In each round t, the environment draws a point X_t, i.i.d. from $\mathbb{B}$, using a fixed (unknown) distribution. For each chosen point, we draw d orthogonal hyperplanes parallel to the d faces of $\mathbb{B}$

passing through the point. The set of experts revealed up to round t is the set of partitions of $\mathbb{B}$ created by the intersection of the d orthogonal hyperplanes passing through each of the t points drawn up to that round, resulting in $(t+1)^d$ experts.[1] The partition of experts for one dimension and two dimensions is illustrated in Fig.1.



(a) An illustration of partitioning experts in 1-dimension over three rounds.



(b) An illustration of partitioning experts in 2-dimensions over three rounds.

Figure 1: We show the partitioning experts setting for the first three rounds for one dimension ($d = 1$) on a bounded interval in (a) and for two dimensions ($d = 2$) on a square region in (b). The new point and the new expert indices in each round are highlighted using bold fonts.

In each round, the environment only reveals the losses of the existing experts, and we allow the losses to be adversarial. We consider the *perfect clone setting* introduced in Gofer et al. [2013], where a new expert is a perfect clone of its parent expert, i.e., the cumulative loss of a new partition is equal to the cumulative loss of its parent partition. Once the new expert is revealed, its cumulative loss evolves independently from its parent expert in the subsequent rounds. We note that, in contrast to the branching experts setting where N_T is bounded and is independent of T, in the partitioning experts setting, the number of experts in round T is $(T+1)^d$.

1.1 MOTIVATING APPLICATIONS

OOD Detection: Detecting OOD samples has been widely studied as DL models fail with high confidence for these samples, resulting in serious consequences in high-risk applications. Many methods that have been developed for OOD detection use a threshold θ on a *score* x_t, calculated from soft-max values or features of the data sample, to differentiate OOD from in-distribution (ID) samples. The detected OOD samples can be deferred to human experts at some cost Vishwakarma et al. [2024]. Thus, selecting the best threshold, denoted by θ^*, that minimizes the false negative, i.e., undetected OOD samples, and false positives, i.e., detecting ID as OOD samples, is critical for the safe and reliable deployment of DL models with minimal costs.

In Fig. 2, we show an example pdf for ID and OOD samples. For a chosen threshold θ, the decision is to classify the input sample as an ID sample if x_t exceeds θ; otherwise, the sample is OOD. Since the pdfs are unknown a priori, a learner needs to learn θ^* using the following loss function.

$$\text{loss}(\theta) = \begin{cases} \text{cost for false positive} & \text{if score } x_t \geq \theta, \\ \text{human expert cost} & \text{if score } x_t < \theta. \end{cases} \quad (1)$$

Note that the challenge is that the scores can take values from a continuous set $\mathbb{B}$. However, with some effort, one can show that it is sufficient to consider only the distinct score values arrived/revealed in T rounds as the thresholds/experts. Also, whenever a new expert, i.e., a new distinct score, is revealed, the cumulative loss of this expert is equal to the cumulative loss of the highest score (revealed) less than this new score. Thus, this problem falls under the setting we study in this paper.

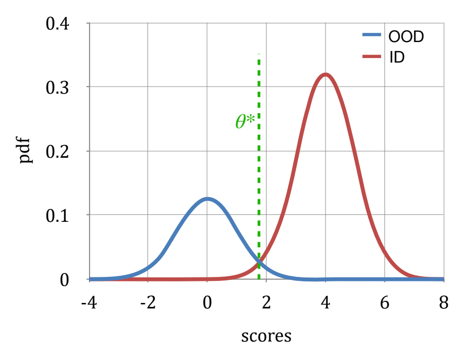


Figure 2: Differentiating ID and OOD samples using a threshold on the score.

Hierarchical Inference: The partitioning experts setting also arises in the Hierarchical Inference system proposed for distributed DL inference for classifications applications in edge AI systems Moothedath et al. [2024], Beytur et al. [2024], Al-Atat et al. [2024]. In this system, in each round t, the environment presents a data sample (e.g., image) to an end device (e.g., mobile device, IoT device, etc.). The data sample is inputted to a pre-trained local DL model that outputs soft-max values corresponding to different classes. The learner computes a confidence metric $x_t \in [0, 1]$ using these soft-max values[2]. The learner accepts the classification

[1] Since the points are drawn i.i.d. from Euclidean space, the probability of a chosen point lying on one of the d hyperplanes parallel to the faces of $\mathbb{B}$ passing through another point drawn in some other round is zero. Thus, in round t, there will be $(t+1)^d$ experts with probability one.

[2] A typical choice for the confidence metric is the maximum

in round t if the confidence metric x_t is above a threshold, which the learner aims to learn. If the learner accepts the classification, it incurs a zero loss when the classification is correct and a loss of one otherwise. If the learner rejects or offloads the classification task, it incurs an offloading cost. Similar to OOD detection, learning an optimal threshold for the confidence metric falls under the problem setting we study in this paper.

The above applications have a single threshold to learn and thus map to the partitioning experts problem with $d = 1$ (Fig. 1a). Note that the expert in our problem setting is an interval – not a threshold θ as in (1). The equivalence between an interval and a threshold can be obtained as follows. Given x_t and an interval, the threshold rule in (1) leads to the same outcome for all the thresholds in that interval. For example, in round 3 of Figure 1a, if expert 2 is chosen, then x_3 is smaller than all the points in expert 2. Thus, the sample will be classified as an OOD sample (or will be offloaded in the case of the Hierarchical Inference system).

For applications where misclassification costs are non-uniform across classes, using different thresholds for the soft-max values corresponding to the different classes will likely improve performance. In this case, learning thresholds for a d class classification task maps to a d-dimensional partitioning experts problem.

1.2 OUR CONTRIBUTIONS

We study the novel stochastically partitioning experts setting. We propose two algorithms, namely, Hedge-G, a natural extension of the Hedge algorithm for the growing experts setting, and AdaHedge-G, an adaptive learning rate variant of Hedge-G. We prove the following results on the regret of the proposed algorithms.

– Even though the number of experts grow as $(t + 1)^d$, we show that Hedge-G has $O(\sqrt{2^d T \log T})$ expected regret, which is order-optimal in T. Compare this with the Hedge algorithm, which has $O(\sqrt{dT \log T})$ regret in the special case where all the $(T + 1)^d$ experts are known apriori.

– We also show that Hedge-G achieves the sample-path regret $O(\sqrt{2^d T^{1+\epsilon} \log T})$ with probability at least $1 - T^{-\epsilon}$, for any $\epsilon > 0$.

– Hedge-G uses a fixed learning rate. We show that there is a trade-off between choosing a rate that gives the optimal expected regret guarantee and a rate that gives a useful sample-path regret guarantee. To address this limitation of Hedge-G, we propose the AdaHedge-G algorithm, a variant of the Hedge-G algorithm that uses a learning rate that adapts according to the cumulative loss of the new experts. We show that AdaHedge-G simultaneously achieves $O(\log(\log T)\sqrt{T \log T})$ expected regret, and $O(\log T \sqrt{T \log T})$ sample-path regret, with probability at least $1 - T^{-c}$, where $c > 0$ is a constant dependent on d.

soft-max value as the data sample is typically classified into the class with the maximum soft-max value.

2 RELATED WORK

The decision-theoretic online learning problem is a variant of the classical prediction with expert advice Littlestone and Warmuth [1994], Vovk [1995] and has received much attention in the past three decades. We summarize the related works that studied the variants of this problem, where the set of experts is very large or growing over time.

For the setting where the number of experts is large, Chaudhuri et al. [2009] proposed a parameter-free version of Hedge and showed that it outperforms the classical Hedge algorithm. Chernov and Vovk [2010] considered the setting with a large number of experts where multiple experts can be near clones of each other. Further, they considered that the regret of the algorithm with respect to any newly arrived expert is assumed to be zero, and it is accumulated thereafter. They provided regret guarantees as a function of the effective number of experts, i.e., the number of unique experts available to the learner. In contrast to the aforementioned works, Luo and Schapire [2015] proposed AdaNormalHedge, which is agnostic to the number of experts and, therefore, can be used in a setting where the number of experts is unknown or changing. At each time-step t, AdaNormalHedge creates N sleeping experts, indexed by (t, i) for $i \in 1, \dots, N$, that are asleep before time-step t, and wake up at time-step t and suffer the same loss as that of expert i from then onwards. It follows that, in total, there will be NT sleeping experts after T rounds. We note that AdaNormalHedge's computation complexity will be t times higher than Hedge-G in round t. Whether AdaNormalHedge can be adapted to the partitioning experts setting and how its regret bound compares to that of Hedge-G remains an open question. In the aforementioned works, however, the newly arriving experts are not correlated with the experts who came before them.

Cohen and Mannor [2017] studied the setting where all the experts are known apriori, and their losses are revealed in each round, but the number of experts is potentially infinite. The focus here was on identifying a small set of experts such that all other experts are close to any one expert in this small set in terms of their cumulative loss. The authors proposed an algorithm with provable performance guarantees that depend on the ϵ-covering number of the sequence of loss functions. They also proposed a method to compute the optimal ϵ in hindsight.

Mourtada and Maillard [2017] studied the growing number of experts setting, where new experts are revealed over time. The key contribution in this work is two-fold. The authors considered multiple definitions of regret, namely shifting regret and sparse shifting regret, to account for the fact that the expert set is growing over time. They designed computationally inexpensive policies with order-optimal regret performance for all the regret definitions considered. The proposed algorithms are anytime and parameter-free. In Gyorfi et al. [1999], the set of experts grew at an exponentially decaying rate, and the goal was to make predictions about a stationary ergodic time series. In Hazan and Seshadhri [2009], Shalizi et al. [2011], the focus was on predicting a non-stationary time series using a growing set of experts. In contrast to the above works, experts arrive at a much faster rate in our setting.

As mentioned, our partitioning experts setting is closely related to the branching experts setting first studied by Gofer et al. [2013]. In this work, even though the number of experts increases with time, N_T, the total number of experts revealed after T rounds is assumed to be large but finite. Wu et al. [2021] further studied the branching experts setting where the losses are stochastic processes with unknown distributions. They proposed an optimal policy for both adversarial and stochastic losses. Our setting differs from the branching experts setting as we have an uncountably infinite set of experts from which $(T+1)^d$ experts are revealed in T rounds. Another difference is that the number of new experts revealed in round t is $(t+1)^d - t^d$.

In the classical Hedge algorithm, the learning rate is a function of the time horizon T. Thus, it is unsuitable for settings where the time horizon is unknown. The algorithms proposed in Erven et al. [2011], De Rooij et al. [2014] addressed this limitation by adapting the learning rate without the need to know the value of T. In contrast, we assume T is given but adapt the learning rate in AdaHedge-G according to the observed losses so that it simultaneously achieves near-optimal bound for expected regret and non-trivial sample-path regret guarantees.

3 STOCHASTICALLY PARTITIONING EXPERTS SETTING

In this work, experts are represented by partitions of a hypercube $\mathbb{B}$ in a d-dimensional Euclidean space. As discussed above, in each round t, the environment draws a point X_t, i.i.d.[3] from $\mathbb{B}$, using a fixed (unknown) distribution. For each such point, we draw d hyperplanes passing through the point, parallel to the d faces of $\mathbb{B}$. The set of experts available in round t is the set of partitions of $\mathbb{B}$ created by all the hyperplanes drawn up to that round. The partitioning process for $d = 1$ and $d = 2$ is illustrated in Fig.1.

In round 1, the environment draws a point $X_1 \in \mathbb{B}$ creating 2^d experts, which we index $1, \dots, 2^d$. Similarly, in round t, the environment samples point $X_t \in \mathbb{B}$ resulting in $n_t = (t+1)^d$ experts. Among these experts, $(t+1)^d - t^d$ are new experts. We say an expert is a child of a parent expert if the former is a sub-partition of the latter expert. We assign the index of each parent expert to one of its children and assign new indices $t^d + 1, \dots, (t+1)^d$ to the remaining unindexed new experts. We use $\mathcal{B}_t = \{1, \dots, n_t\}$ to denote the set of indices at the end of round t.

In round t, the environment first samples X_t, and the learner chooses a probability mass function $\boldsymbol{p}_t$ over the set of experts $\mathcal{B}_t$. Following this, the environment reveals the loss vector $\boldsymbol{l}_t = (l_t(1), \dots, l_t(n_t)) \in [0,1]^{n_t}$. The learner, therefore, incurs an expected loss of $\langle \boldsymbol{p}_r, \boldsymbol{l}_r \rangle$. The cumulative loss of expert $i \in \mathcal{B}_t$ up to time t is $L_t(i) = \sum_{r=1}^{t} l_r(i)$, and the expected cumulative loss of the learner up to time t is

$$L_t = \sum_{r=1}^{t} \langle \boldsymbol{p}_r, \boldsymbol{l}_r \rangle.$$

For each new expert $i \in \mathcal{B}_t \backslash \mathcal{B}_{t-1}$, its cumulative loss up to time t, i.e., $L_{t-1}(i)$ is equal to the cumulative loss of its parent expert from $\mathcal{B}_{t-1}$. However, the subsequent losses of the new experts evolve independently from those of their parent experts.

The loss functions are generated by an *oblivious adversary* and communicated to the environment causally. Our results hold for the setting where the oblivious adversary knows the relative order of the X_ts apriori and selects a sequence of losses according to a deterministic mapping from this ordering. Specifically, at $t = 0$, the adversary knows if $X_u < X_v$ or not, for all $1 \le u < v \le T$, and can exploit this information to design the loss vectors for $t \ge 1$. Note that our adversary is more powerful than an alternative oblivious adversary that does not have this side information.

We define

$$L_t^* = \min_{i \in \mathcal{B}_t} L_t(i).$$

Given the time horizon T, we aim to minimize the *expected regret*

$$R_T = \mathbb{E}[L_T - L_T^*],$$

where the expectation is with respect to the joint distribution of the sequence of points $\mathbf{X}_T = \{X_1, \dots, X_T\}$ drawn by the environment in T rounds. Note that $\mathbb{E}[L_t^*]$ will be equal to L_t^* if the loss vectors generated are independent of the points sampled by the environment and the regret bounds we prove will still hold.

We also study the *sample-path regret*

$$\hat{R}_T = L_T - L_T^*,$$

[3] A typical assumption during the DL training phase is that the data samples are drawn i.i.d. using an unknown distribution. In the Hierarchical Inference application, the same assumption is made for the inference phase, where the data samples are drawn i.i.d. (cf. Al-Atat et al. [2024]), which corresponds to X_t being drawn i.i.d. in our partitioning setting.

Algorithm 1 Hedge-G for partitioning experts

1: **Initialize:** $\mathcal{B}_0 = \{1\}$, $n_0 = 0$, $w_1 = 1$, and $W_1 = 1$.
2: **for** each round $t = 1, 2, \ldots, T$ **do**
3: X_t is drawn i.i.d. from $\mathbb{B}$ and new partitions are revealed
4: $n_t = (t+1)^d$ and $\mathcal{B}_t = \mathcal{B}_{t-1} \cup \{n_{t-1}+1, \ldots, n_t\}$
5: For $i \in \mathcal{B}_t \backslash \mathcal{B}_{t-1}$, given $L_{t-1}(i)$, compute new weights $w_t(i) = e^{-\eta L_{t-1}(i)}$
6: $\hat{W}_t = W_t + \sum_{i \in \mathcal{B}_t \backslash \mathcal{B}_{t-1}} w_t(i)$
7: Compute $p_t(i) = \frac{w_t(i)}{\hat{W}_t}$, for all $i \in \mathcal{B}_t$.
8: Choose an expert using $\boldsymbol{p}_t$, observe $\boldsymbol{l}_t$, and incur the loss $\langle \boldsymbol{p}_t, \boldsymbol{l}_t \rangle$.
9: Update the weights $w_{t+1}(i) = e^{-\eta l_t(i)} w_t(i)$, for all $i \in \mathcal{B}_t$.
10: Cumulative weight $W_{t+1} = \sum_{i=1}^{n_t} w_{t+1}(i)$.
11: **end for**

and provide bounds in the high probability regime.

Remark 1: One can alternatively interpret the partitioning experts setting as follows. Instead of treating each partition as an expert, consider each point in $\mathbb{B}$ as an expert. When the environment draws an expert, it only reveals a single loss value per partition instead of losses for all the points in $\mathbb{B}$. For example, this loss value may be the average loss over all experts (points) in the partition. Since we can only work with the loss values per partition instead of losses of the individual experts, the setting where we carry over cumulative losses of the parent partition to the new sub-partitions is well-motivated, especially given its applicability in the classification application discussed in Section 1.1.

Remark 2: Note that the regret bounds we prove are valid for any sequence of losses the oblivious adversary generates, and, thus, they hold for the supremum over all loss sequences.

4 THE HEDGE-G ALGORITHM: REGRET ANALYSIS

We propose an algorithm called Hedge-G, a natural extension of the Hedge algorithm for the growing experts setting, that introduces a new weight whenever a new expert arrives. Similar to the branching experts setting, in our setting, these new weights can be readily computed as the cumulative losses of the new experts are the same as their parent experts. In Algorithm 1, we present Hedge-G adapted to the partitioning experts setting.

The regret analysis for Hedge-G differs from Hedge in that the introduction of new weights in line 5 of Algorithm 1 implies that W_t does not normalize the weights $w_t(i)$, for $i \in \mathcal{B}_t$. A key step in our analysis of Hedge-G is to compute the expected value of the quantity Y_t, the ratio between the sum of new weights $w_t(i)$ and W_t, given by

$$Y_t = \frac{\sum_{i=n_{t-1}+1}^{n_t} w_t(i)}{W_t} = \frac{\sum_{i=n_{t-1}+1}^{n_t} e^{-\eta L_{t-1}(i)}}{\sum_{j \in \mathcal{B}_{t-1}} e^{-\eta L_{t-1}(j)}}. \quad (2)$$

The following theorem characterizes an upper bound on the cumulative loss of Hedge-G.

Theorem 4.1. *An upper bound for the cumulative loss of Hedge-G is given by*

$$L_T \leq L_T^* + \frac{T\eta}{8} + \frac{\sum_{t=1}^{T} Y_t}{\eta}. \quad (3)$$

Proof. We write

$$\log \frac{W_{t+1}}{W_t} = \log \frac{W_{t+1}}{\hat{W}_t} + \log \frac{\hat{W}_t}{W_t}. \quad (4)$$

Given $\hat{W}_t = \sum_{i \in \mathcal{B}_t} e^{-\eta L_{t-1}(i)}$, we upper bound the second term in RHS of (4) as follows.

$$\log \frac{\hat{W}_t}{W_t} = \log \left(\frac{\sum_{i \in \mathcal{B}_{t-1}} e^{-\eta L_{t-1}(i)} + \sum_{i=n_{t-1}+1}^{n_t} e^{-\eta L_{t-1}(i)}}{\sum_{i \in \mathcal{B}_{t-1}} e^{-\eta L_{t-1}(i)}} \right)$$
$$= \log(1 + Y_t) \leq Y_t. \quad (5)$$

Next, we upper and lower bound $\log \frac{W_{T+1}}{W_1}$. By definition,

$$\begin{aligned} \log \frac{W_{T+1}}{W_1} &= \log \left(\prod_{t=1}^{T} \frac{W_{t+1}}{W_t} \right) \\ &= \sum_{t=1}^{T} \left[\log \frac{W_{t+1}}{\hat{W}_t} + \log \frac{\hat{W}_t}{W_t} \right] \\ &\leq \sum_{t=1}^{T} \left[-\eta \langle \boldsymbol{p}_t, \boldsymbol{l}_t \rangle + \frac{\eta^2}{8} + Y_t \right] \\ &= -\eta L_T + \frac{\eta^2 T}{8} + \sum_{t=1}^{T} Y_t. \end{aligned} \quad (6)$$

In the third step above, we have used (5) and Hoeffding's lemma to upper bound $\log \frac{W_{t+1}}{\hat{W}_t}$. Also,

$$\begin{aligned} \log \frac{W_{T+1}}{W_1} &= \log \sum_{i=1}^{n_T} e^{-\eta L_T(i)} \\ &\geq \log \max_{i \in \mathcal{B}_T} e^{-\eta L_T(i)} \\ &\geq \max_{i \in \mathcal{B}_T} \log e^{-\eta L_T(i)} = -\eta L_T^*. \end{aligned} \quad (7)$$

From (6) and (7), we obtain the result. □

To obtain a bound on the expected regret of Hedge-G from Theorem 4.1, we need to compute $\sum_{t=1}^{T} \mathbb{E}[Y_t]$. A primer for computing $\mathbb{E}[Y_t]$ is the following lemma which states that in any slot the new point is equally likely to belong to any one of the existing partitions of $\mathbb{B}$.

Lemma 4.2. *Given that the sequence of points $\{X_t\}$ are drawn i.i.d. from $\mathbb{B}$, the point X_t drawn in round t is equally likely to belong to any one of the existing t^d partitions, i.e.,*

$$\mathbb{P}(X_t \in \text{ partition } i) = \frac{1}{t^d},\ \forall i \in \mathcal{B}_{t-1}.$$

Our next result uses Lemma 4.2 to compute $\mathbb{E}[Y_t]$.

Lemma 4.3. $\mathbb{E}[Y_t] = \left(1 + \frac{1}{t}\right)^d - 1 \le \frac{2^d}{t}.$

The proofs of Lemmas 4.2 and 4.3 are given in the Appendix.

Taking expectation on both sides in (3) (Theorem 4.1) and using Lemma 4.3, we obtain the following bound on expected regret:

$$R_T \le \frac{\eta T}{8} + \frac{2^d}{\eta}\sum_{t=1}^{T}\frac{1}{t} \le \frac{\eta T}{8} + \frac{2^d(\log T + 1)}{\eta}. \quad (8)$$

The regret bound in the following corollary immediately follows from (8).

Corollary 4.3.1. *For the partitioning experts setting, for Hedge-G with $\eta = \sqrt{2^{d+3}(d\log T + 1)/T}$, the expected regret $R_T = O(\sqrt{2^d T\log T})$.*

Note that, in our problem setting, d is a constant determined by the application under consideration. For instance, in most cases of OOD detection and Hierarchical Inference applications, $d = 1$. In the following, we show that Hedge-G is order-optimal with respect to T.

Lower bound: The Prediction with Expert Advice (PEA) with K experts has the lower bound $\sqrt{T\log K}$ for an oblivious adversary Freund and Schapire [1999]. To prove the lower bound for the partitioning experts, we construct the following problem instance for $d = 1$. Let the oblivious adversary assign 0 loss to the first $T/2$ spawned experts arrived in the first $T/2$ time steps. From $T/2+1$, each new expert always receives a loss higher than its parent, and the first $T/2$ experts receive losses as in the PEA setting. Using only the first $T/2$ experts is sufficient to reduce regret, and we obtain a regret lower bound of $\sqrt{T\log T}$. For $d > 1$, $(T/2)^d$ experts spawn in the first $T/2$ time steps, and we use a similar loss assignment as above to obtain $\Omega(\sqrt{dT\log T})$ lower bound. Since this lower bound is valid for any realization, the expected regret of any algorithm is also $\Omega(\sqrt{dT\log T})$. Thus, from Corollary 4.3.1, we see that Hedge-G has order-optimal expected regret with respect to the time-horizon T. Note that the vanilla Hedge algorithm achieves $O(\sqrt{dT\log T})$ expected regret only when all the $(T+1)^d$ experts are known apriori, and their losses are revealed in each round.

Remark 3: We can improve the regret bound of Hedge-G with respect to d by using a tighter upper bound for $E[Y_t]$. For example, using $E[Y_t] = (1+1/t)^d - 1 \le d/t + 2^d/t^2$ and repeating the analysis with this tighter upper bound and $\eta = \sqrt{8d(\log T + 1)/T}$, we obtain an improved bound $O(\sqrt{dT\log T} + 2^d\sqrt{T/d\log T})$. In the first term, the dependence is on $\sqrt{d}$ instead of $\sqrt{2^d}$. In the second term, the dependence is on $2^d/\sqrt{d\log T}$, but notice that $\sqrt{\log T}$ is in the denominator. Thus, even if d is large, the regret bound is dominated by $\sqrt{dT\log T}$ term.

Corollary 4.3.2. *For the partitioning experts setting, for any $\epsilon > 0$, Hedge-G with $\eta = \sqrt{\frac{2^{d+3}(\log T+1)}{T^{1-\epsilon}}}$ achieves the sample-path regret $\hat{R}_T = O(\sqrt{2^d T^{1+\epsilon}\log T})$ with probability at least $1 - T^{-\epsilon}$.*

Proof. Using Markov inequality for the summation of the random variables Y_t, we get

$$\mathbb{P}\left(\sum_{t=1}^{T} Y_t \le T^{\epsilon}\sum_{t=1}^{T}\mathbb{E}[Y_t]\right) \ge 1 - \frac{\sum_{t=1}^{T}\mathbb{E}[Y_t]}{T^{\epsilon}\sum_{t=1}^{T}\mathbb{E}[Y_t]} = 1 - T^{-\epsilon}.$$

Using this result in (3) and the upper bound for $\mathbb{E}[Y_t]$ from Lemma 4.2, we obtain, with probability at least $1 - T^{-\epsilon}$,

$$\hat{R}_T \le \frac{\eta T}{8} + \frac{2^d T^{\epsilon}(\log T + 1)}{\eta}.$$

Choosing $\eta = \sqrt{\frac{2^{d+3}(\log T+1)}{T^{1-\epsilon}}}$ results in $\hat{R}_T \le \sqrt{2^{d-1}T^{1+\epsilon}(\log T + 1)}$. □

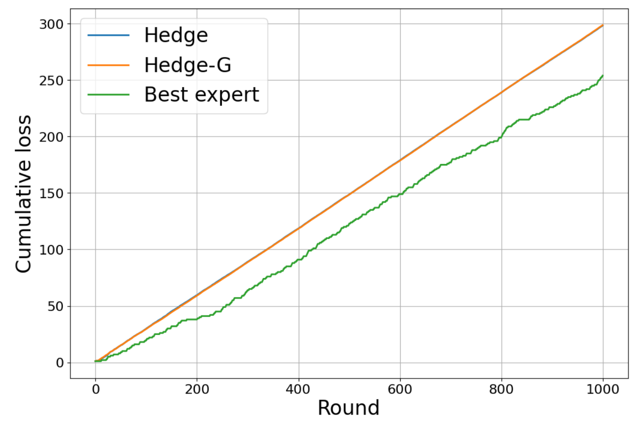
From Corollaries 4.3.1 and 4.3.2, it follows that for $\eta = \sqrt{\frac{2^{d+3}(\log T+1)}{T^{1-\epsilon}}}$, the sample-path regret of Hedge-G is $O(\sqrt{T^{1+\epsilon}\log T})$ with high probability and its expected regret is $O(T^{\frac{\epsilon}{2}}\sqrt{T\log T})$. Compared to this, the expected regret for Hedge-G with $\eta = \sqrt{2^{d+3}(\log T + 1)/T}$ is $O(\sqrt{T\log T})$, but this value of η leads to a sample-path regret bound that holds with probability zero, as $\epsilon = 0$. Therefore, to obtain a high probability bound on sample-path regret of Hedge-G using Theorem 4 and Markov's inequality, we use a value of η for which the expected regret is higher than the optimal by a factor of $O(T^{\frac{\epsilon}{2}})$. In Section 5, we address this limitation of Hedge-G by adapting the learning rate based on the losses revealed by the adversary.

Remark 4: Note that if X_t are drawn adversarially from $\mathbb{B}$, Hedge-G has linear regret. We construct the following problem instance for $d = 1$. The adversary always splits the best expert in each round, resulting in two experts, j and k. Uniformly at random, the adversary assigns a loss of one to one expert in the set $\{j, k\}$ and zero to the other expert. For all other experts $i \ne j, k$, it assigns a loss of one. For this problem instance, at any time t, $L_t^* = 0$, but the expected loss for Hedge-G in that time step will be at least $\frac{1}{2}$. Hence, Hedge-G has expected regret of at least $\frac{T}{2}$. This result is expected because if X_t are adversarially drawn from $\mathbb{B}$, then the partitioning expert setting is a special case of the branching experts setting studied by Gofer et al. [2013]. It is known for the branching experts setting, the regret of any algorithm is $\Omega(\sqrt{TN_T})$, where N_T for the partitioning expert setting is equal to $(T+1)^d$.

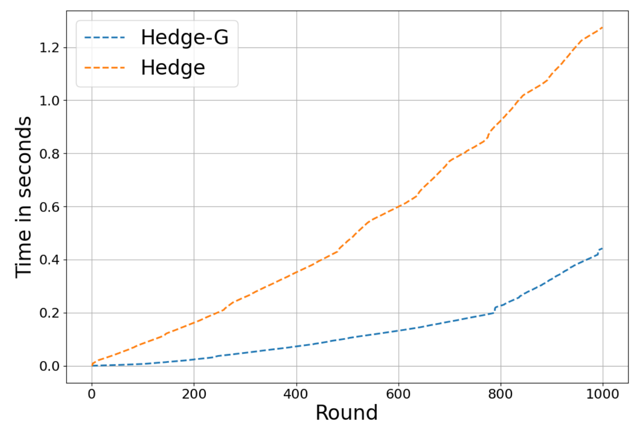
4.1 PERFORMANCE COMPARISON

We compare the cumulative loss and runtime performance of Hedge-G with the Hedge algorithm, which has prior knowledge of all the expert intervals (i.e., the intervals that will be formed in T rounds). We simulate for $d = 1$. At each time step t, the loss assigned to a Hedge expert corresponds to the loss of the corresponding parent expert in the same simulation instance under Hedge-G. The points X_t are sampled independently from a uniform distribution $\mathcal{U}[0,1]$, and the loss for each expert at each time step is generated from a Bernoulli distribution with parameter 0.3, i.e., Bernoulli(0.3). The experiments were performed on a machine equipped with an Intel(R) Xeon(R) CPU running at 2.20GHz. The processor has a cache size of 56.32 MB, and the system is equipped with 12.7 GB of RAM.

The results, illustrated in Figure 3a, demonstrate that Hedge-G achieves performance comparable to that of Hedge, despite lacking prior knowledge of the expert intervals available to the latter. By the end of 1000 rounds, Hedge-G incurs an additional cumulative loss of only 0.383 compared to Hedge. Figure 3b presents the cumulative runtime as a function of the number of rounds for both algorithms. As anticipated, Hedge-G incurs significantly lower computational overhead, leading to a noticeably reduced runtime.



(a) Cumulative loss of Hedge-G, Hedge, and Best expert



(b) Running time of the algorithm vs Number of rounds

Figure 3: Comparison between Hedge-G and Hedge

5 ADAHEDGE-G: HEDGE-G WITH ADAPTIVE LEARNING RATE

In this section, we propose a variant of Hedge-G called AdaHedge-G and show that its expected regret is near-optimal while simultaneously achieving the same high probability bound for the sample-path regret for Hedge-G stated in Corollary 4.3.2.

The details of AdaHedge-G are presented in Algorithm 2. The key idea behind the algorithm is to track the summation of Y_ts using the variable S and suitably change the learning rate over rounds using a doubling trick. In particular, we partition the time into segments, where *segment* i spans the number of rounds for which $S \leq 2^{id}$. At the start of any segment i, we reset the value of S to zero, choose an equal weight for all the existing experts (from the previous segment), and use Hedge-G with learning rate $\eta_i = \sqrt{8(2^{id} + \log \tau_i)/T}$, where τ_i is the round in which the segment starts.

The next theorem characterizes an upper bound on the cumulative loss of AdaHedge-G.

Theorem 5.1. *An upper bound for the cumulative loss of AdaHedge-G is given by*

$$\begin{aligned} L_T \leq & L_T^* + \frac{2^{d-\frac{1}{2}}}{2^{\frac{d}{2}}-1}\sqrt{T\left(\sum_{t=1}^{T} Y_t + 1\right)} \\ & + \left(1 + \frac{2}{d}\log_2\left(\sqrt{\sum_{t=1}^{T} Y_t + 1}\right)\right)\sqrt{dT\log T/2}. \end{aligned} \tag{9}$$

Proof. Let r_i be the length of the i^{th} segment, i.e., the number of rounds in the i^{th} segment. By definition of a segment, we have

$$r_i = \min\left\{r : \sum_{i=\tau_i}^{r} Y_i > 2^{id}\right\} - \tau_i,$$

where τ_i is the round in which the segment i starts and is given by $\tau_i = \sum_{u=1}^{i-1} r_u + 1$. Let $R^{(i)}$ denote the regret

Algorithm 2 AdaHedge-G

Initialize: $r \leftarrow 0, S \leftarrow 0, \tau \leftarrow 1, b \leftarrow 2^d, \mathbf{w}_1 = 1$, and $\eta \leftarrow \sqrt{\frac{8b}{T}}$.
for $t = 1, \ldots, T$ **do**
  X_t is drawn i.i.d. from $\mathbb{B}$
  Calculate Y_t using (2)
  if $S + Y_t > b$ **then**
    Start a new segment
    $\mathbf{w}_t = (w_1, \ldots, w_{t^d}) = \left(\frac{1}{t^d}, \ldots, \frac{1}{t^d}\right)$
    $S \leftarrow 0$
    $b \leftarrow 2^d b$
    $\eta \leftarrow \sqrt{\frac{8(b + d\log t)}{T}}$
  end if
  $S \leftarrow S + Y_t$
  Use Hedge-G with already observed X_t, initial weight vector $\mathbf{w}_t$ and learning rate η.
end for

incurred in segment i. It follows that

$$R^{(i)} = \sum_{u=\tau_i}^{\tau_{i+1}-1} l_u - \min_{j \in \mathcal{B}_{\tau_{i+1}-1}} \sum_{u=\tau_i}^{\tau_{i+1}-1} l_u(j).$$

We repeat the regret analysis from the proof of Theorem 4.1 for $R^{(i)}$ and obtain

$$R^{(i)} \le \frac{\eta_i r_i}{8} + \frac{S_i + d\log\tau_i}{\eta_i} \le \sqrt{T(2^{id} + d\log T)/2},$$

where, we have used $r_i \le T$, $\tau_i \le T$,

$$S_i = \sum_{r=\tau_i}^{\tau_{i+1}-1} Y_r \le 2^{id}, \text{ and } \eta_i = \sqrt{\frac{8(2^{id} + d\log\tau_i)}{T}}.$$

Note the weights are reinitialized to $1/\tau_i^d$ at the start of the segment and this yields the additional term of $d\log\tau_i$ when upper bounding $\log\frac{W_{\tau_{i+1}-1}}{W_{\tau_i}}$ in the analysis leading to (7).

Let m denote the last segment that started before round T. We add regret across all the m segments and obtain,

$$\begin{aligned} L_T - L_T^* \le \sum_{i=1}^{m} R^{(i)} \le& \sqrt{\frac{T}{2}}(\sqrt{2^d + d\log T} \quad (10)\\ &+ \sqrt{2^{2d} + d\log T} \\ &+ \ldots + \sqrt{2^{md} + d\log T}) \\ \le& \sqrt{\frac{T}{2}} \sum_{i=1}^{m} 2^{\frac{id}{2}} + m\sqrt{dT\log T/2} \\ \le& \frac{\sqrt{\frac{T}{2}} 2^{\frac{(m+1)d}{2}}}{2^{\frac{d}{2}} - 1} + m\sqrt{dT\log T/2}. \quad (11) \end{aligned}$$

In the second step above, we have used $\sqrt{x+y} \le \sqrt{x} + \sqrt{y}$. Further, we have

$$\sum_{i=1}^{T} Y_t \ge \sum_{i=1}^{m-1} 2^{id} \ge 2^d \frac{2^{(m-1)d} - 1}{2^d - 1}.$$

Therefore,

$$2^{\frac{md}{2}} \le 2^{\frac{d}{2}} \sqrt{\sum_{t=1}^{T} Y_t + 1} \quad (12)$$

$$\begin{aligned} \implies m &\le \frac{2}{d}\log_2\left(2^{\frac{d}{2}}\sqrt{\sum_{t=1}^{T} Y_t + 1}\right) \\ &= 1 + \frac{2}{d}\log_2\left(\sqrt{\sum_{t=1}^{T} Y_t + 1}\right). \quad (13) \end{aligned}$$

Substituting (12) and (13) in (11), we obtain the result □

The next theorem provides guarantees on the regret of AdaHedge-G.

Theorem 5.2. *For the partitioning experts setting AdaHedge-G has the following regret bounds.*

(i) The expected regret $R_T = O(\log(\log T)\sqrt{T\log T})$.

(ii) For $d \ge 1$ *and some constant* c *depending on dimension* d*, the sample-path regret* $\hat{R}_T = O(\log T\sqrt{T\log T})$*, with probability at least* $1 - T^{-c}$*. For* $d = 1$*, the sample-path regret can be improved to* $\hat{R}_T = O(\log(\log T)\sqrt{T\log T})$*, with probability at least* $1 - (eT)^{-0.25}$.

Proof. (i) From Lemma 4.3 and applying Jensen's inequality to (9) and substituting $\mathbb{E}\left[\sum_{t=1}^{T} Y_t\right] \le \log T + 1$, we get the result.

(ii) We present the proof for $d = 1$. Proof for $d > 1$ follows similar steps with more involved analysis and is deferred to the Appendix. We have

$$Y_t = \frac{e^{-\eta L_{t-1}(n_t)}}{\sum_{j \in \mathcal{B}_{t-1}} e^{-\eta L_{t-1}(j)}}. \quad (14)$$

Note that $e^{\eta(L_{t-1}(i) - L_{t-1}(j))} \ge 0$ for all i, j, and $L_{t-1}(i) \ge L_{t-1}(j)$ implies $e^{\eta(L_{t-1}(i) - L_{t-1}(j))} \ge 1$, since $\eta > 0$. Therefore,

$$\begin{aligned} Y_t &= \frac{1}{\sum\limits_{j: L_{t-1}(j) > L_{t-1}(n_t)} e^{\eta(L_{t-1}(n_t) - L_{t-1}(j))} + \sum\limits_{j: L_{t-1}(j) \le L_{t-1}(n_t)} e^{\eta(L_{t-1}(n_t) - L_{t-1}(j))}} \\ &\le \frac{1}{\sum_{j \in \mathcal{B}_{t-1}} \mathbb{1}_{\{L_{t-1}(j) \le L_{t-1}(n_t)\}}}. \quad (15) \end{aligned}$$

In round t, we define a random variable Z_t such that $Z_t = k^{-1}$, if X_t falls in the k^{th} best expert, i.e., $\sum_{j\in\mathcal{B}_{t-1}} \mathbb{1}_{\{L_{t-1}(j)\leq L_{t-1}(n_t)\}} = k$. Note that in the presence of ties, we break ties arbitrarily and strictly order the experts. In this case, if X_t falls in the k^{th} best partition, it can be shown that the denominator in (15) is at least $k+1$. Thus, $Y_t < Z_t = 1/k$.

From (15), we have $Y_t \leq Z_t$, for all t. From Lemma 4.2, the probability that X_t falls in k^{th} best partition is $\frac{1}{t}$, which implies $\mathbb{P}(Z_t = k^{-1}) = 1/t$. Therefore,

$$\mathbb{E}[Z_t] = \sum_{k=1}^{t} \frac{1}{t}\frac{1}{k} \leq \frac{\log t + 1}{t}, \qquad (16)$$

$$\implies \sum_{t=1}^{T} \mathbb{E}[Z_t] \leq (\log T + 1)^2. \qquad (17)$$

Further, we have

$$\begin{aligned}
&\mathbb{P}\left(\sum_{t=1}^{T} Y_t - \sum_{t=1}^{T} \mathbb{E}[Y_t] > \delta\right) \\
\leq &\mathbb{P}\left(\sum_{t=1}^{T} Z_t - \sum_{t=1}^{T} \mathbb{E}[Y_t] > \delta\right) \\
\leq &\mathbb{P}\left(\sum_{t=1}^{T} Z_t - \sum_{t=1}^{T} \mathbb{E}[Z_t] > \delta - \sum_{t=1}^{T} \mathbb{E}[Z_t] + \sum_{t=1}^{T} \mathbb{E}\left[Y_t\right]\right) \\
\leq &\mathbb{P}\left(\sum_{t=1}^{T} Z_t - \sum_{t=1}^{T} \mathbb{E}[Z_t] > \delta - (\log T + 1)^2 + \log T\right).
\end{aligned} \qquad (18)$$

To get (18), we use (16), (17), and the fact that $\sum_{t=1}^{T} \mathbb{E}\left[Y_t\right] \geq \log T$. Since the Z_ts are independent and are upper bounded by one, using Bernstein's inequality, we get

$$\mathbb{P}\left(\sum_{t=1}^{T} Z_t - \sum_{t=1}^{T} \mathbb{E}[Z_t] > \delta'\right) \leq e^{-\frac{\delta'^2/2}{V_n + \delta'/3}}, \qquad (19)$$

where $V_n = \sum_{t=1}^{T} \text{Var}(Z_t)$, and $\delta' = \delta - (\log T + 1)^2 + \log T$. We have

$$\text{Var}(Z_t) = \sum_{j=1}^{t} \frac{1}{t}\frac{1}{j^2} - \mathbb{E}[Z_t]^2 \leq \frac{\pi^2}{6t}$$

$$\implies V_n = \sum_{t=1}^{T} \text{Var}(Z_t) \leq \frac{\pi^2}{6}(\log T + 1). \qquad (20)$$

Choosing $\delta = (\log T + 1)^2 + 1$ results in $\delta' = \log T + 1$. Substituting δ' and (20) in (19), we obtain

$$\mathbb{P}\left(\sum_{i=1}^{T} Y_i - \sum_{i=1}^{T} \mathbb{E}[Y_i] > \delta\right) \leq e^{-\frac{(\log T+1)^2/2}{\frac{\pi^2}{6}(\log T+1) + (\log T+1)/3}} \leq e^{\frac{-3\log(eT)}{\pi^2+2}} \leq (eT)^{-0.25}.$$

Substituting the above result in (9) proves the final result.

□

From parts (i) and (ii) of Theorem 5.2, we observe that AdaHedge-G has near-optimal expected regret (sub-optimality of a factor of $\log(\log T)$) and it also has the same high probability bound on sample-path regret as that of Hedge-G in Corollary 4.3.2. AdaHedge-G thus addresses the limitation of Hedge-G discussed at the end of Section 4. Further, in part (ii) of the theorem, for the special case $d = 1$, we provide a sample-path regret that is near-optimal with high probability, independent of ϵ. Proving a tighter bound for $d > 1$, similar to the case $d = 1$, remains an open problem.

6 CONCLUDING REMARKS

In this work, we propose an adaptation of Hedge for the partitioning experts setting where the number of experts increases polynomially with time. We show that our algorithm and its adaptive rate variant have (near-)optimal expected regret bounds and non-trivial sample path-regret bounds under the high probability regime.

Possible extensions of this work include: *(i)* designing anytime policies when T is unknown, *(ii)* considering the setting where the rate of growth of the experts is random, i.e., the environment samples a random number of points in each round, and *(iii)* studying the setting where the new experts are approximate clones of the parent experts instead of being perfect clones. Further, the setting of stochastically partitioning experts with stochastic losses can also be explored.

ACKNOWLEDGMENTS

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant (54196-52150) and an SCPP grant from IIT Bombay, India.

References

Ghina Al-Atat, Puranjay Datta, Sharayu Moharir, and Jaya Prakash Champati. Regret bounds for online learning for hierarchical inference. In *Proc. ACM MobiHoc*, pages 281–290, 2024.

Hasan Burhan Beytur, Ahmet Gunhan Aydin, Gustavo de Veciana, and Haris Vikalo. Optimization of offloading policies for accuracy-delay tradeoffs in hierarchical inference. In *Proc. IEEE INFOCOM*, pages 1989–1998. IEEE, 2024.

Nicolo Cesa-Bianchi, Yishay Mansour, and Gilles Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66:321–352, 2007.

Kamalika Chaudhuri, Yoav Freund, and Daniel J Hsu. A parameter-free hedging algorithm. In *Proc. NIPS*, volume 22, 2009.

Alexey Chernov and Vladimir Vovk. Prediction with advice of unknown number of experts. In *Proc. UAI*, page 117–125, 2010.

Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Proc. COLT*, pages 6.1–6.20, 2012.

Alon Cohen and Shie Mannor. Online learning with many experts. *CoRR*, abs/1702.07870, 2017.

Steven De Rooij, Tim Van Erven, Peter D Grünwald, and Wouter M Koolen. Follow the leader if you can, hedge if you must. *The Journal of Machine Learning Research*, 15(1):1281–1316, 2014.

Tim Erven, Wouter M Koolen, Steven Rooij, and Peter Grünwald. Adaptive hedge. In *NIPS*, volume 24, 2011.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55 (1):119–139, 1997.

Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

Eyal Gofer, Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Regret minimization for branching experts. In *Proc. COLT*, pages 618–638, 2013.

L Gyorfi, Gábor Lugosi, and Gusztáv Morvai. A simple randomized algorithm for sequential prediction of ergodic time series. *IEEE Transactions on Information Theory*, 45(7):2642–2650, 1999.

Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80:165–188, 2010.

Elad Hazan and Comandur Seshadhri. Efficient learning algorithms for changing environments. In *Proc. ICML*, pages 393–400, 2009.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108 (2):212–261, 1994.

Haipeng Luo and Robert E Schapire. Achieving all with no parameters: Adanormalhedge. In *Proc. COLT*, pages 1286–1304, 2015.

Vishnu Narayanan Moothedath, Jaya Prakash Champati, and James Gross. Getting the best out of both worlds: Algorithms for hierarchical inference at the edge. *IEEE Transactions on Machine Learning in Communications and Networking*, 2:280–297, 2024.

Jaouad Mourtada and Odalric-Ambrym Maillard. Efficient tracking of a growing number of experts. In *Proc. ALT*, pages 517–539. PMLR, 2017.

Cosma Rohilla Shalizi, Abigail Z Jacobs, Kristina Lisa Klinkner, and Aaron Clauset. Adapting to nonstationarity with growing expert ensembles. *arXiv preprint arXiv:1103.0949*, 2011.

Harit Vishwakarma, Heguang Lin, and Ramya Korlakai Vinayak. Taming false positives in out-of-distribution detection with human feedback. In *Proc. AISTATS*, 2024.

Vladimir G Vovk. A game of prediction with expert advice. In *Proc. COLT*, pages 51–60, 1995.

Yi-Shan Wu, Yi-Te Hong, and Chi-Jen Lu. Lifelong learning with branching experts. In *Proc. ACML*, pages 1161–1175. PMLR, 2021.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, pages 1–28, 2024.

A PROOF OF LEMMA 4.2

We first prove the result for one dimension and then extend the result for the d dimension by establishing the independence of the coordinates of the points in each dimension.

For $d = 1$, the points belong to a closed interval on the real line. We have a strict inequality since X_t is drawn from a continuous i.i.d. distribution. Hence for any two permutations $X_{j_1}, X_{j_2}, \ldots, X_{j_t}$ and $X_{k_1}, X_{k_2}, \ldots, X_{k_t}$ of the sequence $\mathbf{X}_t$, we have

$$\mathbb{P}(X_{j_1} < X_{j_2} < \ldots < X_{j_t}) = \mathbb{P}(X_{k_1} < X_{k_2} < \cdots < X_{k_t}).$$

Since the events $\{X_{j_1} < X_{j_2} < \cdots < X_{j_t}\}$ are mutually exclusive and there are $t!$ possible permutations, we have

$$\sum_{\{j_1, j_2, \ldots, j_n\}} \mathbb{P}\left(X_{j_1} < X_{j_2} < \cdots < X_{j_t}\right) = 1$$
$$\implies \mathbb{P}\left(X_{j_1} < X_{j_2} < \cdots < X_{j_t}\right) = \frac{1}{t!}. \tag{21}$$

Given any realization of the sequence $\mathbf{X}_{t-1}$, for some permutation of $j_1, j_2, \ldots j_{t-1}$ we have $X_{j_1} < X_{j_2} < \cdots < X_{j_{t-1}}$. Let expert i be the kth interval $(X_{j_{k-1}}, X_{j_k})$, then the event $\{X_t \in \text{expert } i\}$ is equivalent to $\{X_t \in (X_{j_{k-1}}, X_{j_k})\}$, i.e., X_t is the kth highest value in the realization $\{\mathbf{X}_{t-1}, X_t\}$. Therefore, we have

$$\begin{aligned}
&\mathbb{P}\left(X_t \in \text{ expert } i \,\middle|\, X_{j_1} < X_{j_2} < \cdots < X_{j_{t-1}}\right) \\
=&\mathbb{P}\left(X_t \in (X_{j_{k-1}}, X_{j_k}) \,\middle|\, X_{j_1} < X_{j_2} < \cdots < X_{j_{t-1}}\right) \\
=&\frac{\mathbb{P}\left(X_{j_1} < \cdots < X_{j_{k-1}} < X_t < X_{j_k} < \cdots < X_{j_{t-1}}\right)}{\mathbb{P}\left(X_{j_1} < X_{j_2} < \cdots < X_{j_{t-1}}\right)} \\
=&\frac{\frac{1}{t!}}{\frac{1}{(t-1)!}} = \frac{1}{t}.
\end{aligned}$$

Note that the conditional probability is independent of k and thus it is true for any expert i. Finally, using total probability law over the permutations $j_1, j_2, \ldots, j_{t-1}$, we obtain $\mathbb{P}(X_t \in \text{ expert } i) = 1/t$, for all i.

For $d > 1$, let $X_t = (Z_t^1, \ldots, Z_t^d)$, where Z_t^r is the Euclidean coordinate of point X_t in r^{th} dimension.

Claim: Z_t^r are i.i.d. across t and r.

From the above claim and from (21), for any permutation $k_1, k_2, \ldots, k_{t-1}$ in dimension r, we obtain

$$\mathbb{P}\left(Z_{k_1}^r < Z_{k_2}^r < \cdots < Z_{k_t}^r\right) = \frac{1}{t!}$$
$$\implies \mathbb{P}\begin{pmatrix}\{Z_{m_1}^1 < Z_{m_2}^1 < \cdots < Z_{m_t}^1\}, \ldots, \\ \{Z_{j_1}^d < Z_{j_2}^d < \cdots < Z_{j_t}^d\}\end{pmatrix} = \frac{1}{(t!)^d}.$$

Again, given any realization of $\mathbf{X}_{t-1}$, the event $\{X_t \in \text{ expert } i\}$ is equivalent to $\{Z_t^r \in (Z_{j_{k-1}}^r, Z_{j_k}^r)\}$ for some permutation $j_1^r, j_2^r, \ldots, j_{t-1}^r$ in each dimension r.

$$\begin{aligned}
&\mathbb{P}\left(X_t \in \text{ expert } i \,\middle|\, \begin{matrix}\{Z_{m_1}^1 < Z_{m_2}^1 < \cdots < Z_{m_{t-1}}^1\}, \ldots, \\ \{Z_{j_1}^d < Z_{j_2}^d < \cdots < Z_{j_{t-1}}^d\}\end{matrix}\right) \\
=&\frac{\mathbb{P}\begin{pmatrix}\{Z_{m_1}^1 < \cdots < Z_{m_{k-1}}^1 < Z_t^1 < Z_{m_k}^1 < \cdots < Z_{j_{t-1}}^1\}, \ldots \\ \{Z_{j_1}^d < \cdots < Z_{j_{k-1}}^d < Z_t^d < Z_{j_k}^d < \cdots < Z_{j_{t-1}}^d\}\end{pmatrix}}{\mathbb{P}\begin{pmatrix}\{Z_{m_1}^1 < Z_{m_2}^1 < \cdots < Z_{m_{t-1}}^1\}, \ldots, \\ \{Z_{j_1}^d < Z_{j_2}^d < \cdots < Z_{j_{t-1}}^d\}\end{pmatrix}} \\
=&\frac{\frac{1}{(t!)^d}}{\frac{1}{[(t-1)!]^d}} = \frac{1}{t^d}.
\end{aligned}$$

B PROOF OF LEMMA 4.3

Let $\phi_i = \frac{e^{-\eta L_{t-1}(i)}}{\sum_{j \in \mathcal{B}_{t-1}} e^{-\eta L_{t-1}(j)}}$.

For any given sequence of arrivals $\mathbf{X}_{t-1}$, Y_t takes t^d possible values, each corresponding to X_t belonging to one of the t^d partitions. From Lemma 4.2, the latter event has probability $1/t^d$. For $i, j \in \mathcal{B}_{t-1}$, let $c_j(i)$ denote the number of partitions of expert i caused by sampling X_t from expert j, and let $C_i = \sum_{j \in \mathcal{B}_{t-1}} c_j(i)$. We compute the expectation of

$$\mathbb{E}[Y_t] = \sum_{(m_1,\ldots,m_t)} \cdots \sum_{(j_1,\ldots,j_t)} \mathbb{E}\left[Y_t \mid \begin{matrix}\{Z^1_{m_1} < Z^1_{m_2} < \cdots < Z^1_{m_t}\}, \ldots, \\ \{Z^d_{j_1} < Z^d_{j_2} < \cdots < Z^d_{j_t}\}\end{matrix}\right] \mathbb{P}\left(\begin{matrix}\{Z^1_{m_1} < Z^1_{m_2} < \cdots < Z^1_{m_t}\}, \ldots, \\ \{Z^d_{j_1} < Z^d_{j_2} < \cdots < Z^d_{j_t}\}\end{matrix}\right)$$

We have for every ordering (along every d dimension)

$$\mathbb{P}\left(\begin{matrix}\{Z^1_{m_1} < Z^1_{m_2} < \cdots < Z^1_{m_t}\}, \ldots, \\ \{Z^d_{j_1} < Z^d_{j_2} < \cdots < Z^d_{j_t}\}\end{matrix}\right) = \frac{1}{(t!)^d}.$$

$$\mathbb{P}\left(X_t \in \text{ expert } j \mid \begin{matrix}\{Z^1_{m_1} < Z^1_{m_2} < \cdots < Z^1_{m_{t-1}}\}, \ldots, \\ \{Z^d_{j_1} < Z^d_{j_2} < \cdots < Z^d_{j_{t-1}}\}\end{matrix}\right) = \frac{1}{t^d} \quad \forall j \in \mathcal{B}_{t-1} \tag{22}$$

It follows that

$$\begin{aligned}
&\mathbb{E}\left[Y_t \mid \begin{matrix}\{Z^1_{m_1} < Z^1_{m_2} < \cdots < Z^1_{m_{t-1}}\}, \ldots, \\ \{Z^d_{j_1} < Z^d_{j_2} < \cdots < Z^d_{j_{t-1}}\}\end{matrix}\right] \\
&= \sum_{j \in \mathcal{B}_{t-1}} \mathbb{P}\left(X_t \in \text{ expert } j \mid \begin{matrix}\{Z^1_{m_1} < Z^1_{m_2} < \cdots < Z^1_{m_{t-1}}\}, \ldots, \\ \{Z^d_{j_1} < Z^d_{j_2} < \cdots < Z^d_{j_{t-1}}\}\end{matrix}\right) \sum_{i \in \mathcal{B}_{t-1}} c_j(i)\phi(i) \\
&= \frac{1}{t^d} \sum_{j \in \mathcal{B}_{t-1}} \sum_{i \in \mathcal{B}_{t-1}} c_j(i)\phi(i) \qquad (23) \\
&= \frac{1}{t^d} \sum_{i \in \mathcal{B}_{t-1}} \phi(i) \sum_{j \in \mathcal{B}_{t-1}} c_j(i) \\
&= \frac{1}{t^d} \sum_{i \in \mathcal{B}_{t-1}} \phi(i) C_i \leq \frac{2^d}{t} \sum_{i \in \mathcal{B}_{t-1}} \phi(i) \leq \frac{2^d}{t}. \qquad (24)
\end{aligned}$$

We note that (23) follows from (22). In the upper bound in (24), we have used $C_i = (t+1)^d - t^d$ (derived below) and the following inequality:

$$(1+x)^r \leq 1 + (2^r - 1)x; \;\; x \in [0,1] \text{ and } r \in \mathbb{R} \setminus (0,1).$$

Note that C_i is the total number of partitions of expert i created due to sampling X_t from all t^d experts. We compute C_i using the following counting argument. We say an expert i *shares* k hyperplanes with expert j if, for any point in j, exactly k out of the d orthogonal hyperplanes (parallel to the faces of $\mathbb{B}$) that pass through that point will partition expert i. We compute the number of experts that share exactly k hyperplanes with i as follows. Choose any k dimensions from d in $\binom{d}{k}$ possible ways. Further, choose any orthogonal hyperplane passing through i that is parallel to some dimension from the rest of $d-k$ dimensions. There will be $t-1$ basis hyperplanes, i.e., the hyperplanes that partitioned $\mathbb{B}$ by passing through $t-1$ points drawn by the environment, that are parallel to the chosen hyperplane and do not partition i. The $(t-1)^{d-k}$ partitions, which are formed by the intersection of the $t-1$ basis hyperplanes corresponding to each of the $d-k$ dimensions, do not share exactly $d-k$ hyperplanes with i, or they share exactly k hyperplanes with i. Therefore, the total number of experts that share exactly k hyperplanes with i is $\binom{d}{k}(t-1)^{d-k}$, and each point drawn from those experts will result in 2^k partitions of expert i. Since index i will be assigned to one of its children (sub-partitions), we have $2^k - 1$ new experts from partitioning i. It follows that

$$C_i = \sum_{k=1}^{d} \binom{d}{k} (2^k - 1)(t-1)^{d-k}$$
$$= (t-1)^d \sum_{k=1}^{d} \binom{d}{k} \left(\frac{2}{t-1}\right)^k - \sum_{k=1}^{d} \binom{d}{k} (t-1)^{d-k}$$
$$= (t-1)^d \left(\frac{t+1}{t-1}\right)^d - t^d = (t+1)^d - t^d.$$

Indeed C_i is independent of i and is equal to the total number of new experts revealed in slot t. From (24) it follows that

$$\sum_{t=1}^{T} \mathbb{E}[Y_t] \leq \sum_{t=1}^{T} \left(\frac{2^d}{t}\right) \leq 2^d \left(\log T + 1\right).$$

C PROOF OF THEOREM 5.2

As above Let $\phi_i = \frac{e^{-\eta L_{t-1}(i)}}{\sum_{j \in \mathcal{B}_{t-1}} e^{-\eta L_{t-1}(j)}}$. Note that $e^{L_{t-1}(i) - L_{t-1}(j)} \geq 0$ for all i, j, and $L_{t-1}(i) \geq L_{t-1}(j)$ implies $e^{L_{t-1}(i) - L_{t-1}(j)} \geq 1$. Therefore,

$$\phi_i = \frac{1}{\sum_{j: L_{t-1}(j) > L_{t-1}(i)} e^{L_{t-1}(n_t) - L_{t-1}(j)} + \sum_{j: L_{t-1}(j) \leq L_{t-1}(i)} e^{L_{t-1}(i) - L_{t-1}(j)}}$$
$$\leq \frac{1}{\sum_{j \in \mathcal{B}_{t-1}} \mathbb{1}_{\{L_{t-1}(j) \leq L_{t-1}(i)\}}} = \delta_i.$$

Note that, for each $i \in \mathcal{B}_{t-1}$, δ_i takes a unique value from $\{1, \frac{1}{2}, \ldots, \frac{1}{t^d}\}$. Specifically, $\delta_i = \frac{1}{k}$, if expert i has the kth highest cumulative loss. Thus, δ_i only depends on the relation between the cumulative losses but not on their values. In the following, we use $[k]$ to denote the expert with k^{th} highest cumulative loss. We have

$$Y_t = \frac{\sum_{i=n_{t-1}+1}^{n_t} w_t(i)}{W_t} = \frac{\sum_{i=n_{t-1}+1}^{n_t} e^{-\eta L_{t-1}(i)}}{\sum_{j \in \mathcal{B}_{t-1}} e^{-\eta L_{t-1}(j)}}.$$

Note that $e^{\eta(L_{t-1}(i) - L_{t-1}(j))} \geq 0$ for all i, j, and $L_{t-1}(i) \geq L_{t-1}(j)$ implies $e^{\eta(L_{t-1}(i) - L_{t-1}(j))} \geq 1$, since $\eta > 0$. Therefore,

$$Y_t = \sum_{i=n_{t-1}+1}^{n_t} \frac{1}{\sum\limits_{j: L_{t-1}(j) > L_{t-1}(i)} e^{\eta(L_{t-1}(i) - L_{t-1}(j))} + \sum\limits_{j: L_{t-1}(j) \leq L_{t-1}(i)} e^{\eta(L_{t-1}(i) - L_{t-1}(j))}}$$
$$\leq \sum_{i=n_{t-1}+1}^{n_t} \frac{1}{\sum_{j \in \mathcal{B}_{t-1}} \mathbb{1}_{\{L_{t-1}(j) \leq L_{t-1}(i)\}}}. \tag{25}$$

In round t, we define a random variable Z_t such that $Z_t = j^{-1}$, if X_t falls in the j^{th} best expert, i.e., if $\sum_{j \in \mathcal{B}_{t-1}} \mathbb{1}_{\{L_{t-1}(j) \leq L_{t-1}(i)\}} = j$. From (25), we have $Y_t \leq Z_t$, for all t. Further, we have

$$\mathbb{P}\left(\sum_{t=1}^{T} Y_t - \sum_{t=1}^{T} \mathbb{E}[Y_t] > \delta\right) \leq \mathbb{P}\left(\sum_{t=1}^{T} Z_t - \sum_{t=1}^{T} \mathbb{E}[Y_t] > \delta\right)$$
$$\leq \mathbb{P}\left(\sum_{t=1}^{T} Z_t - \sum_{t=1}^{T} \mathbb{E}[Z_t] > \delta - \sum_{t=1}^{T} \mathbb{E}[Z_t] + \sum_{t=1}^{T} \mathbb{E}\left[Y_t\right]\right).$$

Using the same argument as the one used for Y_t, we have

$$\begin{aligned}
\mathbb{E}[Z_t] &= \frac{1}{t^d}\sum_{j\in\mathcal{B}_{t-1}}\sum_{i\in\mathcal{B}_{t-1}} c_j(i)\delta(i) \\
&= \frac{1}{t^d}\sum_{i\in\mathcal{B}_{t-1}}\delta(i)\sum_{j\in\mathcal{B}_{t-1}} c_j(i) \\
&= \frac{1}{t^d}\sum_{i\in\mathcal{B}_{t-1}}\delta(i)C_i \le \frac{2^d}{t}\sum_{i\in\mathcal{B}_{t-1}}\delta(i) \le \frac{2^d}{t}d\left(1+\log t\right).
\end{aligned} \tag{26}$$

Further using similar manipulations as Y_t, we get

$$\begin{aligned}
&\mathbb{E}\left[Z_t^2 \;\middle|\; \begin{matrix}\{Z^1_{m_1} < Z^1_{m_2} < \cdots < Z^1_{m_{t-1}}\},\ldots,\\ \{Z^d_{j_1} < Z^d_{j_2} < \cdots < Z^d_{j_{t-1}}\}\end{matrix}\right] \\
&= \sum_{j\in\mathcal{B}_{t-1}}\mathbb{P}\left(X_t \in \text{ expert } j \;\middle|\; \begin{matrix}\{Z^1_{m_1} < Z^1_{m_2} < \cdots < Z^1_{m_{t-1}}\},\ldots,\\ \{Z^d_{j_1} < Z^d_{j_2} < \cdots < Z^d_{j_{t-1}}\}\end{matrix}\right)\left(\sum_{i\in\mathcal{B}_{t-1}} c_j(i)\delta(i)\right)^2 \\
&= \frac{1}{t^d}\sum_{j\in\mathcal{B}_{t-1}}\left(\sum_{i\in\mathcal{B}_{t-1}} c_j(i)\delta(i)\right)^2 \\
&= \frac{1}{t^d}\sum_{j\in\mathcal{B}_{t-1}}\sum_{i\in\mathcal{B}_{t-1}} c_j(i)^2\delta(i)^2 + \frac{1}{t^d}\sum_{j\in\mathcal{B}_{t-1}}\sum_{i\in\mathcal{B}_{t-1}}\sum_{k\in\mathcal{B}_{t-1}} 2c_j(i)c_j(k)\delta(i)\delta(k) \\
&\le \frac{1}{t^d}\sum_{i\in\mathcal{B}_{t-1}}\delta(i)^2\sum_{j\in\mathcal{B}_{t-1}} c_j(i)^2 + \frac{1}{t^d}\sum_{j\in\mathcal{B}_{t-1}}\sum_{i\in\mathcal{B}_{t-1}}\sum_{k\in\mathcal{B}_{t-1}} 2c_j(i)c_j(k)\delta(i)\delta(k).
\end{aligned} \tag{27}$$

Using the logic to compute C_i where 2^k-1 are the new experts being formed, we have the following:

$$\begin{aligned}
D_i = \sum_{j\in\mathcal{B}_{t-1}} c_j(i)^2 &= \sum_{k=1}^{d}\binom{d}{k}(2^k-1)^2(t-1)^{d-k} \\
&= \sum_{k=1}^{d}\binom{d}{k}(2^{2k}-2\cdot 2^k+1)^2(t-1)^{d-k} \\
&= \sum_{k=1}^{d}\binom{d}{k}4^k(t-1)^{d-k} - 2\sum_{k=1}^{d}\binom{d}{k}2^k(t-1)^{d-k} + \sum_{k=1}^{d}\binom{d}{k}(t-1)^{d-k} \\
&= (t+3)^d - (t+1)^d - ((t+1)^d - t^d) \\
&= (t+3)^d - (t+1)^d.
\end{aligned}$$

Indeed D_i is independent of i as all the experts are split with equal probability.

Simplifying (27) using the fact $\sum_{i=1}^{t^d}\delta(i)^2 = \sum_{i=1}^{t^d}\frac{1}{i^2} \le \frac{\pi^2}{6}$, we get

$$\frac{1}{t^d}\sum_{i\in\mathcal{B}_{t-1}}\delta(i)^2 D_i \le \frac{(t+3)^d - t^d}{t^d}\cdot\frac{\pi^2}{6} \le \frac{5\cdot 2^d}{t}. \tag{28}$$

The second term in (27) can be simplified as follows:

$$\begin{aligned}
\frac{1}{t^d}\sum_{j\in\mathcal{B}_{t-1}}\sum_{i\in\mathcal{B}_{t-1}}\sum_{k\in\mathcal{B}_{t-1}} 2c_j(i)c_j(k)\delta(i)\delta(k) &\le \frac{1}{t^d}\sum_{j\in\mathcal{B}_{t-1}}\sum_{i\in\mathcal{B}_{t-1}}\sum_{k\in\mathcal{B}_{t-1}} 2^{d+1}c_j(i)\delta(i)\delta(k) &(29)\\
&\le \frac{2^{d+1}\left(d\log t+1\right)}{t^d}\sum_{j\in\mathcal{B}_{t-1}}\sum_{i\in\mathcal{B}_{t-1}} c_j(i)\delta(i) &(30)\\
&\le \frac{2^{2d+1}\left(d\log t+1\right)^2}{t}, &(31)
\end{aligned}$$

where (29) follows as $c_j(k) \leq 2^d$ and (30) follows using the fact that $\sum_{k \in \mathcal{B}_{t-1}} \delta(k) \leq (\log t^d + 1)$. Further, (31) follows from (26). Substituting (28) and (31) in (27), we get

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}[Z_t^2] &\leq \sum_{t=1}^{T} \frac{5 \cdot 2^d}{t} + \frac{2^{2d+1} \left(d \log t + 1\right)^2}{t} \\ &\leq 5 \cdot 2^d \left(1 + \log T\right) + d^2 \cdot 2^{2d+3} \sum_{t=1}^{T} \frac{\log t^2}{t} \\ &\leq 5 \cdot 2^d \left(1 + \log T\right) + d^2 \cdot 2^{2d+3} \left(1 + \log T^3\right).\end{aligned}$$

To get (18), we use (16), (17), and the fact that $\sum_{t=1}^{T} \mathbb{E}\left[Y_t\right] \geq \log T$. Since the Z_ts are independent and are upper bounded by one, using Bernstein's inequality, we get

$$\mathbb{P}\left(\sum_{t=1}^{T} Z_t - \sum_{t=1}^{T} \mathbb{E}[Z_t] > \delta'\right) \leq e^{-\frac{\delta'^2/2}{V_n + \delta'/3}}, \tag{32}$$

where $V_n = \sum_{t=1}^{T} \mathrm{Var}(Z_t)$, and $\delta' = \delta - \sum_{t=1}^{T} \mathbb{E}[Z_t] + \sum_{t=1}^{T} \mathbb{E}[Y_t] \geq \delta - c_1 \log T^2$ for some constant c_1 depending on dimension d. We also have $\sum_{t=1}^{T} \mathrm{Var}(Z_t) \leq c_2 \log T^3$ for some constant c_2 depending on dimension d.
Choosing $\delta' = c_3 \log T^2$ results in $\delta = O(\log T^3)$. Substituting δ' and $\sum_{t=1}^{T} \mathrm{Var}(Z_t) \leq \frac{\pi^2}{6}(\log T + 1)$ in (32),

$$\mathbb{P}\left(\sum_{i=1}^{T} Y_i - \sum_{i=1}^{T} \mathbb{E}[Y_i] > \delta\right) \leq e^{-c \log T} = O\left(\frac{1}{T^c}\right).$$