Optimization of Offloading Policies for Accuracy-Delay Tradeoffs in Hierarchical Inference

Hasan Burhan Beytur, Ahmet Gunhan Aydin, Gustavo de Veciana and Haris Vikalo
Electrical and Computer Engineering Department, The University of Texas at Austin
{hbbeytur, ahmetgunhanaydin, deveciana}@utexas.edu, hvikalo@ece.utexas.edu

***Abstract*—We consider a hierarchical inference system with multiple clients connected to a server via a shared communication resource. When necessary, clients with low-accuracy machine learning models can offload classification tasks to a server for processing on a high-accuracy model. We propose a distributed online offloading algorithm which maximizes the accuracy subject to a shared resource utilization constraint thus indirectly realizing accuracy-delay tradeoffs possible given an underlying network scheduler. The proposed algorithm, named Lyapunov-EXP4, introduces a loss structure based on Lyapunov-drift minimization techniques to the bandits with expert advice framework. We prove that the algorithm converges to a near-optimal threshold policy on the confidence of the clients' local inference without prior knowledge of the system's statistics and efficiently solves a constrained bandit problem with sublinear regret. We further consider settings where clients may employ multiple thresholds, allowing more aggressive optimization of overall accuracy at a possible loss in fairness. Extensive simulation results on real and synthetic data demonstrate convergence of Lyapunov-EXP4, and show the accuracy-delay-fairness tradeoffs achievable in such systems.**

***Index Terms*—Lyapunov optimization, online learning, hierarchical inference, computation offloading**

I. Introduction

Machine learning (ML) applications and services are evolving to enable finding solutions to increasingly more challenging problems by deploying computationally intense models [1], [2]. This presents a challenge to their widespread deployment, especially on platforms with limited power and computational capabilities such as smartphones and IoT devices.

At one extreme, commonly seen in many real-world ML based applications [3], the entire ML model is executed on a server with sufficient computational power allowing devices to draw on minimal local computation. Although this approach relieves the computational burden on the users' devices, it leads to increased communication costs, reduced responsiveness, and may bring up privacy concerns.

At the other extreme, one can run ML based applications solely on users' devices. To facilitate this, there has been significant research on techniques for compressing complex ML models, resulting in what is commonly referred to as tinyML [4]. While this allows running models with small computational and memory footprints, they often come at the cost of sacrificing performance and accuracy as compared to the larger complex ML models typically run at the edge/cloud.

This material is based upon work supported by the National Science Foundation (NSF) under grant CNS-2212202 and grant No. 2148224 and is supported in part by funds from OUSD R&E, NIST, and industry partners as specified in the Resilient & Intelligent NextG Systems (RINGS) program.

Between these extremes is Hierarchical Inference, a framework aiming to enable combining execution of ML tasks on users' devices with computational offloading to a server [5]–[8]. In particular, users' devices employ a low-complexity ML model (L-ML) while the server hosts a more powerful high-complexity ML model (H-ML). A task is first processed by L-ML on the user's device. If the result requires further investigation or is deemed to have low confidence, the task is offloaded to the server to be processed by H-ML. Unlike DNN Partitioning [9], [10], Hierarchical Inference does not (partially) offload every task to the server, which helps improve the system's responsiveness. Although a hierarchical inference system can be built based on any pair of L-ML and H-ML, or with an ML model allowing early exit [11], it requires an offloading policy based on the output of the L-ML decides if offloading is needed. Furthermore, such decisions might depend on congestion of the communication resources.

Our work is focused on hierarchical inference systems with multiple user devices and random offloading costs. We propose an online algorithm that learns how to manage offloading in such settings while being oblivious to the statistics of ML models, input task and offloading cost distributions. The proposed algorithm utilizes a multi-armed bandit with expert advice framework [12] to maximize system's inference accuracy while deploying a novel loss structure based on Lyapunov optimization theory to meet a constraint on resource utilization or, more generally, on offloading costs. The algorithm converges to an optimal threshold policy applied to the inference confidence seen on the L-ML model.

A. Related Work

Recently, hierarchical inference systems have attracted significant attention from the research community [5]–[8]. An important line of this research focuses on designing metrics that capture the confidence of the local inference without knowing the correctness of the result. Examples include [6], where a confidence metric based on the variance of the local classifier's output is proposed, and [7], where the confidence of a prediction is assessed by applying a radial function kernel to the entropy of the local classifier's output. Ultimately, confidence metrics are meant to aid in designing offloading algorithms for hierarchical inference systems such as the one in [8], which deploys multiple local ML models on devices and a more complex/accurate ML model on a server. Given the different accuracy, processing and computation times of the models, the accuracy maximization problem under a time

constraint is formulated as an integer linear program. An online learning approach proposed in [5] seeks a threshold policy on the confidence of L-ML that maximizes the accuracy under a fixed offloading cost. Although our work has a similar goal as [5], it is distinct as it focuses on multiple users under a shared utilization constraint and the distributed online offloading algorithm.

The long-term stochastic optimization problems encountered in various computation offloading problem, are studied using Lyapunov optimization techniques, specifically Drift-plus-Penalty algorithm [13]. The Drift-plus-Penalty formulation is used in [14], [15] to solve offloading problems, where [14] investigates a mobile-edge computing system with energy harvesting devices under a hard execution delay deadline, and [15] formalizes the joint service caching and task offloading problem for minimizing computation latency under a long-term energy consumption constraint. Similarly, in [16], [17] authors present long-term optimization problems as two-step optimization problems and leverage Drift-plus-Penalty algorithm to solve one of the steps. In all these works, the Lyapunov formulation involves an integer problem and the solution requires knowing the system parameters at each slot.

In another line of related recent work, Lyapunov optimization techniques have been combined with online learning mechanisms to develop algorithms addressing constrained bandit optimization problems [18]–[23]. Specifically, [18], [19], utilize Lyapunov-drift minimization methods for constrained online-dispatching via the UCB algorithm. In [20], a generic constrained bandit problem which incorporates random costs of actions, a knapsack constraint, and a stochastic feasibility constraint was studied; there, a Lyapunov-drift minimization technique was utilized to design a low-complexity bandit algorithm based on UCB. Our work addresses constrained contextual bandit problems by combining Lyapunov drift minimization techniques with EXP4 algorithm.

B. Contributions

The contributions of this paper are summarized as follows.

- For a hierarchical inference system supporting multiple clients, we formulate an accuracy maximization problem under a shared resource utilization constraint and explore the associated accuracy-delay trade-offs. We combine Lyapunov drift minimization techniques and bandits with expert advice framework to propose a low-complexity online offloading algorithm that we refer to as Lyapunov-EXP4 (Ly-EXP4). Ly-EXP4 converges to a near-optimal thresholding policy on the local inference confidence while satisfying the utilization constraint without prior knowledge of the statistics of tasks, confidence metric, and service times across clients. We prove that for finite horizon N and M discretized thresholds, Ly-EXP4 has a regret bound of $O(\sqrt{2N\zeta\log(M)})$ and an optimality-gap of $O(\frac{1}{V})+\epsilon_0$, where V is a parameter controlling tradeoff between accuracy and constraint violation. To the best of our knowledge, our work is the first one in the literature discussing a hierarchical inference problem with multiple clients under a utilization constraint.
- We show that Ly-EXP4 can be used as a distributed online offloading algorithm maximizing the total accuracy of a system under a resource utilization constraint. In such settings, each client or a group of clients simultaneously learns an individual thresholding policy.
- Ly-EXP4 is not limited to the hierarchical inference problem; instead, it can be used in a more general class of constrained contextual bandit problems – a potentially valuable contribution in itself.
- We provide extensive simulation results using both real and synthetic ML models and datasets, demonstrating the performance of Ly-EXP4 in terms of convergence, delay and fairness. Additionally, we explore the accuracy-fairness trade-off in settings with multiple thresholds which allow Ly-EXP4 to achieve higher accuracy at the expense of fairness across clients.

II. System Model for the Hierarchical Inference

We consider a hierarchical inference system with a set $\mathcal{K}$ of clients $k \in \mathcal{K}$ connected to a server over a shared communication resource, executing classification tasks using a neural network based classifier. The classification tasks that the clients face are first processed locally using a small ML model with low complexity and low accuracy (L-ML); in addition to the prediction, L-ML quantifies the confidence in its decision. After comparing the confidence with a threshold, the client decides whether to offload the task to a server to be processed with a more complex ML classifier with higher accuracy (H-ML). However, shared communication resources limit the number of tasks that the clients can offload.

We assume that classification tasks arrive to a client according to a Poisson process. Let λ_k denote the arrival rate of tasks to client k, and let $\lambda = \sum_{k\in\mathcal{K}} \lambda_k$. The n^{th} task arriving to the system is received by client K_n at time T_n. Let us denote the true class of task n and the class predicted by the local L-ML by C_n and $\hat{C}_n$, respectively. We assume that the true class of a task received by client k is independent and identically distributed according to an unknown client-specific distribution, i.e., $\boldsymbol{\alpha}_k = (\alpha_k(c) : c \in \mathcal{C})$, where $\alpha_k(c)$ denotes the probability that a task arriving to client k is of class c, and $\mathcal{C}$ is the set of classes. Since true class distributions of tasks differ from one client to another, and the clients deploy potentially different L-ML models, the predicted class distributions generally vary across the clients.

An L-ML model locally processing classification task n provides predicted class $\hat{C}_n$ and confidence measure Z_n. Given C_n and $\hat{C}_n$, Z_n is assumed to be independent and identically distributed on a bounded support Ω_Z. As with C_n and $\hat{C}_n$, we do not assume a specific distribution for Z_n.

Since in our hierarchical inference system the offloading decisions are based on the local inference confidence, the quality of confidence metric is essential. Nevertheless, the offloading algorithm introduces in Section III converges to an optimal threshold policy without prior knowledge of the distribution of Z_n, regardless of the choice of metric.

When L-ML is a neural network used for classification, a simple choice for the confidence metric is the largest output of

the last soft-max layer; such a quantity must be at least $1/|\mathcal{C}|$ (when all class outputs are equal), and cannot exceed 1.

Next, we define the stationary threshold policy for the hierarchical inference.

Definition 1 (Stationary Threshold Policy)**.** *The offloading decision under a threshold policy π_θ with a threshold $\theta \in \Omega_Z$ is given by*

$$I_n^{\pi_\theta} = \mathbb{I}\{Z_n < \theta\}, \quad \theta \in \Omega_Z, \tag{1}$$

where $I_n^{\pi_\theta} = 1$ indicates offload of task n.

Offloading a task incurs resource expenditures, e.g., extra load on the communication channel or an operational cost of using the server. The offloading cost of task n is modelled by a random variable X_n; the offloading costs of the tasks received by client k are assumed to be independent and identically distributed according to an unknown client-specific distribution with a bounded support (without a loss of generality, [0,1]) and mean $\mu_k^{-1} = \mathbb{E}[X_n | K_n = k]$. We consider the case where the clients share a pool of resources, e.g., wireless access to a BS/server. The constraint on resource utilization is given by

$$\sum_{k \in \mathcal{K}} \hat{\lambda}_k^{\pi_\theta} \mu_k^{-1} \leq \gamma, \tag{2}$$

where $\hat{\lambda}_k^{\pi_\theta}$ is the offloading rate of client k under the stationary threshold policy π_θ. In practical systems with shared resources, the offloading cost can be thought of as the monetary value of using computational resources at the server, where γ can be interpreted as the budget per unit operational time. When it comes to communications, the mean offloading cost μ_k^{-1} can be thought of as the mean service time of client k, with γ being a utilization constraint which indirectly controls the average delay in the system. The latter interpretation of γ is due to an observation that in a network where a scheduling policy controls how multiple clients share communication resources, utilization can be treated as a proxy for average delay as the two quantities are monotonically related to each other.

Although our model and algorithm can be used in more general settings, in the remainder of this paper we focus on the accuracy-delay tradeoff in hierarchical inference systems where the task offloading follows a scheduling policy. Since the offloading rates $\hat{\lambda}_k^{\pi_\theta}$ are well-defined under the stationary threshold policy π_θ, we have that

$$\hat{\lambda}_k^{\pi_\theta} = \lim_{N \to \infty} \frac{1}{T_N} \sum_{n \in [N]: K_n = k} \mathbb{E}[I_n^{\pi_\theta}] \tag{3}$$

$$= \lim_{N \to \infty} \frac{\lambda}{N} \sum_{n \in [N]: K_n = k} \mathbb{E}[I_n^{\pi_\theta}]. \tag{4}$$

Once task n is received by client K_n at time T_n, the client processes the task using its L-ML model. Based on the confidence Z_n of the L-ML prediction, the client decides whether to send the task to the server to be classified by the H-ML model. We assume that the H-ML classifies the tasks perfectly, i.e., has 100% inference accuracy, and that the computation time by the client and the server is negligible compared to the communication time needed to offload the task. In addition, we assume that there is a feedback channel between the server and the clients, which is used to transmit parameters required by the algorithm (e.g., inference results and bandit loss of an offload, formally defined later in Section III-B). Since the transmitted data is relatively small, we assume that the delay on the feedback channel is negligible.

A. Problem Formulation

Given the shared communication resource and the hierarchical inference mechanism, we want to find an offloading policy that maximizes the overall inference accuracy of the system while satisfying the utilization constraint. By substituting (4) in (2), we write the inference accuracy maximization problem as the minimization

$$\min_{\theta \in \Omega_Z} \quad \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}[Y_n (1 - I_n^{\pi_\theta})] \tag{5}$$

$$\text{s.t.} \quad \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}[I_n^{\pi_\theta} X_n] \leq \frac{\gamma}{\lambda}, \tag{6}$$

where $Y_n = \mathbb{I}\{C_n \neq \hat{C}_n\}$, e.g., the indicator of an incorrect local classification. If X_n and Y_n were available at each time step before taking the offloading decision, the constrained optimization problem in (5) could be solved using the Lyapunov drift minimization technique referred to as the drift-plus-penalty algorithm [13]. However, due to the structure of the hierarchical inference system, X_n and Y_n are only revealed if task n is offloaded. Since they are observed in hindsight, we need a methodology that allows efficient exploration/exploitation of the system's unknown statistics. To this end, we adopt a constrained bandit framework.

Next, we introduce an algorithm based on Lyapunov-EXP4 that makes offloading decision based on a single threshold. In Section III-C, this algorithm is extended to the multiple thresholds case, enabling its distributed implementation.

III. Threshold Offloading Policy using Lyapunov-EXP4 (Ly-EXP4)

To solve the optimization problem (5), we propose a computationally efficient online learning algorithm Lyapunov-EXP4 (Ly-EXP4) based on the contextual bandits framework. Ly-EXP4 makes decisions using a loss structure that employs the Lyapunov drift minimization technique, maximizing accuracy while satisfying a long-term average expectation constraint.

A. Virtual Queue and Drift-plus-Penalty Ratio

In this subsection, we introduce a virtual queue capturing the constraint (6) and a drift-plus-penalty ratio motivated by [13], which are then utilized to specify the bandit loss. To start, we define the causal policy space for our problem.

Definition 2 (Causal Policy)**.** *Let π be a policy that yields a sequence of offloading decisions $\{I_n^\pi \in \{0,1\} : N \geq n \geq 1\}$. Under π, the history until time n is the filtration*

$$\mathcal{F}_n^\pi = \sigma\left(\{Z_{t+1}, I_t^\pi, X_t I_t^\pi, Y_t I_t^\pi : 1 \leq t \leq n\}\right), \tag{7}$$

where $\sigma(\cdot)$ denotes the sigma-field, and X_t and Y_t are observed only if task t is offloaded (as implied by the terms $X_t I_t^\pi$ and $Y_t I_t^\pi$). A policy π is said to be causal if π is non-anticipating, i.e., $\{I_n^\pi = a\} \in \mathcal{F}_{n-1}^\pi$ for all a, n.

We define the virtual queue under casual policy π as

$$Q_{n+1}^{\pi} = \max[Q_n^{\pi} + I_n^{\pi} X_n - \frac{\gamma}{\lambda}, 0], \tag{8}$$

where $Q_1^{\pi} = 0$. The time average expectation constraint (6) can be viewed as a mean rate stability constraint of a virtual queue [13] under causal policy π. To see how the mean rate stability of Q_n^{π} enforces (6), note that by the telescoping argument

$$\frac{Q_{N+1}^{\pi}}{N} - \frac{Q_1^{\pi}}{N} \geq \frac{1}{N} \sum_{n=1}^{N} I_n^{\pi} X_n - \frac{\gamma}{\lambda}. \tag{9}$$

Taking expectations and letting $N \to \infty$ one can show that

$$\lim_{N\to\infty} \frac{\mathbb{E}[Q_{N+1}^{\pi}]}{N} \geq \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}[I_n^{\pi} X_n] - \frac{\gamma}{\lambda}. \tag{10}$$

If Q_n^{π} is mean rate stable, i.e., $\lim_{N\to\infty} \frac{\mathbb{E}[Q_{N+1}^{\pi}]}{N} = 0$, then the constraint (6) is satisfied.

With a Lyapunov function $\mathcal{L}(Q_n) = \frac{1}{2} Q_n^2$, for any causal policy π we have

$$\begin{aligned}
&\mathcal{L}(Q_{n+1}^{\pi}) - \mathcal{L}(Q_n^{\pi}) \\
&\quad = \frac{1}{2} \left(\max[Q_n^{\pi} + I_n^{\pi} X_n - \frac{\gamma}{\lambda}, 0] \right)^2 - \frac{1}{2} Q_n^{\pi 2} \quad (11) \\
&\quad = \frac{1}{2} ((I_n^{\pi} X_n)^2 + (\frac{\gamma}{\lambda})^2) + Q_n^{\pi} I_n^{\pi} X_n - Q_n^{\pi} \frac{\gamma}{\lambda} \quad (12) \\
&\quad \leq B + Q_n^{\pi} I_n^{\pi} X_n - Q_n^{\pi} \frac{\gamma}{\lambda}, \quad (13)
\end{aligned}$$

where $B = \frac{1}{2}(1 + (\frac{\gamma}{\lambda})^2)$.

Let $\mathbb{E}_n[\cdot] = \mathbb{E}[\cdot|\mathcal{F}_n^{\pi}]$ denote the conditional expectation given the history up to time n. Denote the Lyapunov drift by $\Delta(Q_n^{\pi}) = \mathbb{E}_{n-1}[\mathcal{L}(Q_{n+1}^{\pi}) - \mathcal{L}(Q_n^{\pi})]$, where Q_n^{π} is measurable with respect to $\mathcal{F}_{n-1}^{\pi}$ for all n (π is causal). At each slot, the drift-plus-penalty algorithm opportunistically minimizes an upper bound on the drift (13) summed with the one-slot penalty, which in our setting leads to

$$\begin{aligned}
&\Delta(Q_n^{\pi}) + V \mathbb{E}_{n-1}[Y_n(1 - I_n^{\pi})] \\
&\leq B + Q_n^{\pi} \mathbb{E}_{n-1}[I_n^{\pi} X_n] + V \mathbb{E}_{n-1}[Y_n(1 - I_n^{\pi})] - Q_n^{\pi} \frac{\gamma}{\lambda}, \quad (14)
\end{aligned}$$

where $V > 0$ is the parameter controlling the trade-off between penalty and drift. Note that if X_n and Y_n were observed before the offloading decision of task n, i.e., if $X_n, Y_n \in \mathcal{F}_{n-1}^{\pi}$, the one-slot optimization problem the drift-plus-penalty algorithm tackles would become

$$\min_{I_n^{\pi} \in \{0,1\}} \quad Q_n^{\pi} I_n^{\pi} X_n / V + Y_n(1 - I_n^{\pi}). \tag{15}$$

The aforementioned bandit algorithm Ly-EXP4, formally presented in the next subsection, deploys (15) as the loss at time n and exhibits convergence to a near-optimal solution of (5).

B. Ly-EXP4: A Bandit with Expert Advice

In the bandits with expert advice framework, instead of learning an arm selection policy, the agent learns a way of combining experts' predictions that minimizes the regret. One common algorithm to tackle this problem is the EXP4 [12], which randomly selects arms according to a distribution computed by a soft-max operation on the cumulative loss each expert received.

We formulate hierarchical inference as a multi-arm-bandit problem with expert advice; the offloading decisions are denoted by $a \in \{0, 1\}$, where $a = 0$ corresponds to "not offloading" while $a = 1$ corresponds to "offloading". In addition, we assume that there are M experts, each corresponding to a threshold policy as in (1) with a threshold $\theta_m \in \mathcal{M},\ m \in [1, \ldots, M]$, where $\mathcal{M}$ is a set of discretized thresholds on the support of the confidence metric Ω_Z. Note that discretizing the thresholds induces a discretization error; while we do not assume a specific discretization scheme, we assume that the optimality gap between the best threshold in $\mathcal{M}$ and the optimal (continuous) solution to (5) is negligible.

Assumption 1. *Let $A_N(\pi)$ denote the accuracy at time N under policy π, i.e.,*

$$A_N(\pi) = \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}[Y_n(1 - I_n^{\pi})]. \tag{16}$$

Given a set of discretized thresholds $\mathcal{M}$, there exists at least one threshold $\theta_m \in \mathcal{M}$ such that

$$\lim_{N\to\infty} A_N(\pi_{\theta_m}) - A_N(\pi_{\theta^*}) \leq \epsilon_0, \tag{17}$$

where $\theta^* \in \Omega_Z$ is the optimal threshold for problem (5)*.*

A simple discretization scheme would be to select uniform discretization intervals, which ensures $\epsilon_0 \leq \frac{1}{M}$. Note that Assumption 1 enforces not only a condition on the discretization scheme but also a smoothness property on the distribution of the confidence metric Z_n.

Next, we define bandit loss based on (15); for task n, let $l_{n,a}$ denote the loss corresponding to decision $a \in \{0, 1\}$,

$$l_{n,a} = \begin{cases} Y_n & a = 0 \ \ \text{(do not offload)}, \\ \dfrac{Q_n X_n}{V} & a = 1 \ \ \text{(offload)}. \end{cases} \tag{18}$$

Note that such a loss is a combination of the minimization objective and the utilization constraint. Based on (18), the loss associated with the offloading decisions taken by the algorithm, L_n, and the loss associated with the m^{th} expert, $L_{n,m}$, can be written as

$$L_n = l_{n,I_n}, \qquad L_{n,m} = l_{n,I_{n,m}}, \quad \forall n, m, \tag{19}$$

where $I_{n,m} = \mathbb{I}\{\theta_m > Z_n\}$ is the offloading decision of expert m for the task n. Since the virtual queue process Q_n is driven by the decisions Ly-EXP4 makes, expert loss $L_{n,m}$ depends not only on the expert's own decisions $I_{n,m}$ but also on the previous decisions of Ly-EXP4.

The regret that we want to minimize is defined relative to the best expert in hindsight,

$$R_N = \mathbb{E}\left[\sum_{n=1}^{N} L_n - \min_{m \in \mathcal{M}} \sum_{n=1}^{N} L_{n,m} \right]. \tag{20}$$

The regret R_N, loss L_n and expert loss $L_{n,m}$ are all incurred under the Ly-EXP4 policy. However, for notational simplicity, we drop superscript π_{LyEXP4} from these expressions.

Ly-EXP4 acts in this bandit setting similarly to EXP4. Depending on Z_n, each expert incurs a loss due to either incorrect local inference or offloading cost. Based on the experts' cumulative loss, Ly-EXP4 computes normalized weight for each expert via soft-max, where a weight represents "contribution" of an expert to the stochastic offloading decision.

Since X_n and Y_n are observed only if a task is offloaded, experts' losses are not always known. To overcome this problem, we employ an importance-weighted estimator of the experts' loss, $\hat{L}_{n,m}$, as defined below. Let $\hat{S}_{n,m} = \sum_{t=1}^{n} \hat{L}_{t,m}$ denote the cumulative estimated loss for expert m until time n. The normalized weight of expert m at time n is defined as

$$\omega_{n,m} = \frac{\exp\left(-\eta \hat{S}_{n-1,m}\right)}{\sum_{m \in \mathcal{M}} \exp\left(-\eta \hat{S}_{n-1,m}\right)}. \tag{21}$$

It follows from (21) that an expert with a higher loss will be assigned a smaller weight. At time n, Ly-EXP4 takes a random offloading decision with probability p_n equal to the sum of the weights of the experts advising to offload based on the observed confidence Z_n,

$$p_n = \sum_{m:\theta_m > Z_n} \omega_{n,m}. \tag{22}$$

Note that this procedure is effectively as same as sampling an expert m at random from distribution $\boldsymbol{\omega}_n = (\omega_{n,m} : m \in \mathcal{M})$ and following the offloading decision of the selected expert, where the experts with lower cumulative loss have higher probability of being chosen. Since p_n depends on the losses received until time $(n-1)$, we define the importance-weighted loss estimate as

$$\hat{L}_{n,m} = \begin{cases} 0, & I_n = 0 \\ \dfrac{Y_n}{p_n}, & I_n = 1 \text{ and } I_{n,m} = 0 \\ \dfrac{Q_n X_n}{V p_n}, & I_n = 1 \text{ and } I_{n,m} = 1 \end{cases}, \quad \forall n, m. \tag{23}$$

If task n is offloaded, the experts incur a loss according to their own offloading decisions $I_{n,m}$; if task n is not offloaded, experts incur zero loss. The next lemma establishes that $\hat{L}_{n,m}$ is an unbiased estimator of $l_{n,I_{n,m}}$.

Lemma 1. *Since $\mathbb{E}_{n-1}[\mathbb{I}\{I_n = 1\}] = p_n$ and p_n is $\mathcal{F}_{n-1}$-measurable, $L_{n,m}$ is an unbiased estimator of $l_{n,I_{n,m}}$, i.e.,*

$$\mathbb{E}_{n-1}[\hat{L}_{n,m}] = l_{n,I_{n,m}}. \tag{24}$$

The proof of Lemma 1 is given in Appendix A.

In contrast to EXP4, the offloading decisions in Ly-EXP4 affect the loss in subsequent steps through the virtual queue Q_n, representing the constraint violation. As a consequence of the adopted loss function, when the resource utilization grows the weights of the experts in favor of offloading become smaller. Similarly, when the resource utilization is low, the algorithm tends to offload more due to decreased offloading loss. Ly-EXP4 is formalized as Algorithm 1. In what follows, we prove finite-horizon regret bounds for Ly-EXP4.

Proposition 1 (Ly-EXP4 Regret Analysis)**.** *Given the number of experts M, learning rate $\eta > 0$, and $V > 0$, the regret under Ly-EXP4 satisfies*

$$R_N \leq R_{N,m} \leq \frac{\log(M)}{\eta} + \frac{\eta N}{2}\zeta, \quad \forall m \in \mathcal{M}, \tag{25}$$

where $\zeta > 2$. Specifically, for $\eta = \sqrt{\frac{2\log(M)}{N\zeta}}$ the regret is

$$R_N = O\left(\sqrt{2N\zeta \log(M)}\right). \tag{26}$$

The proof is outlined in Appendix B.

Note that Ly-EXP4 converges to the threshold minimizing the regret based on the loss formed as a combination of the objective and the problem constraint. Therefore, the best expert's policy Ly-EXP4 converges is not necessarily the optimal threshold policy for problem (5). In the next theorem, we prove Ly-EXP4 is asymptotically near-optimal.

Theorem 1 (Optimality of Ly-EXP4)**.** *Let π and π_{θ^*} denote the policy of Ly-EXP4 and the optimal threshold policy, respectively. By Assumption 1, the regret bound in Proposition 1, and for $B = \frac{1}{2}(1 + (\frac{\gamma}{\lambda})^2)$,*

$$A_N(\pi) - A_N(\pi_{\theta^*}) \leq \frac{B}{V} + \frac{R_{N,m'}}{N} + \epsilon_0, \tag{27}$$

where m' is the expert with threshold $\theta_{m'} \in \mathcal{M}$ that yields the best solution to problem (5)*. For $\eta = \sqrt{\frac{2\log(M)}{N\zeta}}$ and as $N \to \infty$, the optimality gap between Ly-EXP4 and the optimal threshold policy is*

$$\lim_{N\to\infty} [A_N(\pi) - A_N(\pi_{\theta^*})] = O(\frac{1}{V}) + \epsilon_0. \tag{28}$$

The proof is provided in Appendix C.

Algorithm 1 Lyapunov - EXP4 (Ly-EXP4)

Input: N, γ, λ, V, $\mathcal{M}$, η
Initialize $Q_1 = 0$, $\hat{S}_{0,m} = 0, \forall m$
for $n = 1, \cdots, N$ **do**
  Observe Z_n,
  $p_n = \dfrac{\sum_{m:\theta_m > Z_n} e^{-\eta \hat{S}_{n-1,m}}}{\sum_{m \in \mathcal{M}} e^{-\eta \hat{S}_{n-1,m}}}, \quad \forall m$
  Choose the offloading decision $I_n \sim p_n$,
  Compute estimates of expert losses $\hat{L}_{n,m}, \ \forall m$
  $\hat{S}_{n,m} = \hat{S}_{n-1,m} + \hat{L}_{n,m}, \ \forall m$
  $Q_{n+1} = \max[Q_n + I_n X_n - \frac{\gamma}{\lambda}, 0]$

As seen from Proposition 1, the regret achieved by Ly-EXP4 differs by only a constant factor from EXP4. This means that by Theorem 1, Ly-EXP4 solves constrained bandit problems at a convergence rate similar to that of EXP4. Note that increasing M and/or V will decrease the sub-optimality error, but M increases the complexity of the algorithm while V increases the sensitivity to constraint violation and decreases convergence speed.

C. Extension to Multiple Thresholds

In Section III-B, we provide a regret and discuss an optimality analysis for the setting where a single instance of Ly-EXP4 is used to find a single threshold policy applied by all clients. However, by treating different groups of clients as different "contexts", one can deploy separate instances of Ly-EXP4 for each group of clients to learn a multi-threshold policy for the offloading problem (5). At one extreme, each client can use a separate instance of Ly-EXP4 to find an individual threshold policy, which eventually allows our algorithm to be used in a distributed manner.

In what follows, we show the regret for the multi-threshold case based on the principles outlined in Chapter 18.1 of [12].

Let $g \in \mathcal{G}$ denote the set of disjoint groups of clients, and G_n denote the group task n belongs. By Proposition 1, the regret of the Ly-EXP4 instance executed for group $g \in \mathcal{G}$ is

$$R_N^{(g)} \leq \sqrt{2 \sum_{n=1}^{N} \mathbb{I}\{G_n = g\} \zeta \log(M)}, \tag{29}$$

where the sum under the square root counts the number of tasks that belong to group $g \in \mathcal{G}$. Since the number of tasks received by the clients in a group is not known in advance, we use the changing learning rate encountered in the anytime version of EXP4,

$$\eta_n^{(g)} = \sqrt{\frac{\log(M)}{\zeta \sum_{t=1}^{n} \mathbb{I}\{G_t = g\}}}. \tag{30}$$

Defining R_N as the sum of individual regrets across all groups, the regret for multi-threshold case can be bounded as

$$R_N \leq \sum_{g \in \mathcal{G}} R_N^{(g)} \leq \sum_{g \in \mathcal{G}} \sqrt{2 \sum_{n=1}^{N} \mathbb{I}\{G_n = g\} \zeta \log(M)} \tag{31}$$

$$\leq \sqrt{2N|\mathcal{G}|\zeta \log(M)}, \tag{32}$$

where (32) is the worst-case regret corresponding to the setting where each group receives same number of tasks.

As seen from (32), the regret bound for multiple threshold case scales up with the square root of the number of client groups. As discussed in Section IV, although Ly-EXP4 learns a multiple threshold policy slower than a single threshold policy, it achieves significantly better accuracy with a multiple threshold policy by exploiting the heterogeneity in service time and confidence metric distributions across client groups.

IV. Simulation Results

We conducted extensive simulations running Ly-EXP4 in both single and multiple thresholds settings on real and synthetic models. Specifically, we consider a hierarchical inference system with 4 client groups – with 25 clients each, each experiencing different service times due to various factors including distance to the server and channel quality. For this system, we report results that demonstrate convergence of Ly-EXP4 as well as delay-accuracy and fairness-accuracy tradeoffs.

In the synthetic model, client $k \in \mathcal{K}^{(g)}$ receives tasks according to a Poisson process with rate λ_k, where $\mathcal{K}^{(g)}$ denotes the set of clients in client group $g \in \mathcal{G}$. The true class C_n is assigned to each task according to a client-specific class distribution $\boldsymbol{\alpha}_k$, where the arrival rates λ_k, and the class probabilities $\alpha_k(c),\ c \in \mathcal{C}$ are sampled from a uniform distribution with a support $[0, 1]$ and normalized so that total arrival rate $\lambda = \sum_{g \in \mathcal{G}} \sum_{k \in \mathcal{K}^{(g)}} \lambda_k = 1$ and $\sum_{c \in \mathcal{C}} \alpha_k(c) = 1$. A synthetic classifier model for L-ML is characterized by a confusion matrix, denoted by H, where each element represents the likelihood of the L-ML model predicting class c' given that the true class is c, i.e., $H_{c,c'} = \mathbb{P}(\hat{C}_n = c' | C_n = c)$. The elements of H are generated uniformly at random from $[0, 1]$, and the diagonal elements are adjusted such that the model is typically most confident about the true class. Each row is then normalized so that $\sum_{c' \in \mathcal{C}} H_{c,c'} = 1,\ \forall c \in \mathcal{C}$.

In addition to the above, confidence Z_n is sampled from a distribution depending on the correctness of the inference,

$$Z_n \sim \begin{cases} \text{Uniform}(0.1, 0.9), & \text{if } C_n \neq \hat{C}_n, \\ \text{Uniform}(0.5, 1), & \text{if } C_n = \hat{C}_n. \end{cases} \tag{33}$$

In the case of real datasets and models, the tasks are assigned to the clients with probability λ_k/λ regardless of the true class of a task; confidence Z_n is observed as the maximum of the last soft-max layer of the classifier model. We utilize relatively small models that are deployable on IoT devices, including a simple neural network with a single dense layer applied to FMNIST dataset [24], and a LeNet-5 model [25] applied to CIFAR-10 dataset [26].

The clients in a client group experience random service times with group-specific distributions. We assign equally-separated mean service times $[0.2, 0.4, 0.6, 0.8]$ to the client groups. The service times of a client in group g are distributed according to a beta distribution with mean μ_g^{-1} and variance 0.01. The service times for synthetic and real dataset/model cases are generated the same way.

To our knowledge, this is the first work focusing on the hierarchical inference under utilization constraint. To benchmark the proposed Ly-EXP4 algorithm, we compared it with the optimal threshold policies and a simple token bucket policy. When running Ly-EXP4, confidence support Ω_Z is uniformly discretized using $M = 1000$ thresholds. To characterize upper performance limits, we introduce a genie algorithm: a non-causal offline algorithm with access to the ground truth, capable of offloading the maximum number of incorrectly classified tasks at the lowest cost within the constraint. We also compute the optimal single and multiple threshold policies using an exhaustive search over $[1/C, 1]^{|\mathcal{G}|}$. The aforementioned token bucket policy makes greedy offloading decisions based on the service times of the tasks and the token bucket increments by $\frac{\gamma}{\lambda}$ at each task arrival. It offloads task n if the token bucket has more tokens than the service time X_n. When task n is offloaded, X_n gets deducted from the token bucket. Therefore, it is aware of the service times before offloading.

A. Convergence of Ly-EXP4

Figure 1 shows the average accuracy and utilization achieved by Ly-EXP4 and the benchmarks on both synthetic and real datasets/models. The service times, arrival rates and utilization constraint parameter γ are kept constant across the simulations; however, inference confidences, L-ML accuracy of the models, and task class distributions $\boldsymbol{\alpha}_k$ depend on the dataset and model used. From these graphs, it is evident that both versions of the Ly-EXP4 algorithm successfully converge to their respective optimal solutions within the specified constraint. Additionally, it is observed that Ly-EXP4 achieves higher accuracy with a slightly slower convergence speed by exploiting heterogeneity across clients while meeting the utilization constraint.

Note that in some scenarios the token bucket algorithm achieves accuracy similar to Ly-EXP4 with multiple thresholds (see Figure 1a), while in other scenarios it performs worse than Ly-EXP4 with single threshold (see Figure 1c). This inconsis-

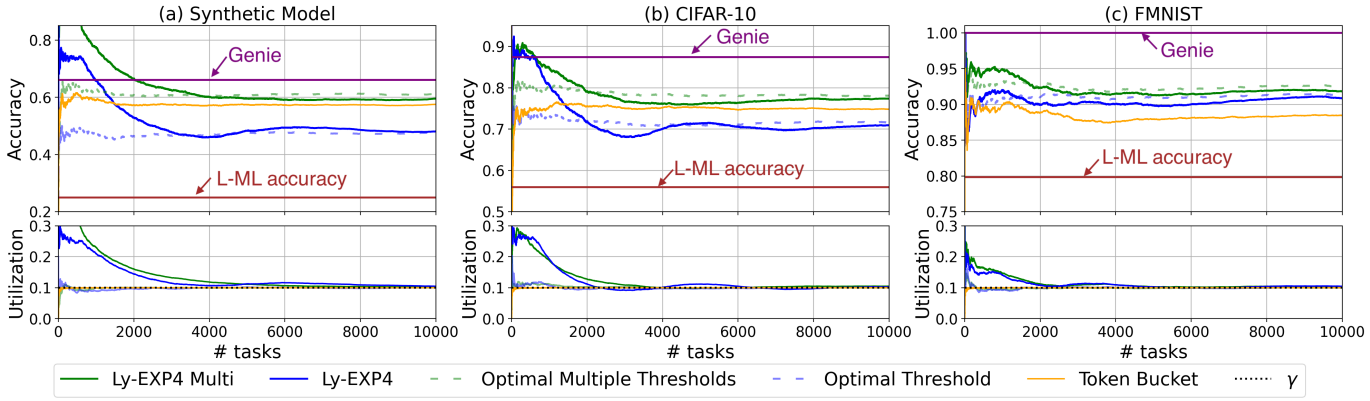

Fig. 1: Sample accuracy as the number of tasks grows. The learning rate η is determined according to Theorem 1 and (30). In all cases $V = 30$, $\gamma = 0.1$ and $\lambda = 1$.

tent performance of the token bucket algorithm is due to its greedy policy that prioritizes the clients with small service times. If those clients' L-ML have low local accuracy, token bucket achieves very good performance; otherwise, it offloads many correctly classified task and thus uses the resources inefficiently. Ly-EXP4 avoids this by making decisions based on the inference confidence.

B. Delay-Accuracy Tradeoff

We further investigate the accuracy-delay trade-off relying on the monotonic relation between utilization and average delay that is typical of scheduling policies. We simulate a network scenario where at most one task from each client can be offloaded using a shared communication channel under a processor sharing scheduling. If a client has an ongoing offload, any new tasks received by the client wait in a client-specific FCFS queue. If a task is offloaded, the delay includes the waiting time in the client's FCFS queue and the transmission time on the processor sharing channel; if a task is not offloaded, its delay is zero. The communication time of task n depends not only on the random service time X_n but also on other tasks simultaneously being offloaded.

Ly-EXP4 updates its weights based on the loss it receives after each offload, which is acceptable for scheduling policies where offloads are performed sequentially (e.g., FCFS). However, this poses a problem for scheduling policies allowing simultaneous offloads such as ours, since some offloading decisions may be taken based on outdated parameters. It was shown in [27] that delayed feedback in EXP3 causes an increase in the regret bound which scales with the number of decisions based on outdated parameters. However, the effect of delayed feedback in our settings is negligible since the offloading decisions for the upcoming tasks are made only after the ongoing offload is completed.

For increasing utilization constraint, corresponding average delay and accuracy results are illustrated together in Figure 2. One can observe diminishing returns in accuracy with respect to delay. Moreover, the accuracy improvements obtained by utilizing multiple thresholds is more significant when the utilization constraint is low, implying that using multiple thresholds is more beneficial when the resources are scarce and thus more efficient offloading decisions are needed.

C. Fairness-Accuracy Tradeoff

As the results show, Ly-EXP4 with multiple thresholds generally achieves higher accuracy than the single threshold strategy. This is accomplished by exploiting the heterogeneity in service time, and task and class distributions across clients. We complement those results by characterizing fairness as quantified by the Jain's index

$$\mathcal{J}(A_N^{(1)}, A_N^{(2)}, \dots, A_N^{(K)}) = \frac{(\sum_{k=1}^K A_N^{(k)})^2}{K \sum_{k=1}^K A_N^{(k)^2}}, \qquad (34)$$

where $A_N^{(k)} = \frac{1}{N_k} \sum_{n=1}^N Y_n(1 - I_n)\mathbb{I}\{K_n = k\}$ is the sample accuracy of client k and N_k is the number of tasks it receives.

Figure 3 shows that Ly-EXP4 with a single threshold is almost perfectly fair across clients, which also means that a single instance of Ly-EXP4 is fair to the clients within a group. As the number of client groups grows, one observes an improvement in the overall accuracy at the expense of fairness. This allows flexibility when utilizing Ly-EXP4 in different settings. By comparison, the token bucket policy achieves fairness similar to Ly-EXP4 with 4 groups. However, token bucket cannot provide an option of trading fairness for accuracy the way Ly-EXP4 can.

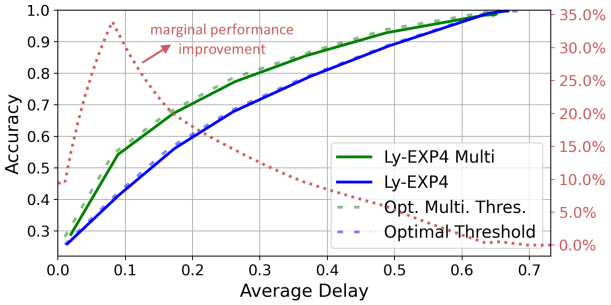

Fig. 2: Average delay vs. accuracy under increasing utilization constraint.

V. Conclusion

In this paper, we proposed a low-complexity online offloading algorithm that we refer to as Lyapunov-EXP4 (Ly-EXP4).

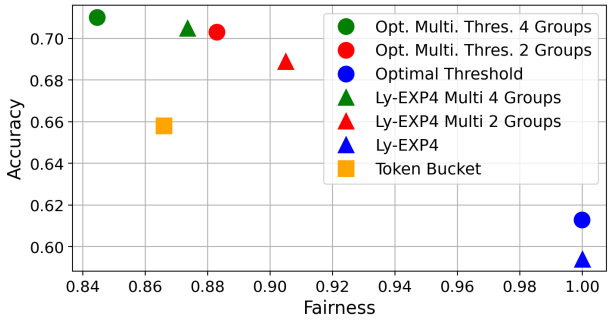

Fig. 3: Accuracy vs. fairness as the number of client groups varies.

Ly-EXP4 combines Lyapunov drift minimization techniques and the bandits with expert advice framework to maximize accuracy in a hierarchical system with multiple clients, converging to an optimal threshold policy within the utilization constraint. We proved a sublinear regret bound and near-optimality of Ly-EXP4; the proposed framework and presented analysis may be relevant to a broader class of constrained bandit problems. The convergence of Ly-EXP4 and the delay-accuracy-fairness tradeoffs achievable in hierarchical inference systems are demonstrated via simulations on real and synthetic datasets and models. As part of the future work, we aim to explore settings with multiple servers having varying communication capacities and model accuracy, and offloading policies that exploit heterogeneity across not only clients but also task classes. Exploring fairness-performance tradeoffs in such system present another line of potentially interesting future work.

APPENDIX

A. Proof of Lemma 1

The loss estimate (23) can be re-written as

$$\hat{L}_{n,m} = \frac{l_{n,I_{n,m}} \mathbb{I}\{I_n = 1\}}{p_n}, \quad \forall n, m. \tag{35}$$

Since $\mathbb{E}_{n-1}[\mathbb{I}\{I_n = 1\}] = p_n$,

$$\mathbb{E}_{n-1}[\hat{L}_{n,m}] = \mathbb{E}_{n-1}\left[\frac{l_{n,I_{n,m}} \mathbb{I}\{I_n = 1\}}{p_n}\right] \tag{36}$$

$$= \frac{l_{n,I_{n,m}}}{p_n} \mathbb{E}_{n-1}\left[\mathbb{I}\{I_n = 1\}\right] = l_{n,I_{n,m}}, \tag{37}$$

which shows $\hat{L}_{n,m}$ is indeed an unbiased estimate of $l_{n,I_{n,m}}$.

B. Proof of Proposition 1

Lemma 2 (Bounded first and second moment of Q_n)**.** *For $V > 0$ and a lower bound on service times $x_{min} \leq X_n$, $\forall n$, the virtual queue process under Ly-EXP4 policy has bounded first and second moment*

$$\mathbb{E}[Q_n] = O(\tfrac{V}{x_{min}}), \quad \mathbb{E}[Q_n^2] = O((\tfrac{V}{x_{min}})^2). \tag{38}$$

Proof Sketch. This proof is inspired by [28] and the extensions in [20] which allow us to bound the moment generating function of Q_n. Due to space limitations we refer the reader to Lemma 7 in [20] for the intermediate steps of the proof.

From (8), for some $l^* > \frac{\gamma}{\lambda}$, $\beta^* < \frac{\gamma}{\lambda}$ and $\epsilon > 0$, we obtain

$$\mathbb{E}_{n-1}[Q_{n+1} - Q_n \mathbb{I}\{Q_n \geq l^*,\ p_n < \beta^*\}] \leq -\epsilon. \tag{39}$$

By using the Taylor expansion on $\mathbb{E}_{n-1}[e^{\eta_0(Q_{n+1}-Q_n)}]$, the negative drift (39), and due to the fact $|Q_{n+1} - Q_n| < 1$ because X_n is bounded and Y_n is in $(0, 1]$, for some $\eta_0 < 1$ and $\rho_0 = 1 - \epsilon\eta_0 + \eta_0^2(e-2)$ it holds that

$$\mathbb{E}_{n-1}\left[e^{\eta_0(Q_{n+1}-Q_n)} \mathbb{I}\{Q_n \geq l^*,\ p_n < \beta^*\}\right] \leq \rho_0 < 1, \tag{40}$$

$$\mathbb{E}_{n-1}\left[e^{\eta_0(Q_{n+1}-Q_n)} \mathbb{I}\left\{Q_n < l^*\right\}\right] \leq e^{\eta_0}. \tag{41}$$

Let $x_{\min}$ denote a lower bound on X_n, i.e., $X_n \geq x_{\min} > 0$, $\forall n$. Defining $l^* = \frac{V}{x_{\min}}$, we argue that $\{Q_n \geq l^*,\ p_n \geq \beta^*\}$ is a low-probability event, i.e.,

$$\mathbb{P}\{Q_n \geq l^*,\ p_n \geq \beta^*\} \leq \delta_0. \tag{42}$$

Let L_{n,m_a} be the loss of any expert m with offloading decision $I_{n,m} = a$. The low-probability event argument stems from the fact that by the definition of the expert weights and expert losses, if $Q_n \geq \frac{V}{x_{\min}}$, then $L_{n,m_1} \geq L_{n,m_0}$ for all m, and the weight distribution $\boldsymbol{\omega}_n$ shifts towards smaller values, making the algorithm less likely to offload.

Following the steps in [20], and using (40), (41) and (42), leads us to the inequality

$$\mathbb{E}_0\left[e^{\eta_0 Q_{n+1}}\right] \leq \rho \mathbb{E}_0\left[e^{\eta_0 Q_n}\right] + e^{\eta_0(l^*+1)}, \tag{43}$$

where $\rho = \rho_0 + \sqrt{\delta_0(\frac{1}{1-\psi} + 1)} < 1$ and $\psi = \frac{1}{4}(e^{3\eta_0} - e^{\eta_0})$. Taking telescoping sum over n yields

$$\mathbb{E}_0\left[e^{\eta_0 Q_n}\right] \leq \rho^n e^{\eta_0 Q_1} + \frac{1-\rho^n}{1-\rho} e^{\eta_0(l^*+1)}. \tag{44}$$

Applying the Chernoff bound, we obtain a bound on the cumulative CDF of Q_n,

$$\mathbb{P}\left[Q_n \geq q\right] \leq \rho^n e^{\eta_0 Q_1 - q} + \frac{1-\rho^n}{1-\rho} e^{\eta_0(l^*+1-q)}. \tag{45}$$

Since Q_n is non-negative for all n and $Q_1 = 0$, for $\eta_0 = \frac{2}{l^*} < 1$ we obtain a bound on the second moment

$$\mathbb{E}[Q_n^2] = 2\int_0^\infty q\mathbb{P}\left[Q_n \geq q\right] dq \tag{46}$$

$$\leq \frac{2}{\eta_0^2}(\rho^n + \frac{1-\rho^n}{1-\rho}) e^{\eta_0(l^*+1)} = O(l^{*2}). \tag{47}$$

Setting $l^* = \frac{V}{x_{\min}}$ completes the proof. The proof for the first moment follows similar procedure. □

Proof of the Regret Bound. Let us define the expected regret relative to expert m as

$$R_{N,m} = \mathbb{E}\left[\sum_{n=1}^N L_n - \sum_{n=1}^N l_{n,I_{n,m}}\right]. \tag{48}$$

To prove the claim, we will bound $R_{n,m}$ for all m, including the best expert. Let $\hat{S}_{n,m} = \sum_{t=1}^n \hat{L}_{t,m}$ denote the cumulative estimated loss for expert m. For the sake of convenience and brevity, we introduce $p_{n,a} = \mathbb{P}\{I_n = a\}$, $\forall a \in \{0, 1\}$, where $p_{n,1} = p_n$ and $p_{n,0} = 1 - p_n$. Let $\hat{L}_{n,m_a}$ be the loss estimate of any expert m with offloading decision $I_{n,m} = a$. From Lemma 1,

$$\mathbb{E}[\hat{S}_{n,m}] = \sum_{t=1}^n l_{t,I_{t,m}}, \tag{49}$$

$$\mathbb{E}_{n-1}[L_n] = \sum_{a\in\{0,1\}} p_{n,a} l_{n,a} = \sum_{a\in\{0,1\}} p_{n,a} \mathbb{E}_{n-1}[\hat{L}_{n,m_a}]. \tag{50}$$

By the tower rule for conditional expectations and (50),

$$R_{N,m} = \mathbb{E}[\sum_{n=1}^{N} \sum_{a\in\{0,1\}} p_{n,a}\mathbb{E}_{n-1}[\hat{L}_{n,m_a}]] - \mathbb{E}[\hat{S}_{N,m}] \quad (51)$$

$$= \mathbb{E}[\sum_{n=1}^{N} \sum_{a\in\{0,1\}} p_{n,a}\hat{L}_{n,m_a}] - \mathbb{E}[\hat{S}_{N,m}] \quad (52)$$

$$= \mathbb{E}[\hat{S}_N - \hat{S}_{N,m}]. \quad (53)$$

Let us introduce $W_n = \sum_{m=1}^{M} \exp(-\eta\hat{S}_{n,m})$ and note that for all experts $\hat{S}_{0,m} = 0$. Due to $\hat{S}_{n,m} \geq 0$, $\forall m, n$,

$$\exp(-\eta\hat{S}_{N,m}) \leq \sum_{m=1}^{M} \exp(-\eta\hat{S}_{N,m}) = W_N = M\prod_{n=1}^{N}\frac{W_n}{W_{n-1}} \quad (54)$$

The ratio in the product above can be written in terms of $p_{n,a}$,

$$\frac{W_n}{W_{n-1}} = \sum_{m=1}^{M} \frac{\exp(-\eta\hat{S}_{n-1,m})}{W_{n-1}} \exp(-\eta\hat{L}_{n,m}) \quad (55)$$

$$= \sum_{a\in\{0,1\}} p_{n,a}\exp(-\eta\hat{L}_{n,m_a}) \quad (56)$$

By using the fact that $1 - x \leq \exp(-x) \leq 1 - x + \frac{x^2}{2}$, where the lower bound is valid $\forall x \in \mathbb{R}$ and the upper bound is valid $\forall x \geq 0$,

$$\frac{W_n}{W_{n-1}} \overset{(a)}{\leq} \sum_{a\in\{0,1\}} p_{n,a}(1 - \eta\hat{L}_{n,m_a} + \frac{\eta^2}{2}\hat{L}^2_{n,m_a}) \quad (57)$$

$$\overset{(b)}{\leq} \exp(-\eta \sum_{a\in\{0,1\}} p_{n,a}\hat{L}_{n,m_a} + \frac{\eta^2}{2}\sum_{a\in\{0,1\}} p_{n,a}\hat{L}^2_{n,m_a}), \quad (58)$$

where in (a) the upper bound and in (b) the lower bound of $\exp(-x)$ are used. When we substitute (57) in (54), we obtain

$$e^{-\eta\hat{S}_{N,m}} \leq M\exp(-\eta\sum_{n=1}^{N}\sum_{a\in\{0,1\}} p_{n,a}\hat{L}_{n,m_a} + \frac{\eta^2}{2}\sum_{t=1}^{N}\sum_{a\in\{0,1\}} p_{n,a}\hat{L}^2_{n,m_a}) \quad (59)$$

$$= M\exp(-\eta\hat{S}_N + \frac{\eta^2}{2}\sum_{t=1}^{N}\sum_{a\in\{0,1\}} p_{n,a}\hat{L}^2_{n,m_a}) \quad (60)$$

By taking the logarithm of both sides and rearranging the terms, it follows that

$$\hat{S}_N - \hat{S}_{N,m} \leq \frac{log(M)}{\eta} + \frac{\eta}{2}\sum_{n=1}^{N}\sum_{a\in\{0,1\}} p_{n,a}\hat{L}^2_{n,m_a}. \quad (61)$$

Taking the expectation gives us a bound on regret relative to any expert,

$$R_{N,m} \leq \frac{log(M)}{\eta} + \frac{\eta}{2}\mathbb{E}[\sum_{n=1}^{N}\sum_{a\in\{0,1\}} p_{n,a}\hat{L}^2_{n,m_a}]. \quad (62)$$

Finally, we need to bound the last term on the right side. Since $\mathbb{E}_{n-1}[\hat{L}_{n,m}] = l_{t,I_{n,m}}$,

$$\mathbb{E}[\sum_{n=1}^{N}\sum_{a\in\{0,1\}} p_{n,a}\hat{L}^2_{n,m_a}] = \mathbb{E}[\sum_{n=1}^{N}\sum_{a\in\{0,1\}} p_{n,a}\mathbb{E}_{n-1}[\hat{L}^2_{n,m_a}]]$$

$$= \mathbb{E}[\sum_{n=1}^{N}(p_{n,0}l^2_{n,0} + p_{n,1}l^2_{n,1})] \leq N + \sum_{n=1}^{N}\mathbb{E}[p_{n,1}(\frac{Q_nX_n}{V})^2]$$

$$\leq N + \frac{1}{V^2}\sum_{n=1}^{N}\mathbb{E}[Q^2_n] \leq N\zeta, \quad (63)$$

where the last inequality follows from Lemma (2) and $\zeta = O(\frac{1}{x_{\min}}) + 1$ is a constant. Substituting (63) in (62) completes the proof. □

C. Proof of Suboptimality of Ly-EXP4

Let π denote the Ly-EXP4 policy. From (14),

$$\Delta(Q^\pi_n) + V\mathbb{E}_{n-1}[Y_n(1 - I^\pi_n)]$$

$$\leq B + Q^\pi_n\mathbb{E}_{n-1}[I^\pi_n X_n] + V\mathbb{E}_{n-1}[Y_n(1 - I^\pi_n)] - Q^\pi_n\frac{\gamma}{\lambda} \quad (64)$$

$$= B + \mathbb{E}_{n-1}[L_n] - Q^\pi_n\frac{\gamma}{\lambda}, \quad (65)$$

where the last equation stems from the definition of L_n. By the telescoping argument and taking the expectation we obtain

$$\mathbb{E}[\mathcal{L}(Q^\pi_N)] + V\sum_{n=1}^{N}\mathbb{E}[Y_n(1 - I^\pi_n)]$$

$$\leq NB + V\mathbb{E}[S_N] - \frac{\gamma}{\lambda}\sum_{n=1}^{N}E\ [Q^\pi_n]. \quad (66)$$

Let $\theta_{m'} \in \mathcal{M}$ denote the best discretized threshold for the optimization problem (5). For brevity, we denote threshold policy π_θ by (θ). Using the regret bound relative to $\theta_{m'}$ in (66) yields

$$\mathbb{E}[\mathcal{L}(Q^\pi_N)] + V\sum_{n=1}^{N}\mathbb{E}[Y_n(1 - I^\pi_n)]$$

$$\leq NB + V\mathbb{E}[S_{N,m'}] + VR_{N,m'} - \frac{\gamma}{\lambda}\sum_{n=1}^{N}E\ [Q^\pi_n] \quad (67)$$

$$= NB + \sum_{n=1}^{N}\mathbb{E}[Q^\pi_n]\mathbb{E}[I^{(\theta_{m'})}_n X_n]\ + V\sum_{n=1}^{N}\mathbb{E}[Y_n(1 - I^{(\theta_{m'})}_n)]$$

$$+ VR_{N,m'} - \frac{\gamma}{\lambda}\sum_{n=1}^{N}E\ [Q^\pi_n] \quad (68)$$

$$\leq NB + V\sum_{n=1}^{N}\mathbb{E}[Y_n(1 - I^{(\theta_{m'})}_n)] + VR_{N,m'}, \quad (69)$$

where in the last inequality we use the fact that $\theta_{m'}$ satisfies the utilization constraint (6). Since $\mathcal{L}(Q^\pi_N) \geq 0$ and by Assumption 1, rearranging the terms and dividing both sides by NV leads to

$$A_N(\pi) - A_N(\theta^*) \leq \frac{B}{V} + \frac{R_{N,m'}}{N} + \epsilon_0, \quad (70)$$

where $A_N(\pi)$ and $A_N(\theta^*)$ are the average accuracy at time N under Ly-EXP4 and the optimal threshold policy, respectively. According to Proposition 1, Ly-EXP4 converges to a threshold policy corresponding to the best expert in the bandit setting. Therefore, the limit $\lim_{N\to\infty} A_N(\pi)$ exists, which completes the proof.

REFERENCES

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[2] L. Shen, Y. Sun, Z. Yu, L. Ding, X. Tian, and D. Tao, "On efficient training of large-scale deep learning models: A literature review," Apr. 2023, arXiv:2304.03589.

[3] K. Y. Chan, B. Abu-Salih, R. Qaddoura, A. M. Al-Zoubi, V. Palade, D.-S. Pham, J. D. Ser, and K. Muhammad, "Deep neural networks in the cloud: Review, applications, challenges and research directions," *Neurocomputing*, vol. 545, p. 126327, 2023.

[4] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1595–1623, Apr. 2022.

[5] V. N. Moothedath, J. P. Champati, and J. Gross, "Online algorithms for hierarchical inference in deep learning applications at the edge," 2023, arXiv:2304.00891.

[6] I. Nikoloska and N. Zlatanov, "Data selection scheme for energy efficient supervised learning at iot nodes," *IEEE Communications Letters*, vol. 25, no. 3, pp. 859–863, Mar. 2021.

[7] A. Chakrabarti, R. Guérin, C. Lu, and J. Liu, "Real-time edge classification: Optimal offloading under token bucket constraints," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, Dec. 2021, pp. 41–54.

[8] A. Fresa and J. P. V. Champati, "An offloading algorithm for maximizing inference accuracy on edge device in an edge intelligence system," in *Proceedings of the 25th International ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, New York, NY, USA, Oct. 2022, pp. 15–23.

[9] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," in *Proceedings of the Twenty-Second International Conference on Architectural Support for Programming Languages and Operating Systems*, New York, NY, USA, 2017, p. 615–629.

[10] C. Hu, W. Bao, D. Wang, and F. Liu, "Dynamic adaptive dnn surgery for inference acceleration on the edge," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 1423–1431.

[11] S. Teerapittayanon, B. McDanel, and H. Kung, "BranchyNet: Fast inference via early exiting from deep neural networks," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Dec. 2016, pp. 2464–2469.

[12] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[13] M. Neely, *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool Publishers, 2010.

[14] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[15] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, Apr. 2018, pp. 207–215.

[16] K. Guo, R. Gao, W. Xia, and T. Q. S. Quek, "Online learning based computation offloading in mec systems with communication and computation dynamics," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 1147–1162, Feb. 2021.

[17] H. Liao, Z. Zhou, B. Ai, and M. Guizani, "Learning-based energy-efficient channel selection for edge computing-empowered cognitive machine-to-machine communications," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, May 2020, pp. 1–6.

[18] X. Liu, B. Li, P. Shi, and L. Ying, "POND: Pessimistic-optimistic online dispatching," May 2021, arXiv:2010.09995.

[19] ——, "An efficient pessimistic-optimistic algorithm for stochastic linear bandits with general constraint," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 24 075–24 086.

[20] S. Cayci, Y. Zheng, and A. Eryilmaz, "A lyapunov-based methodology for constrained optimization with bandit feedback," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 3716–3723.

[21] S. Cayci, S. Gupta, and A. Eryilmaz, "Group-fair online allocation in continuous time," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 750–13 761, 2020.

[22] A. Badanidiyuru, R. D. Kleinberg, and A. Slivkins, "Bandits with knapsacks," *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 207–216, 2013.

[23] S. Agrawal and N. R. Devanur, "Bandits with concave rewards and convex knapsacks," in *Proceedings of the fifteenth ACM conference on Economics and computation*, ser. EC '14. New York, NY, USA: Association for Computing Machinery, Jun. 2014, pp. 989–1006.

[24] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. [Online]. Available: https://arxiv.org/abs/1708.07747

[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[26] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[27] I. Bistritz, Z. Zhou, X. Chen, N. Bambos, and J. Blanchet, "Online exp3 learning in adversarial bandits with delayed feedback," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[28] B. Hajek, "Hitting-time and occupation-time bounds implied by drift analysis with applications," *Advances in Applied Probability*, vol. 14, no. 3, pp. 502–525, 1982.