Provisioning Edge Inference as a Service via Online Learning

Yibo Jin[1], Lei Jiao[2], Zhuzhong Qian[1], Sheng Zhang[1], Ning Chen[1], Sanglu Lu[1], Xiaoliang Wang[1]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, China [2]University of Oregon, USA

***Abstract*—Provisioning machine learning inference as a service at the mobile network edge for distributed users in an online setting faces multiple challenges, including the accuracy-resource trade-off for model selection, the time-coupled decision for model distribution, and the unpredictable user inference workload. To overcome such challenges, we firstly model an online time-varying non-linear integer program of maximizing the overall service's inference accuracy through dynamic model instance selection, delivery and workload distribution. Afterwards, we design an online learning algorithm to make fractional control decisions, which alternates between minimizing an outer problem and maximizing an inner problem of an equivalent convex-concave formulation by only taking previously observable inputs. We further design a randomized rounding algorithm to convert the fractional decisions into integers. We rigorously prove that our approach only incurs sub-linear dynamic regret for the optimality loss and sub-linear dynamic fit for the long-term constraints violation. Finally, we conduct extensive evaluations with real-world data and confirm the empirical superiority of our approach over state-of-the-art algorithms in terms of up to 30% reduction on accuracy loss and 34% reduction on constraints violation.**

I. Introduction

While machine learning models are mostly trained in cloud data centers today, there is a push towards moving machine learning inference to the network edge of the mobile edge computing infrastructures [1, 2] in closer proximity to end users. Such *edge inference* can bring ultra low response time to end users, reduce traffic beyond the edge networks, and ensure better user privacy [3] as users' inference queries are answered locally; compared to on-device inference [4–6], executing inference in nearby edge infrastructures overcomes the drawback of the limited resource and battery capacities of mobile devices, requires no high-end processors on the device, and can thus serve a wide range of users.

However, provisioning edge inference is a complicated non-trivial process for service providers, which involves the management of loading machine learning models across networks and serving users' inference workload over distributed edges, as depicted in Fig. 1. Particularly, managing edge inference optimally in an online manner faces fundamental challenges:

First and foremost, loading machine learning models from the cloud to distributed edges entails dynamically navigating the trade-offs between accuracy and resource consumption [7, 8] within the heterogenous resource and network capacities of the underlying infrastructures. While models of higher inference accuracies often have larger sizes and require more computation [7, 9] when executing inference queries, it is uneasy to determine the number of model instances to fit into each edge in each time slot in an online setting, due to the following dilemma: loading more instances will consume more resources than necessary; but hosting fewer instances in the current time slot and if additional instances are needed as it goes to the next time slot, loading them may be prohibitive due to the available bandwidth constraints. That is, any decision made currently potentially restricts the decisions which will be made next, and such decision coupling over time is generally hard to handle in online problems [10, 11].

978-1-7281-6630-8/20/$31.00 © 2020 IEEE

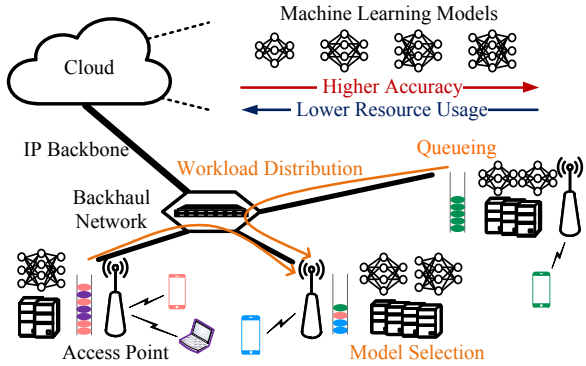


Fig. 1: System architecture for inference provisioning at edge

Furthermore, user inference queries are often unpredictable due to users' dynamic arrivals, departures, mobility, and device usage patterns [12, 13]. Machine learning models also need to be chosen and placed at the edges *before* the inference workload arrives. The difficulty for online algorithm design caused by such obliviousness to the uncertain inputs further escalates due to the queueing state transitions, as we need to determine how many inference queries to serve from each queue at each edge before new inference queries arrive and enter these queues, and ensure all such queries are served eventually. Intuitively, in every time slot, one can "learn" in an online manner [10, 14] from the "penalty" incurred by the online decisions just made regarding the machine learning model provisioning and inference workload distribution after the inference workload actually arrives, and seek to make better decisions as time goes; however, how to design such an effective "online learning" algorithm to clear the queues remains a challenging problem.

Existing research falls insufficient for addressing the aforementioned challenges. Some works [2, 15–19] focused on optimizing and executing machine learning inference in individual devices/systems, and rarely studied inference optimization over heterogeneous edges from a *service perspective* as well

as in an *online* setting. Other literatures [10, 20–23] aimed at online service provisioning, but failed to treat the challenges for machine learning inference and *integral* online decisions.

In this paper, we investigate the online problem of optimizing the overall inference accuracy of the machine learning service over the heterogenous, resource-constrained, distributed edge infrastructure while accommodating the unpredictable inference workload. We make the following contributions:

We model this problem as a time-varying non-linear integer program with *long-term constraints*. Our problem maximizes the overall inference accuracy subject to the constraints of queueing state transition, machine learning model selection and delivery, and inference workload distribution. The blindness that the workload is only revealed after decisions with regards to models and workload are made in each time slot hampers us from satisfying the constraints for each time slot. We thus choose to design online algorithms that optimize the objective and upper-bound the cumulative, long-term constraints violation over time. Besides, the problem is NP-hard.

We design a novel polynomial-time online algorithm that consists of an online learning component that makes fractional decisions in each time slot without observing current inputs and a randomized rounding component that converts the fractional decisions into integers without changing the constraints violation in expectation. Our online learning component is based on a convex-concave equivalent formulation, and alternates between minimizing the outer convex problem and maximizing the inner concave problem by taking only previous inputs instead of current inputs to our problem. Our randomized rounding component rounds fractional decisions in pair into integers while letting the two fractions compensate each other and keeping the expectation of randomized integers equal the corresponding fractions. Through rigorous theoretical analysis, we prove that the performance metrics of both the *dynamic regret*, which characterizes the optimality loss relative to a sequence of instantaneous optimizers with known costs and constraints, and the *dynamic fit*, which characterizes the long-term constraint violations, with regard to our entire online algorithm only grow *sub-linearly* along with time.

We conduct extensive numerical evaluations using London's 268 underground stations as the edge infrastructure and the corresponding real-world dynamic passenger statistics in each station over 4 days in November 2016 as the inference workload. We observe that our online algorithm achieves up to 30% and 34% reduction on accuracy loss and constraint violation, respectively, compared with multiple state-of-the-art algorithms. Our proposed algorithm also exhibits the sub-linear growth in the dynamic regret and fit, aligning with our theoretical analysis, and behaves well for different workloads as we appropriately control its parameters.

II. System Model and Problem Formulation

A. System Model

Edge Computing Infrastructure: We consider a computing infrastructure that consists of a group of distributed heterogeneous edges (e.g., cellular base stations with colocated microservers), denoted as $\mathcal{N} = \{1, 2, ..., N\}$. Each edge has its own access point that can be used by the end users, and all the edges connect to one another through the wireline backhaul networks, and further connect to the cloud that hosts the pre-trained machine learning models via the IP backbone. We use c_n, $\forall n \in \mathcal{N}$ to represent the resource capacity (e.g., the total CPUs or memory) of edge n.

TABLE I: Summary of Notations

Symbol	Description	Vector[1]
$r_{n,t}$	Number of queries submitted to edge n at time t	$\mathbf{r}_t$
$q_{n,t}$	Number of untreated queries at edge n at time t	$\mathbf{q}_t$
$a_{m,t}$	Accuracy loss of model m at time t	$\mathbf{a}_t$
$b_{n,t}$	Transference budget of edge n at time t	$\mathbf{b}_t$
τ	Cost for migrating a single inference query	-
c_n	Resource capacity of edge n	$\mathbf{c}$
d_m	Resource requirement of an instance of model m	$\mathbf{d}$
s_m	Size of model m	$\mathbf{s}$
p_m	Processing capability of model m	$\mathbf{p}$
$f_t(\cdot), g_t(\cdot)$	Abstract objective and long-term constraint	-,$\mathbf{g}_t(\cdot)$
α, μ, λ_t	Non-negative algorithmic parameters	-,-,$\boldsymbol{\lambda}_t$
Decision	Description[2]	Vector
$x_{n,m,t}$	$\#_I$ of model m hosted at edge n at time t	$\mathbf{x}_t$
$y_{n,n',t}$	$\#_Q$ migrated from edge n to edge n' at time t	$\mathbf{y}_t$
$z_{n,m,t}$	Indicator of hosting model m at edge n at time t	$\mathbf{z}_t$

1. All vectors are column vectors in this paper, e.g., $\mathbf{a}_t^\top := [a_{1,t}, ..., a_{M,t}]$.
2. $\#_I$: Number of instances; $\#_Q$: Number of queries.

Machine Learning Models: We consider a group of machine leaning models, denoted as $\mathcal{M} = \{1, 2, ..., M\}$, pre-trained, updated and stored in the cloud. For each model $m \in \mathcal{M}$, we denote the inference "accuracy loss" (defined as one minus its percentile accuracy) of its latest version at time t as $a_{m,t}$, its resource requirement (e.g., in terms of CPU or memory) for a single model instance as d_m, and its size (e.g., in terms of bytes) as s_m. We also use $p_m, \forall m \in \mathcal{M}$ to refer to the processing capability of a single instance of model m, i.e., the number of inference queries that a single instance of model m can serve per time slot.

Inference Workload Processing: We consider a series of consecutive time slots $\mathcal{T}$. We denote by $r_{n,t}$, $\forall n \in \mathcal{N}$, $t \in \mathcal{T}$ the number of inference queries submitted by the end users to edge n at time t. Each edge has a local first-in-first-out queue, and all the inference queries join the queue before getting served. We denote the length of the queue as $q_{n,t}$, i.e., the number of untreated queries in the queue at edge n at time t. The inference queries can be distributed or migrated across edges, and can thus enter different queues and then get served. We denote the cost (e.g., in terms of traffic) of migrating a single inference query across edges as τ. We denote the migration or transference budget of edge n at time t as $b_{n,t}$, capturing the (two-way) traffic limit, available bandwidth, etc.

Control Variables: We use $z_{n,m,t} \in \{0, 1\}, \forall n \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T}$ as the indicator to represent whether model m is hosted ($z_{n,m,t} = 1$) or not ($z_{n,m,t} = 0$) at edge n at time t. We denote by $x_{n,m,t}$ the number of the instances of model m that are hosted at edge n at time t. We also denote by $y_{n,n',t}$ the number of users' inference queries that are migrated from edge n to edge n' at time t. These variables are non-negative integers, i.e., $x_{n,m,t}, y_{n,n',t} \in \mathbb{N}, \forall n, n' \in \mathcal{N}, m \in \mathcal{M}, t \in \mathcal{T}$.

For clarity, the main notations are summarized in Table 1.

B. Problem Formulation

Having the system models, we aim to minimize the overall accuracy loss of $\sum_{t=1}^{T}\sum_{n}\sum_{m} a_{m,t}x_{n,m,t}$. Ideally, for each time t, we would need to meet the following constraints.

$$\forall t,n: q_{n,t+1} = [q_{n,t} + \textstyle\sum_{n'} y_{n',n,t} - \sum_{m} p_m x_{n,m,t}]^+, \quad q_{n,1} \geq 0, q_{n,T+1} = 0, \tag{0a}$$

Constraint (0a) characterizes the queue state transition between any two consecutive time slots, where the function of $[\cdot]^+ = \max\{\cdot, 0\}$ ensures the non-negative queue length. The queue length is increased by the arriving queries, and decreased by the queries served. Every queue is eventually cleared.

$$\forall t,n: \textstyle\sum_{n'} y_{n,n',t} = r_{n,t}. \tag{0b}$$

Constraint (0b) captures the workload distribution.

$$\forall t,n: \textstyle\sum_{m} s_m[z_{n,m,t} - z_{n,m,t-1}]^+ + \tau\sum_{n',n'\neq n}(y_{n,n',t} + y_{n',n,t}) \leq b_{n,t}. \tag{0c}$$

Constraint (0c) ensures that downloading models from the cloud and distributing workload over the edges obeys the transference budget. Only one copy of a model needs to be downloaded to an edge for creating instances, if that edge does not host that model previously.

$$\forall t,n,m: d_m x_{n,m,t} \leq z_{n,m,t} c_n. \tag{0d}$$

Constraint (0d) ensures that an edge can have instances of a model only if that model is decided to be hosted at that edge.

$$\forall t,n: \textstyle\sum_{m} d_m x_{n,m,t} \leq c_n. \tag{0e}$$

Constraint (0e) ensures that the resources consumed to process inference queries are within the edge capacity.

$$\forall t,n,n',m: x_{n,m,t}, y_{n,n',t} \in \mathbb{N}, z_{n,m,t} \in \{0,1\}. \tag{0f}$$

Constraint (0f) enforces the variables are appropriate integers.

Problem Formulation: We observe that, because we have no priori knowledge of users' inference queries, it is actually very hard, if ever possible, to make decisions at each time slot on the fly before knowing such workload, while still satisfying the constraints *for each time slot*; therefore, we choose to only enforce the constraints *in the long run*, and aim to design online algorithms to minimize the objective and bound the cumulative constraint violation over time. We formulate the edge inference provisioning problem as follows:

$$\begin{aligned}
\min \quad & \textstyle\sum_{t=1}^{T}\{\sum_{n}\sum_{m} a_{m,t}x_{n,m,t}\} \qquad (1)\\
s.t. \quad & \forall n: \textstyle\sum_{t=1}^{T} g_t^{0,n} \leq 0,\\
& \forall n: \textstyle\sum_{t=1}^{T} g_t^{1,n} := \sum_{t=1}^{T}\{\sum_{n'} y_{n,n',t} - r_{n,t}\} \leq 0,\\
& \forall n: \textstyle\sum_{t=1}^{T} g_t^{2,n} := \sum_{t=1}^{T}\{r_{n,t} - \sum_{n'} y_{n,n',t}\} \leq 0,\\
& \forall n: \textstyle\sum_{t=1}^{T} g_t^{3,n} \leq 0,\\
& \forall n,m: \textstyle\sum_{t=1}^{T} g_t^{4,n,m} \leq 0,\\
& \forall t,n: h_t^n := \textstyle\sum_{m} d_m x_{n,m,t} - c_n \leq 0,\\
var. \quad & \forall t,n,m: x_{n,m,t}, y_{n,n',t} \in \mathbb{N}, z_{n,m,t}^{load} \in \{0,1\},
\end{aligned}$$

where we have converted Constraints (0a)$\sim$(0d) to their long-term versions correspondingly. For (0a), we have

$$\begin{aligned}
\forall n: q_{n,T+1} &\geq q_{n,T} + \textstyle\sum_{n'} y_{n',n,T} - \sum_{m} p_m x_{n,m,T}\\
&\geq ... \geq q_{n,1} + \textstyle\sum_{t=1}^{T}\{\sum_{n'} y_{n',n,t} - \sum_{m} p_m x_{n,m,t}\},
\end{aligned}$$

which is due to the property of $[\cdot]^+$ and $0 \geq q_{n,T+1} - q_{n,1}$, all by definition. Thus, we have

$$\textstyle\sum_{t=1}^{T} g_t^{0,n} := \sum_{t=1}^{T}\{\sum_{n'} y_{n',n,t} - \sum_{m} p_m x_{n,m,t}\} \leq 0.$$

For (0b), we simply have $\sum_{t=1}^{T} g_t^{1,n}$ as well as $\sum_{t=1}^{T} g_t^{2,n}$, as adopted in the formulation of workload distribution above. For (0c), we introduce auxiliary binary variables to denote whether model m should be downloaded from the cloud to edge n at time t, i.e., binary variable $z_{n,m,t}^{load}$, and then have

$$g_t^{3,n} := \textstyle\sum_{m} s_m z_{n,m,t}^{load} + \tau\sum_{n',n'\neq n}(y_{n,n',t} + y_{n',n,t}) - b_{n,t} \leq 0,$$

where $z_{n,m,t}^{load} := [z_{n,m,t} - z_{n,m,t-1}]^+$, $\forall n,m,t$. For (0d), we have $d_m x_{n,m,t}/c_n \leq z_{n,m,t} \leq ... \leq z_{n,m,0} + \sum_{t'=1}^{t} z_{n,m,t'}^{load}$, since $z_{n,m,t}^{load} \geq z_{n,m,t} - z_{n,m,t-1}$. After applying $z_{n,m,0} = 0$ and converting the inequality to its long term version, we have

$$\textstyle\sum_{t=1}^{T} g_t^{4,n,m} := \sum_{t=1}^{T}\{\frac{d_m x_{n,m,t}}{c_n} - (T+1-t)z_{n,m,t}^{load}\} \leq 0.$$

Concise Representation: For the ease of the presentation, we simplify the representation of our problem formulation:

$$\begin{aligned}
&\min \textstyle\sum_{t=1}^{T} f_t(\mathbf{I}_t), \quad s.t. \textstyle\sum_{t=1}^{T} \mathbf{g}_t(\mathbf{I}_t) \preceq \mathbf{0},\ \mathbf{h}(\mathbf{I}_t) \preceq \mathbf{0}, \qquad (2)\\
&\forall t: f_t(\mathbf{I}_t) := \left[\mathbf{a}_t^\top, (\mathbf{0}_{NN\times 1})^\top, (\mathbf{0}_{NM\times 1})^\top\right] \cdot \mathbf{I}_t,\\
&\forall t: \mathbf{g}_t(\mathbf{I}_t) := \big[..., \underbrace{g_t^{0,n}, g_t^{1,n}, g_t^{2,n}, g_t^{3,n}, ..., g_t^{4,n,m}, ...}_{edge\ \ n}, ...\big]^\top,\\
&\forall t: \mathbf{h}(\mathbf{I}_t) := \left[h_t^1, ..., h_t^N\right]^\top,\\
&\forall t: \mathbf{I}_t = \left[\mathbf{x}_t^\top, \mathbf{y}_t^\top, (\mathbf{z}_t^{load})^\top\right]^\top \in \mathbb{N}^D, \mathbf{z}_t^{load} \in \{0,1\}^{NM},
\end{aligned}$$

where the vector $\mathbf{I}_t$ is the aggregation of $\mathbf{x}_t, \mathbf{y}_t$ and $\mathbf{z}_t^{load}$, and $D = dim(\mathbf{I}_t)$ is the dimension[1] of $\mathbf{I}_t$.

Note that our problem can be proved to be NP-hard even in the offline setting (i.e., all inputs over time are known at once and all decisions for all time slots are made at once), due to its discrete variables and its connection to the minimum knapsack problem. We omit this proof due to the page limit.

III. Online Algorithm Design

Our intuition is that, in each time slot, we should "learn" from the cost incurred by the online decision just made, and seek to make a better decision in the next time slot. We design a novel polynomial-time Online Algorithm for Edge Inference ($OAEI$) with two components: an online learning component that overcomes the obliviousness to the uncertain user queries and returns fractional decisions based on previously observable

[1]In this paper, decision $\tilde{\mathbf{I}}_t$ and its corresponding domain $\tilde{\mathcal{X}}$ are defined in real domain while decision $\mathbf{I}_t$ and $\mathcal{X}$ are defined in integral domain. $\tilde{\mathcal{X}} = \{[\tilde{\mathbf{x}}_t^\top, \tilde{\mathbf{y}}_t^\top, (\tilde{\mathbf{z}}_t^{load})^\top]^\top | \tilde{\mathbf{x}}_t \in \mathbb{R}_{\geq 0}^{NM}, \tilde{\mathbf{y}}_t \in \mathbb{R}_{\geq 0}^{NN}, \tilde{\mathbf{z}}_t^{load} \in [0,1]^{NM}\}$. For the provisioning scenario, the radius of the convex feasible set $\tilde{\mathcal{X}}$ is bounded, i.e., $||\mathbf{a} - \mathbf{b}|| \leq R, \forall \mathbf{a}, \mathbf{b} \in \tilde{\mathcal{X}}$ by assuming that the resource capacity of edges and the number of queries are both limited, Similarly, $\mathcal{X} = \{[\mathbf{x}_t^\top, \mathbf{y}_t^\top, (\mathbf{z}_t^{load})^\top]^\top | \mathbf{x}_t \in \mathbb{N}^{NM}, \mathbf{y}_t \in \mathbb{N}^{NN}, \mathbf{z}_t^{load} \in \{0,1\}^{NM}\}$.

Algorithm 1 Online Algorithm for Edge Inference ($OAEI$)

Input: Initial decision $\tilde{\mathbf{I}}_1$; Initial update parameter $\boldsymbol{\lambda}_1 = \mathbf{0}$; Proper step sizes α and μ.

1: **for** $t = 1, 2, ..., T$ **do**
2: Obtain $\mathbf{I}_t$ by using ***Randomized Rounding*** on $\tilde{\mathbf{I}}_t$
3: Provision machine learning inference based on $\mathbf{I}_t$.
4: Observe current cost $f_t(\mathbf{I}_t)$ and constraint $\mathbf{g}_t(\mathbf{I}_t)$.
5: Update $\boldsymbol{\lambda}_{t+1}$ according to (5).
6: Update $\tilde{\mathbf{I}}_{t+1}$ according to (4).
7: **end for**

inputs, and a randomized rounding component that converts such fractional decisions into integers.

A. Online Learning Component

We design an alternating "primal-dual" approach. We note that solving the convex problem of

$$\min \textstyle\sum_t f_t(\tilde{\mathbf{I}}_t), s.t. \textstyle\sum_t \mathbf{g}_t(\tilde{\mathbf{I}}_t) \preceq \mathbf{0},\ \mathbf{h}(\tilde{\mathbf{I}}_t) \preceq \mathbf{0},\ \tilde{\mathbf{I}}_t \in \tilde{\mathcal{X}},$$

where $\tilde{\mathbf{I}}_t$ represents the fraction version. Solving such problem is equivalent to solving the convex-concave problem of

$$\min_{\tilde{\mathbf{I}}_t} \max_{\boldsymbol{\lambda}_t} \textstyle\sum_t \left(f_t(\tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_t^\top \mathbf{g}_t(\tilde{\mathbf{I}}_t)\right),\ s.t.\ \mathbf{h}(\tilde{\mathbf{I}}_t) \preceq \mathbf{0}, \tilde{\mathbf{I}}_t \in \tilde{\mathcal{X}},$$

where $\boldsymbol{\lambda}_t \in \mathbb{R}_{\geq 0}^{dim(\mathbf{g}_t(\tilde{\mathbf{I}}_t))}$ is the Lagrange multiplier. To solve this convex-concave problem in an online manner, we consider

$$\mathcal{L}_t(\tilde{\mathbf{I}}, \boldsymbol{\lambda}) := f_t(\tilde{\mathbf{I}}) + \boldsymbol{\lambda}^\top \mathbf{g}_t(\tilde{\mathbf{I}}). \tag{3}$$

Therefore, we can alternate between minimizing $\mathcal{L}_t(\tilde{\mathbf{I}}, \boldsymbol{\lambda}_{t+1})$ with respect to the primal variable $\tilde{\mathbf{I}}$ via a *modified* descent step and maximizing $\mathcal{L}_t(\tilde{\mathbf{I}}_t, \boldsymbol{\lambda})$ with respect to the Lagrange multiplier $\boldsymbol{\lambda}$ via a dual ascent step. Specifically, at time $t+1$, we solve the following problem

$$\begin{aligned} \min_{\tilde{\mathbf{I}} \in \tilde{\mathcal{X}}} & \nabla f_t(\tilde{\mathbf{I}}_t)^\top (\tilde{\mathbf{I}} - \tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}) + \frac{||\tilde{\mathbf{I}} - \tilde{\mathbf{I}}_t||^2}{2\alpha}, \\ & s.t.\ \mathbf{h}(\tilde{\mathbf{I}}) \preceq \mathbf{0}, \end{aligned} \tag{4}$$

to get $\tilde{\mathbf{I}}_{t+1}$, where $\nabla f_t(\tilde{\mathbf{I}}_t)$ is the gradient of primal objective $f_t(\cdot)$ at $\tilde{\mathbf{I}} = \tilde{\mathbf{I}}_t$, and α is a positive step size. We also update the Lagrange multiplier as

$$\boldsymbol{\lambda}_{t+1} = [\boldsymbol{\lambda}_t + \mu \nabla_{\boldsymbol{\lambda}} \mathcal{L}_t(\tilde{\mathbf{I}}_t, \boldsymbol{\lambda}_t)]^+ = [\boldsymbol{\lambda}_t + \mu \mathbf{g}_t(\tilde{\mathbf{I}}_t)]^+, \tag{5}$$

where μ is also a positive step size, and $\nabla_{\boldsymbol{\lambda}} \mathcal{L}_t(\tilde{\mathbf{I}}_t, \boldsymbol{\lambda}_t) = \mathbf{g}_t(\tilde{\mathbf{I}}_t)$ is the gradient of $\mathcal{L}_t(\tilde{\mathbf{I}}_t, \cdot)$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}_t$.

We highlight that, at $t+1$, updating $\boldsymbol{\lambda}_{t+1}$ as in (5) and updating $\tilde{\mathbf{I}}_{t+1}$ as in (4) only requires information from t, which is the key feature of $OAEI$. We also point out that (4) is not a standard but a *modified* descent step that directly penalizes the constraint violation, which facilitates our performance analysis shown later. The first two terms in (4) form an approximation to $\mathcal{L}_t(\tilde{\mathbf{I}}, \boldsymbol{\lambda}_{t+1})$, and the last term is a proximal term.

Our online component is exhibited as Algorithm 1. The dual update of $\boldsymbol{\lambda}_{t+1}$ and the primal update of $\tilde{\mathbf{I}}_{t+1}$ are in Lines 5 and 6, respectively. In order to convert the fractional decisions $\tilde{\mathbf{I}}_t$, $\forall t$ into integers, we propose a randomized rounding component as Algorithm 2, which is described next.

Algorithm 2 Randomized Rounding

Input: Fractional decision $\tilde{\mathbf{I}}_t \in \tilde{\mathcal{X}}$

// **Step 1** rounds $\tilde{\mathbf{y}}_t$ and $\tilde{\mathbf{z}}_t^{load}$.

1: $\mathbf{A}_t = \left[\tilde{\mathbf{y}}_t^\top, (\tilde{\mathbf{z}}_t^{load})^\top\right]^\top$, $\mathbf{A}'_t = \mathbf{A}_t - \lfloor \mathbf{A}_t \rfloor$.
2: **for** Each column c in $\mathbf{A}'_t$ **do**
3: $\mathrm{A}''_{c,t} = 1$ with the probability of $\mathrm{A}'_{c,t}$; otherwise 0.
4: **end for**
5: $\left[\mathbf{y}_t^\top, (\mathbf{z}_t^{load})^\top\right] = \lfloor \mathbf{A}_t \rfloor^\top + \mathbf{A}_t^{''\top}$

// **Step 2** rounds $\tilde{\mathbf{x}}_t$.

6: **for** Each model $m \in \mathcal{M}$ **do**
7: $\mathbf{B}_t = \left[\tilde{x}_{1,m,t}, ..., \tilde{x}_{N,m,t}\right]^\top$.

// **Step 2.1** ensures the sum of $x_{n,m,t}$ is an integer.

8: $k = \mathbf{1}^\top \mathbf{B}_t, \gamma_1 = 1 - \frac{k - \lfloor k \rfloor}{k}, \gamma_2 = 1 + \frac{\lceil k \rceil - k}{k}$.
9: $\mathbf{U}_t^\top = \begin{cases} [\gamma_1 B_{1,t}, ..., \gamma_1 B_{N,t}] \ with\ prob.\ \lceil k \rceil - k; \\ [\gamma_2 B_{1,t}, ..., \gamma_2 B_{N,t}] \ with\ prob.\ k - \lfloor k \rfloor. \end{cases}$

// **Step 2.2** ensures each $x_{n,m,t}$ is an integer.

10: $\mathbf{V}_t = \mathbf{U}_t - \lfloor \mathbf{U}_t \rfloor$.
11: **while** $V_{i,t} \in (0,1) \wedge V_{j,t} \in (0,1)$ **do**
12: $\theta_1 = \min\{1 - V_{i,t}, V_{j,t}\}$, $\theta_2 = \min\{V_{i,t}, 1 - V_{j,t}\}$.
13: $(V_{i,t}, V_{j,t}) = \begin{cases} (V_{i,t} + \theta_1, V_{j,t} - \theta_1) \ with\ prob.\ \frac{\theta_2}{\theta_1 + \theta_2}; \\ (V_{i,t} - \theta_2, V_{j,t} + \theta_2) \ with\ prob.\ \frac{\theta_1}{\theta_1 + \theta_2}. \end{cases}$
14: **end while**
15: $x_{n,m,t} = V_{n,t}, \forall n \in \mathcal{N}$.
16: **end for**
17: Return $\mathbf{I}_t = \left[\mathbf{x}_t^\top, \mathbf{y}_t^\top, (\mathbf{z}_t^{load})^\top\right]^\top$.

B. Randomized Rounding Component

Our randomized rounding algorithm proceeds in two steps as shown in Algorithm 2. The first step rounds $\tilde{\mathbf{y}}_t$ and $\tilde{\mathbf{z}}_t^{load}$ independently, because $\mathbf{h}(\tilde{\mathbf{I}}_t)$ only restricts $\tilde{\mathbf{x}}_t$ in the sub-problem (4). The second step rounds $\tilde{\mathbf{x}}_t$ in a randomized manner without violating $\mathbf{h}(\cdot) \preceq \mathbf{0}$, where in every iteration a pair of fractions are selected and rounded simultaneously.

We elaborate these steps below. The first step treats separately each fractional value, and rounds it to an integer, whose expectation is the fractional value itself after rounding, shown in Lines 1 through 5. More specifically, the fractional values can be split into two parts: the integral part $\lfloor \mathbf{A}_t \rfloor$ and the real part $\mathbf{A}'_t$. The real part is used as the probability of rounding.

Step 2.1 ensures that the sum of $x_{n,m,t}, \forall n \in \mathcal{N}$ equals an integer and that the expectation of each $x_{n,m,t}$ equals its corresponding value, as shown in Lines 8 and 9. This step either decreases each column of $\mathbf{B}_t$ by multiplying $\gamma_1 < 1$ so that the sum of all columns in $\mathbf{U}_t$ is $\lfloor \mathbf{1}^\top \mathbf{B}_t \rfloor$, or increases each column of $\mathbf{B}_t$ by multiplying $\gamma_2 > 1$ so that the sum of all columns in $\mathbf{U}_t$ is $\lceil \mathbf{1}^\top \mathbf{B}_t \rceil$. The probabilities of taking these two choices are $\lceil k \rceil - k$ and $k - \lfloor k \rfloor$, respectively, which can thus ensure $E[\mathbf{U}_t] = \mathbf{B}_t$. Furthermore, given m, and $k \leq \Omega_m, \Omega_m \in \mathbb{Z}$, the sum of $x_{n,m,t}$, i.e., k, increases to at most $\lceil k \rceil$, which also obeys $\lceil k \rceil \leq \Omega_m$, i.e., $\mathbf{h}(\cdot) \preceq \mathbf{0}$ is also kept.

Step 2.2 further rounds the values into integers in a randomized manner, while guaranteeing that the sum of all the

values stay unchanged after rounding, and that the expectation of each randomized integer equals its corresponding value before rounding, as shown in Lines 10 through 16. First, the vector $\mathbf{U}_t$ is split again into the integral part and the real part. Then, we use the real part as the probability to round the columns in pairs into integers, while letting the two fractions compensate each other. Since the sum of all columns is an integer beforehand as a result of the previous step, $\mathbf{V}_t$ can be guaranteed as a vector that only contains 0 and 1 after the loop. The complexity of the inner while loop reaches $O(N^2)$ [24]. Lastly, combing $\mathbf{x}_t$, $\mathbf{y}_t$ and $\mathbf{z}_t^{load}$ together produces the final control decisions $\mathbf{I}_t$. We emphasize that the results of $E[\mathbf{I}_t] = \tilde{\mathbf{I}}_t$ that we get from our rounding algorithm is necessary for our performance analysis later.

IV. Performance Analysis

A. Performance Metrics

We focus on two metrics that measure the performance of an online algorithm: *dynamic regret* and *dynamic fit*. We exhibit that the dynamic regret and the dynamic fit for our algorithms grow only *sub-linearly* along with time.

Dynamic Regret: The dynamic regret is defined as the difference between the long-term objective function value of the online decisions $\{\mathbf{I}_t\}$ that are made without knowing the inputs in each time slot and the long-term objective function value of the optimal decisions $\{\mathbf{I}_t^*\}$ that optimize the objective function in each time slot by observing the corresponding inputs. Both integral and real domains are considered, namely:

$$Reg_T^d := E[\textstyle\sum_{t=1}^T f_t(\mathbf{I}_t)] - \sum_{t=1}^T f_t(\mathbf{I}_t^*), \tag{6a}$$
$$\mathbf{I}_t^* \in arg\min_{\mathbf{I}\in\mathcal{X}} f_t(\mathbf{I}), \quad s.t.\ \mathbf{g}_t(\mathbf{I}_t^*) \preceq \mathbf{0},\ \mathbf{h}(\mathbf{I}_t^*) \preceq \mathbf{0},$$
$$\widetilde{Reg}_T^d := \textstyle\sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) - \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t^*), \tag{6b}$$
$$\tilde{\mathbf{I}}_t^* \in arg\min_{\tilde{\mathbf{I}}\in\tilde{\mathcal{X}}} f_t(\tilde{\mathbf{I}}), \quad s.t.\ \mathbf{g}_t(\tilde{\mathbf{I}}_t^*) \preceq \mathbf{0},\ \mathbf{h}(\tilde{\mathbf{I}}_t^*) \preceq \mathbf{0},$$

where the expectation is introduced due to the randomized rounding component of our online algorithm.

Dynamic Fit: The dynamic fit is defined as the norm of the cumulative violation of the long-term constraints, incurred by the online decisions $\{\mathbf{I}_t\}$. We use the function of $[\cdot]^+$ to capture such violation. Also, both of the integral and real domains are considered as follows:

$$Fit_T^d := ||\big[E[\textstyle\sum_{t=1}^T \mathbf{g}_t(\mathbf{I}_t)]\big]^+||, \forall t: \mathbf{I}_t \in \mathcal{X}, \tag{7a}$$
$$\widetilde{Fit}_T^d := ||[\textstyle\sum_{t=1}^T \mathbf{g}_t(\tilde{\mathbf{I}}_t)]^+||, \forall t: \tilde{\mathbf{I}}_t \in \tilde{\mathcal{X}}. \tag{7b}$$

B. Regret and Fit Analysis

Roadmap: We firstly present Lemmas 1 and 2, via which we connect the dynamic regret and the dynamic fit in the integral domain to those in the real domain. Next, we bound the fit in Theorem 1 and bound the regret in Theorem 2. Last, we show in Corollary 1 that by choosing proper step sizes we can concretize these bounds into sub-linear functions of time.

Lemma 1. *The relationship on dynamic regret and dynamic fit in the domain of integers and reals can be illustrated as*

$$Reg_T^d \le \widetilde{Reg}_T^d, \quad Fit_T^d \le \widetilde{Fit}_T^d. \tag{8}$$

Proof. See Appendix A. □

Assumptions: Before proceeding further, we introduce the following assumptions to facilitate our analysis. These assumptions are very common, and easy to be satisfied.

Assumption 1: $\forall t$, $f_t(\tilde{\mathbf{I}})$ has bounded gradients in $\tilde{\mathcal{X}}$, i.e., $||\nabla f_t(\tilde{\mathbf{I}})|| \le F, \forall \tilde{\mathbf{I}} \in \tilde{\mathcal{X}}$; and $\mathbf{g}_t(\tilde{\mathbf{I}})$ is bounded in $\tilde{\mathcal{X}}$, i.e., $||\mathbf{g}_t(\tilde{\mathbf{I}})|| \le G, \forall \tilde{\mathbf{I}} \in \tilde{\mathcal{X}}$.

Assumption 2: There exists a constant $\varepsilon > 0$, and an interior point $\widehat{\mathbf{I}}_t \in \tilde{\mathcal{X}}$ such that $\forall t, \mathbf{g}_t(\widehat{\mathbf{I}}_t) \le -\varepsilon\mathbf{1}$.

Assumption 3: The slack constant ε in Assumption 2 satisfies $\varepsilon > \overline{V}(\mathbf{g})$, where the point-wise maximal variation of the consecutive constraints is defined as

$$\overline{V}(\mathbf{g}) := \max_t \max_{\tilde{\mathbf{I}}\in\tilde{\mathcal{X}}} ||[\mathbf{g}_{t+1}(\tilde{\mathbf{I}}) - \mathbf{g}_t(\tilde{\mathbf{I}})]^+||. \tag{9}$$

Assumption 1 bounds both primal and dual gradients per slot, which is a very common assumption [25]. Assumption 2 is Slater's condition, which guarantees the existence of a bounded optimal Lagrange multiplier. Assumption 3 implies that the slack constant ε is larger than the maximal variation of the constraints, requiring $\min_{i,t}\max_{\tilde{\mathbf{I}}\in\tilde{\mathcal{X}}}[-g_{i,t}(\tilde{\mathbf{I}})]^+ > \max_t \max_{\tilde{\mathbf{I}}\in\tilde{\mathcal{X}}} ||[\mathbf{g}_{t+1}(\tilde{\mathbf{I}}) - \mathbf{g}_t(\tilde{\mathbf{I}})]^+||$, which is valid when the feasible region defined by $\mathbf{g}_t(\tilde{\mathbf{I}}) \preceq \mathbf{0}$ is large enough, or the trajectory of $\mathbf{g}_t(\tilde{\mathbf{I}})$ is smooth enough across time.

Lemma 2. *Under previous assumptions and the dual variable initialization of $\boldsymbol{\lambda}_1 = \mathbf{0}$, we have the following:*

$$\tfrac{(||\boldsymbol{\lambda}_{t+1}||^2 - ||\boldsymbol{\lambda}_t||^2)}{2} \le \mu\boldsymbol{\lambda}_t^\top \mathbf{g}_t(\tilde{\mathbf{I}}_t) + \tfrac{\mu^2}{2}||\mathbf{g}_t(\tilde{\mathbf{I}}_t)||^2, \tag{10a}$$
$$\forall t, ||\boldsymbol{\lambda}_t|| \le ||\bar{\boldsymbol{\lambda}}|| := \mu G + \tfrac{2FR + R^2/(2\alpha) + (\mu G^2)/2}{\varepsilon - \overline{V}(\mathbf{g})}. \tag{10b}$$

Proof. See Appendix B. □

Theorem 1. *Under previous assumptions and the dual variable initialization of $\boldsymbol{\lambda}_1 = \mathbf{0}$, the integral dynamic fit in (7a) is upper-bounded:*

$$Fit_T^d \le \widetilde{Fit}_T^d \le \tfrac{\boldsymbol{\lambda}_{T+1}}{\mu} \le \tfrac{||\bar{\boldsymbol{\lambda}}||}{\mu}. \tag{11}$$

Proof. See Appendix C. □

Theorem 2. *Under previous assumptions and the dual variable initialization of $\boldsymbol{\lambda}_1 = \mathbf{0}$, the integral dynamic regret in (6a) is upper-bounded:*

$$Reg_T^d \le \widetilde{Reg}_T^d \le \mathcal{R}_T,$$

where

$$\mathcal{R}_T = \frac{R \cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} + \frac{\alpha F^2 T}{2} + \frac{\mu G^2(T+1)}{2} + \frac{R^2}{2\alpha} + ||\bar{\boldsymbol{\lambda}}|| V(\{\mathbf{g}_t\}_{t=1}^T), \tag{12a}$$
$$V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T) := \textstyle\sum_{t=1}^T \underbrace{||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t-1}^*||}_{V(\tilde{\mathbf{I}}_t^*)}, \tag{12b}$$
$$V(\{\mathbf{g}_t\}_{t=1}^T) := \textstyle\sum_{t=1}^T \underbrace{\max_{\tilde{\mathbf{I}}\in\tilde{\boldsymbol{\mathcal{X}}}} ||[\mathbf{g}_{t+1}(\tilde{\mathbf{I}}) - \mathbf{g}_t(\tilde{\mathbf{I}})]^+||}_{V(\mathbf{g}_t)}. \tag{12c}$$

Proof. See Appendix D. □

Corollary 1. *Under previous assumptions and initialization, dynamic regret and fit are bounded by controlling step sizes:*

$$\alpha = \mu = \max\{\sqrt{\frac{V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{T}}, \sqrt{\frac{V(\{\mathbf{g}_t\}_{t=1}^T)}{T}}\},$$
$$Reg_T^d = \mathcal{O}(\max\{\sqrt{V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)T}, \sqrt{V(\{\mathbf{g}_t\}_{t=1}^T)T}\}),$$
$$Fit_T^d \leq \frac{||\bar{\boldsymbol{\lambda}}||}{\mu} = \mathcal{O}(\max\{\frac{T}{V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}, \frac{T}{V(\{\mathbf{g}_t\}_{t=1}^T)}\}).$$

Following this corollary, if we set the step sizes as

$$\alpha = \mu = \mathcal{O}(T^{-\frac{1}{3}}), \tag{13}$$

then the dynamic regret and the dynamic fit can be bounded, respectively, by

$$Reg_T^d = \mathcal{O}(\max\{V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)T^{\frac{1}{3}}, V(\{\mathbf{g}_t\}_{t=1}^T)T^{\frac{1}{3}}, T^{\frac{2}{3}}\}),$$
$$Fit_T^d = \mathcal{O}(T^{\frac{2}{3}}). \tag{14}$$

V. Experimental Study

A. Data and Settings

Edge, Inference Workload, and Processing: We use the dynamic passenger numbers at the 268 underground stations of London [13] to represent the workload originated from that station. Such passenger data is measured for every quarter (15 minutes) for four days around Nov. 16, 2016. Thus, we consider a four-day period of 384 quarters or time slots. We assume every passenger issues 1∼20 inference queries to the nearby access point colocated at the station. Without loss of generality, each instance of machine learning models has its accuracy loss, ranging 10%∼90%, and each instance processes 1000∼5000 queries per time slot.

Machine Learning Models, Resource, and Usage: The typical size of a machine learning model can be of hundreds of MBs, and the size variation can be up to tens of times [9]. We set the model size as 100∼1000 MB and set the transmission budget, according to real network bandwidths of edges [26, 27], as 1000∼2000 KB/s. The computing capacity for each edge is randomly configured as 80∼300 [27]. Due to the fact that the resource consumed by different models are quite different, we set the resource consumption as 1∼20 [28].

Algorithms and Metrics: Except for the online schema we proposed, i.e., OAEI with $\alpha = \mu = 0.15$, i.e., $\mathcal{O}(384^{-\frac{1}{3}})$ according to the corollary mentioned before, we also compare our schema with multiple step sizes and other algorithms:

- $\boldsymbol{FullUse}$ fully uses edge resource for the model with highest performance-price ratio, which is defined as its process ability dividing its consumption on resource;
- $\boldsymbol{Equally}$ assigns equal amount of queries to each model based on the query number in previous time slot;
- $\boldsymbol{MaxUtility}$ only chooses the most valuable model with highest performance-price ratio within each edge, and switches on delicate calculated number of instances to cover the query number in previous time slot.

All algorithms run online, and will not obtain the actual query numbers before provisioning any instance in each time slot.

B. Evaluation Results

Fig. 2(a) shows the normalized cost, i.e., the total accuracy loss of the system, per time slot for all the algorithms. $OAEI$ with the step size of $\alpha = \mu = 0.15$ reduces at least 24.1% cost on average, compared to other strategies. Further, the dynamic changes of the cost per time slot of $OAEI$ are more stable. Although $Equally$ and $MaxUtility$ have lower costs than $OAEI$ at some time slots, the changes of their costs are quite severe compared to that of $OAEI$, with larger peaks. We point out that the computation overhead of $OAEI$ is only several seconds for thousands of variables, which is acceptable for 15-minute time intervals and hundreds of edges. It is better than other optimization approaches such as linear programming and Newton's method which often need several minutes.

Fig. 2(b) depicts the dynamic regret and the dynamic fit for all the algorithms. Both of the dynamic regret and fit of $OAEI$ perform the best compared with other strategies, gaining at least 30.0% and 34.3% reduction on the means, respectively. This figure also visualizes the sub-linear growth of the dynamic regret and the dynamic fit of $OAEI$, aligned with our theoretical analysis. $OAEI$ updates the deployment of the instances for different models only based on users' queries in each previous time slot, and maintains a well balance between both sides, i.e., the objective and the constraints.

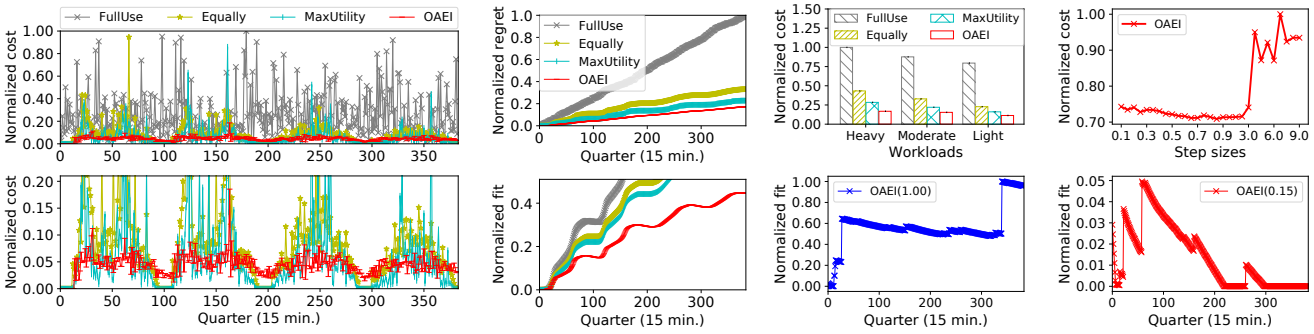
Fig. 2(c) illustrates the results for diverse workloads and in various settings. $OAEI$ is the best for all the three workloads, whose average number of queries per passenger are 5, 10 and 20, respectively; it also gains at least 40.8%, 30.0% and 29.6% reduction in terms of the mean cost, respectively. This figure also shows the mean cost of $OAEI$ under various step sizes, illustrating that the step sizes chosen from our online schema actually performs well compared with others. The figure further shows the impact of step sizes on the dynamic fit. $OAEI$ with small step sizes prefers to update at a fast speed to shorten the violation on constraints while $OAEI$ with large step sizes updates at a mild speed.

VI. Related Work

We summarize prior research in two categories, and highlight their drawbacks compared to our work, respectively.

Machine Learning Inference at Edge: Facebook [2] introduced their study on bringing machine learning inference to the edge, presenting both opportunities and design challenges. Ogden *et al.* [15] proposed a novel mobile deep inference platform that delivered good inference performance. Hu *et at.* [16] minimized the overall delay for partitioned deep inference at edges. Gobieski *et al.* [17] designed and implemented an intermittence-aware software system with specialized support for edge inference. Chinchali *et al.* [18] proposed a distributed DNN architecture for varying network bandwidths between the edge and the cloud. Jiang *et al.* [19] proposed a two-stage pipeline that optimized deep learning on target edge devices.

These works often optimize and execute machine learning inference in individual devices/systems, and rarely study inference optimization at distributed edge computing infrastructures from a *service* perspective for large-scale users. In contrast, we



(a) Cost per time slot (b) Dynamic regret and fit (c) Impact of workloads and step sizes

Fig. 2: Results of London underground traces

treat edge inference service provisioning in an online setting and design algorithms with rigorously provable performance.

Online Service Provisioning at Edge: Wang *et al.* [22] deployed service entities at edges online to facilitate mobile applications and edge cloud providers. Xu *et al.* [20] proposed online service caching and offloading for stochastic inputs. Gao *et al.* [21] proposed an online iteration-based algorithm for access selection and service placement at edges, but failed to consider the uncertainty of users' queries. The long-term effect of instantaneous violation was also studied in [10, 23], where online algorithms with sub-linear static/dynamic regret and accumulated constraint violation were developed, but they failed to consider the integral decision for machine learning inference provisioning among edges.

These works focus on service provisioning, but are often not about machine learning inference; regardless, their algorithmic techniques are insufficient for addressing the challenges in our work. Few of these works consider integral online decisions. Further, we have addressed the time-coupled ramp constraints and the queueing state-transition constraints in an online learning setting with bounded long-term constraints violation.

VII. Conclusion

Provisioning machine learning inference as a service over the mobile edge computing infrastructures for large-scale distributed users is an important step towards realizing universal artificial intelligence. We model an online non-linear integer program to maximize the edge service's overall inference accuracy, subject to the challenging constraints of time-coupling restrictions, obliviousness to uncertain inputs, and integral decisions. We design an online algorithm that consists of an online learning component and a randomized rounding component to overcome these challenges, and rigorously prove the sub-linear dynamic regret and dynamic fit of our approach. We also validate the practical superiority of our approach via trace-driven evaluations and comparison to other algorithms.

Appendix

A. Proof of Lemma 1

Proof. Dynamic regret Reg_T^d can be treated as

$$E[\sum_{t=1}^T f_t(\mathbf{I}_t)] - \sum_{t=1}^T f_t(\mathbf{I}_t^*) \overset{(15a)}{=} \sum_{t=1}^T f_t(E[\mathbf{I}_t]) - \sum_{t=1}^T f_t(\mathbf{I}_t^*)$$
$$\overset{(15b)}{=} \sum_{t=1}^T f_t(E[\mathbf{I}_t]) - \sum_{t=1}^T f_t(\mathbf{I}_t^*) + (\sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) - \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t))$$
$$\overset{(15c)}{=} \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) - \sum_{t=1}^T f_t(\mathbf{I}_t^*) + \sum_{t=1}^T f_t(E[\mathbf{I}_t]) - \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t)$$
$$\overset{(15d)}{\leq} \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) - \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t^*) = \widetilde{Reg}_T^d, \quad (15)$$

where the equation (15a) holds since the expectation as well as f_t is linear; the equation (15b) holds since we add two temporary items whose sum is 0; the equation (15c) holds since we re-arrange the terms, and the inequality (15d) holds since the optimum in reals is lower than the optimum in integers for minimization, and $E[\mathbf{I}_t] = \tilde{\mathbf{I}}_t$ is guaranteed by our delicate designed randomized rounding.

Dynamic fit Fit_T^d can be also treated as follows:

$$||[E[\sum_{t=1}^T \mathbf{g}_t(\mathbf{I}_t)]]^+|| \overset{(16a)}{\leq} ||E[\sum_{t=1}^T \mathbf{g}_t(\mathbf{I}_t)]||$$
$$\overset{(16b)}{=} ||\sum_{t=1}^T \mathbf{g}_t(E[\mathbf{I}_t])|| \overset{(16c)}{=} ||\sum_{t=1}^T \mathbf{g}_t(\tilde{\mathbf{I}}_t)|| = \widetilde{Fit}_T^d, \quad (16)$$

where the inequality (16a) holds because $[]^+$ on each dimension would only decrease the value of its absolute value. Thus, the value of 2-norm increases after we omit $[]^+$. Equation (16b) holds due to the linearity of constraints, and equation (16c) holds also due to our randomized rounding. □

B. Proof of Lemma 2

Proof. Updating $\boldsymbol{\lambda}$ by using the equation in (5), we have

$$||\boldsymbol{\lambda}_{t+1}||^2 = ||[\boldsymbol{\lambda}_t + \mu \mathbf{g}_t(\tilde{\mathbf{I}}_t)]^+||^2 \overset{(17a)}{\leq} ||\boldsymbol{\lambda}_t + \mu \mathbf{g}_t(\tilde{\mathbf{I}}_t)||^2$$
$$= ||\boldsymbol{\lambda}_t||^2 + 2\mu \boldsymbol{\lambda}_t^\top \mathbf{g}_t(\tilde{\mathbf{I}}_t) + \mu^2 ||\mathbf{g}_t(\tilde{\mathbf{I}}_t)||^2, \quad (17)$$

where inequality (17a) holds with the same reason as inequality (16a). After re-arranging terms in (17), we obtain (10a). Since $\tilde{\mathbf{I}}_{t+1}$ is the optimum for objective in (4), by using the interior point $\widehat{\mathbf{I}}_t$ mentioned in Assumption 2, we have

$$\nabla f_t(\tilde{\mathbf{I}}_t)^\top (\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) + \frac{||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2}{2\alpha}$$
$$\leq \nabla f_t(\tilde{\mathbf{I}}_t)^\top (\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\widehat{\mathbf{I}}_t) + \frac{||\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t||^2}{2\alpha}$$
$$\overset{(18a)}{\leq} \nabla f_t(\tilde{\mathbf{I}}_t)^\top (\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t) - \varepsilon \boldsymbol{\lambda}_{t+1}^\top \mathbf{1} + \frac{||\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t||^2}{2\alpha}$$
$$\overset{(18b)}{\leq} \nabla f_t(\tilde{\mathbf{I}}_t)^\top (\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t) - \varepsilon ||\boldsymbol{\lambda}_{t+1}|| + \frac{||\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t||^2}{2\alpha}, \quad (18)$$

where inequality (18a) holds due to Assumption 2, and inequality (18b) holds because $||\boldsymbol{\lambda}_{t+1}||$ is less or equal to $\boldsymbol{\lambda}_{t+1}^\top \mathbf{1}$ for any non-negative vector $\boldsymbol{\lambda}_{t+1}$. Then, we re-arrange the terms in (18) as follows:

$$\boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) \leq \nabla f_t(\tilde{\mathbf{I}}_t)^\top (\widehat{\mathbf{I}}_t - \tilde{\mathbf{I}}_t) - \nabla f_t(\tilde{\mathbf{I}}_t)^\top (\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t)$$

$$
\begin{aligned}
&-\varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{(||\tilde{\mathbf{I}}_t-\tilde{\mathbf{I}}_t||^2-||\tilde{\mathbf{I}}_{t+1}-\tilde{\mathbf{I}}_t||^2)}{2\alpha} \\
&\overset{(19a)}{\leq} \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\widehat{\mathbf{I}}_t-\tilde{\mathbf{I}}_t) - \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\tilde{\mathbf{I}}_{t+1}-\tilde{\mathbf{I}}_t) - \varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{R^2}{2\alpha} \\
&\overset{(19b)}{\leq} ||\nabla f_t(\tilde{\mathbf{I}}_t)||\,(||\widehat{\mathbf{I}}_t-\tilde{\mathbf{I}}_t|| + ||\tilde{\mathbf{I}}_{t+1}-\tilde{\mathbf{I}}_t||) - \varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{R^2}{2\alpha} \\
&\overset{(19c)}{\leq} 2FR - \varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{R^2}{2\alpha} \overset{def}{=} \Phi_{t+1}, \qquad (19)
\end{aligned}
$$

where inequality (19a) holds since the bounded radius on the domain mentioned in footnote, and $||\tilde{\mathbf{I}}_{t+1}-\tilde{\mathbf{I}}_t||^2 \geq 0$; inequality (19b) holds by using Cauchy-Schwartz inequality twice on the first two terms; and inequality (19c) holds by using the bounded gradient in Assumption 1 and bounded domain. After plugging inequality in (19) into inequality (10a), we have

$$
\begin{aligned}
&\triangle(\boldsymbol{\lambda}_{t+1}) := \tfrac{(||\boldsymbol{\lambda}_{t+1}||^2-||\boldsymbol{\lambda}_t||^2)}{2} \\
&\leq \mu\boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) + \tfrac{\mu^2}{2}||\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1})||^2 \\
&\overset{(20a)}{\leq} \mu\boldsymbol{\lambda}_{t+1}^\top(\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) - \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})) + \tfrac{\mu^2G^2}{2} + \Phi_{t+1} \\
&\overset{(20b)}{\leq} \mu\boldsymbol{\lambda}_{t+1}^\top[\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) - \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})]^+ + \tfrac{\mu^2G^2}{2} + \Phi_{t+1} \\
&\overset{(20c)}{\leq} \mu\overline{V}(\mathbf{g})||\boldsymbol{\lambda}_{t+1}|| + \tfrac{\mu^2G^2}{2} + 2FR - \varepsilon||\boldsymbol{\lambda}_{t+1}|| + \tfrac{R^2}{2\alpha}, \qquad (20)
\end{aligned}
$$

where inequality (20a) holds by adding two complementary terms to the right side, i.e., $\pm\boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})$, as well as by using the upper-bound of $\mathbf{g}$; inequality (20b) holds due to the non-negative property of $\boldsymbol{\lambda}_{t+1}$ and the property of $[]^+$; and inequality (20c) holds due to Assumption 3.

Next, we show the correctness of inequality (10b) by contradiction. Without loss of generality, we suppose that $t+2$ is the first time index that breaks inequality (10b), namely:

$$
||\boldsymbol{\lambda}_{t+1}|| \leq ||\bar{\boldsymbol{\lambda}}|| < ||\boldsymbol{\lambda}_{t+2}||. \qquad (21)
$$

However, by using the equation in (5), the relationship can be obtained on $\boldsymbol{\lambda}$ between consecutive time slots as follows:

$$
\begin{aligned}
||\boldsymbol{\lambda}_{t+1}|| &\overset{(22a)}{\geq} ||\boldsymbol{\lambda}_{t+2}|| - ||\boldsymbol{\lambda}_{t+2} - \boldsymbol{\lambda}_{t+1}|| \\
&= ||\boldsymbol{\lambda}_{t+2}|| - ||[\boldsymbol{\lambda}_{t+1} + \mu\mathbf{g}_{t+1}(\mathbf{x}_{t+1})]^+ - \boldsymbol{\lambda}_{t+1}|| \\
&\overset{(22b)}{\geq} ||\boldsymbol{\lambda}_{t+2}|| - ||\boldsymbol{\lambda}_{t+1} + \mu\mathbf{g}_{t+1}(\mathbf{x}_{t+1}) - \boldsymbol{\lambda}_{t+1}|| \\
&= ||\boldsymbol{\lambda}_{t+2}|| - ||\mu\mathbf{g}_{t+1}(\mathbf{x}_{t+1})|| \overset{(22c)}{>} ||\bar{\boldsymbol{\lambda}}|| - \mu G, \qquad (22)
\end{aligned}
$$

where inequality (22a) holds due to the triangle inequality; inequality (22b) holds because of the non-expansive property of the projection, i.e., $[]^+$; and inequality (22c) holds by using the hypothesis on $||\boldsymbol{\lambda}_{t+2}||$ from (21). Then, by plugging (22) into (20), we obtain that $\triangle(\boldsymbol{\lambda}_{t+1}) < 0$, leading to $||\boldsymbol{\lambda}_{t+2}|| < ||\boldsymbol{\lambda}_{t+1}||$, which contradicts (21). Thus, $\forall t$, inequality (10b) holds. □

C. Proof of Theorem 1

Proof. $\boldsymbol{\lambda}$ is updated by using equation in (5), namely:

$$
[\boldsymbol{\lambda}_T + \mu\mathbf{g}_T(\tilde{\mathbf{I}}_T)]^+ \geq ... \geq \boldsymbol{\lambda}_1 + \textstyle\sum_{t=1}^T \mu\mathbf{g}_t(\tilde{\mathbf{I}}_t). \qquad (23)
$$

Since $\boldsymbol{\lambda}_1 = \mathbf{0}$, by re-arranging the terms in (23), we obtain

$$
\textstyle\sum_{t=1}^T \mathbf{g}_t(\tilde{\mathbf{I}}_t) \leq \frac{\boldsymbol{\lambda}_{T+1}}{\mu} - \frac{\boldsymbol{\lambda}_1}{\mu} \leq \frac{\boldsymbol{\lambda}_{T+1}}{\mu}. \qquad (24)
$$

Therefore, $\widetilde{Fit}_T^d = ||[\sum_{t=1}^T \mathbf{g}_t(\tilde{\mathbf{I}}_t)]^+||$ can be treated as

$$
\widetilde{Fit}_T^d \overset{(25a)}{\leq} ||\textstyle\sum_{t=1}^T \mathbf{g}_t(\tilde{\mathbf{I}}_t)|| \leq ||\frac{\boldsymbol{\lambda}_{T+1}}{\mu}|| \leq \tfrac{||\bar{\boldsymbol{\lambda}}||}{\mu}, \qquad (25)
$$

where inequality (25a) holds due to the same reason for (16a). By using (16) again, we complete the proof. □

D. Proof of Theorem 2

Proof. The objective in (4) implies that it is $1/\alpha$-strongly convex with respect to $\tilde{\mathbf{I}}$, denoted by $J_t(\tilde{\mathbf{I}})$, i.e., $\forall \mathbf{a}, \mathbf{b} \in \tilde{\mathcal{X}}$:

$$
J_t(\mathbf{b}) \geq J_t(\mathbf{a}) + \nabla J_t(\mathbf{a})^\top(\mathbf{b}-\mathbf{a}) + \tfrac{||\mathbf{b}-\mathbf{a}||^2}{2\alpha}. \qquad (26)
$$

Since $\tilde{\mathbf{I}}_{t+1}$ is the optimum for $\min_{\tilde{\mathbf{I}}\in\tilde{\mathcal{X}}} J_t(\tilde{\mathbf{I}})$, then we have

$$
\nabla J_t(\tilde{\mathbf{I}}_{t+1})^\top(\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}) \geq 0. \qquad (27)
$$

Thus, by setting $\mathbf{a} = \tilde{\mathbf{I}}_{t+1}, \mathbf{b} = \tilde{\mathbf{I}}_t^*$, as well as plugging inequality (27) into inequality (26), we have

$$
J_t(\tilde{\mathbf{I}}_t^*) \geq J_t(\tilde{\mathbf{I}}_{t+1}) + \tfrac{1}{2\alpha}||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2. \qquad (28)
$$

After adding $f_t(\tilde{\mathbf{I}}_t)$ on both two sides, expanding $J_t(\cdot)$ according to its definition, i.e., the objective in (4), as well as using the property of convex function on $f_t(\cdot)$, i.e., $f_t(\tilde{\mathbf{I}}_t^*) \geq f_t(\tilde{\mathbf{I}}_t) + \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t)$, we have

$$
\begin{aligned}
&f_t(\tilde{\mathbf{I}}_t) + \nabla f_t(\tilde{\mathbf{I}}_t)^\top(\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) + \tfrac{||\tilde{\mathbf{I}}_{t+1}-\tilde{\mathbf{I}}_t||^2}{2\alpha} \\
&\leq f_t(\tilde{\mathbf{I}}_t^*) + \boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_t(\tilde{\mathbf{I}}_t^*) + \tfrac{||\tilde{\mathbf{I}}_t^*-\tilde{\mathbf{I}}_t||^2}{2\alpha} - \tfrac{||\tilde{\mathbf{I}}_t^*-\tilde{\mathbf{I}}_{t+1}||^2}{2\alpha} \\
&\overset{(29a)}{\leq} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{||\tilde{\mathbf{I}}_t^*-\tilde{\mathbf{I}}_t||^2}{2\alpha} - \tfrac{||\tilde{\mathbf{I}}_t^*-\tilde{\mathbf{I}}_{t+1}||^2}{2\alpha}, \qquad (29)
\end{aligned}
$$

where inequality (29a) comes from the fact that $\boldsymbol{\lambda}_{t+1} \succeq \mathbf{0}$ and the per-slot optimal solution $\tilde{\mathbf{I}}_t^*$ is feasible, i.e., $\mathbf{g}_t(\tilde{\mathbf{I}}_t^*) \preceq \mathbf{0}$, such that $\boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_t(\tilde{\mathbf{I}}_t^*) \leq 0$. Then, we analyze the gradient term as

$$
\begin{aligned}
&-\nabla f_t(\tilde{\mathbf{I}}_t)^\top(\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t) \overset{(30a)}{\leq} ||\nabla f_t(\tilde{\mathbf{I}}_t)||\,||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t|| \qquad (30) \\
&\overset{(30b)}{\leq} \tfrac{||\nabla f_t(\tilde{\mathbf{I}}_t)||^2}{2\eta} + \tfrac{\eta}{2}||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2 \overset{(30c)}{\leq} \tfrac{F^2}{2\eta} + \tfrac{\eta}{2}||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2,
\end{aligned}
$$

where η is an arbitrary positive constant. Inequality (30a) holds because of the property of norms; inequality (30b) holds because $a^2 + b^2 \geq 2ab$; and inequality (30c) holds due to the bounded gradient of f_t. After that, we plug inequality (30) into inequality (29) and re-arrange the terms as

$$
\begin{aligned}
&f_t(\tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) \leq f_t(\tilde{\mathbf{I}}_t^*) + (\tfrac{\eta}{2} - \tfrac{1}{2\alpha})||\tilde{\mathbf{I}}_{t+1} - \tilde{\mathbf{I}}_t||^2 \\
&\quad + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) + \tfrac{F^2}{2\eta} \\
&\overset{(31a)}{=} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) + \tfrac{\alpha F^2}{2}, \qquad (31)
\end{aligned}
$$

where inequality (31a) holds because η is chosen, i.e., $\eta = 1/\alpha$, such that $(\frac{\eta}{2} - \frac{1}{2\alpha}) = 0$. By applying (31) into (10a), we have

$$
\begin{aligned}
&\tfrac{\triangle(\boldsymbol{\lambda}_{t+1})}{\mu} + f_t(\tilde{\mathbf{I}}_t) \overset{(32a)}{\leq} \boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) + \tfrac{\mu}{2}||\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1})||^2 \\
&\quad + f_t(\tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) - \boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) \\
&\overset{(32b)}{=} f_t(\tilde{\mathbf{I}}_t) + \boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) + \tfrac{\mu}{2}||\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1})||^2 \\
&\quad + \boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) - \boldsymbol{\lambda}_{t+1}^\top\mathbf{g}_t(\tilde{\mathbf{I}}_{t+1}) \\
&\overset{(32c)}{\leq} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) + \tfrac{\alpha F^2}{2} \\
&\quad + \tfrac{\mu}{2}||\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1})||^2 + \boldsymbol{\lambda}_{t+1}^\top(\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) - \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})) \\
&\overset{(32d)}{\leq} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) + \tfrac{\alpha F^2}{2} \\
&\quad + \tfrac{\mu G^2}{2} + \boldsymbol{\lambda}_{t+1}^\top[\mathbf{g}_{t+1}(\tilde{\mathbf{I}}_{t+1}) - \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})]^+ \\
&\overset{(32e)}{\leq} f_t(\tilde{\mathbf{I}}_t^*) + \tfrac{1}{2\alpha}(||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) + \tfrac{\alpha F^2}{2}
\end{aligned}
$$

$$+\frac{\mu G^2}{2} + ||\boldsymbol{\lambda}_{t+1}||V(\mathbf{g}_t), \tag{32}$$

where inequality (32a) holds because we add the term $f_t(\tilde{\mathbf{I}}_t)$ on both two sides based on (10a) as well as two complementary terms, i.e., $\pm\boldsymbol{\lambda}_{t+1}^\top \mathbf{g}_t(\tilde{\mathbf{I}}_{t+1})$; equation (32b) holds because we re-arrange the terms; inequality (32c) holds due to the application of inequality (31); inequality (32d) holds due to the bounded value of $\mathbf{g}_{t+1}$ as well as the property of $[]^+$; and inequality (32e) holds based on Assumption 3. Next, we consider the intermediate terms as follows:

$$\begin{aligned}
&||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 \overset{(33a)}{=} ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_t||^2 - ||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2 + ||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2 \\
&\overset{(33b)}{=} ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t-1}^*||\,||\tilde{\mathbf{I}}_t^* - 2\tilde{\mathbf{I}}_t + \tilde{\mathbf{I}}_{t-1}^*|| + ||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2 \\
&\overset{(33c)}{\leq} 2R||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t-1}^*|| + ||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2,
\end{aligned} \tag{33}$$

where equation (33a) holds because we add two complementary terms; equation (33b) holds because we apply difference of two squares on the first two terms; and inequality (33c) holds due to the triangle inequality for vectors and the bounded radius on domain. Applying inequality (33) to (32), we have

$$\begin{aligned}
&\frac{\triangle(\boldsymbol{\lambda}_{t+1})}{\mu} + f_t(\tilde{\mathbf{I}}_t) \leq f_t(\tilde{\mathbf{I}}_t^*) + ||\boldsymbol{\lambda}_{t+1}||V(\mathbf{g}_t) + \frac{\alpha F^2}{2} + \frac{\mu G^2}{2} \\
&+\frac{1}{2\alpha}(2R||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t-1}^*|| + ||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2).
\end{aligned}$$

Summing up previous inequality over $t = 1$ to T, we have

$$\begin{aligned}
&\textstyle\sum_{t=1}^T \frac{\triangle(\boldsymbol{\lambda}_{t+1})}{\mu} + \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) \leq \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t^*) + \frac{\alpha F^2 T}{2} + \frac{\mu G^2 T}{2} \\
&\quad + \frac{R\cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} + \textstyle\sum_{t=1}^T\{||\boldsymbol{\lambda}_{t+1}||V(\mathbf{g}_t)\} \\
&\quad + \frac{1}{2\alpha}\textstyle\sum_{t=1}^T(||\tilde{\mathbf{I}}_t - \tilde{\mathbf{I}}_{t-1}^*||^2 - ||\tilde{\mathbf{I}}_t^* - \tilde{\mathbf{I}}_{t+1}||^2) \\
&\overset{(34a)}{\leq} \textstyle\sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t^*) + \frac{\alpha F^2 T}{2} + \frac{\mu G^2 T}{2} + \frac{R\cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} \\
&\quad + ||\bar{\boldsymbol{\lambda}}||\textstyle\sum_{t=1}^T V(\mathbf{g}_t) + \frac{1}{2\alpha}(||\tilde{\mathbf{I}}_1 - \tilde{\mathbf{I}}_0^*||^2 - ||\tilde{\mathbf{I}}_T^* - \tilde{\mathbf{I}}_{T+1}||^2) \\
&\overset{(34b)}{\leq} \textstyle\sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t^*) + \frac{\alpha F^2 T}{2} + \frac{\mu G^2 T}{2} + \frac{R\cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} \\
&\quad + ||\bar{\boldsymbol{\lambda}}||V(\{\mathbf{g}_t\}_{t=1}^T) + \frac{1}{2\alpha}(||\tilde{\mathbf{I}}_1 - \tilde{\mathbf{I}}_0^*||^2),
\end{aligned} \tag{34}$$

where inequality (34a) holds due to the definition of $||\bar{\boldsymbol{\lambda}}||$ and (12c), and inequality (34b) holds also due to (12c). Then,

$$\begin{aligned}
&\widetilde{Reg}_T^d = \textstyle\sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t) - \sum_{t=1}^T f_t(\tilde{\mathbf{I}}_t^*) \leq \frac{\alpha F^2 T}{2} + ||\bar{\boldsymbol{\lambda}}||V(\{\mathbf{g}_t\}_{t=1}^T) \\
&\quad + \frac{\mu G^2 T}{2} + \frac{R\cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} + \frac{||\tilde{\mathbf{I}}_1 - \tilde{\mathbf{I}}_0^*||^2}{2\alpha} - \textstyle\sum_{t=1}^T \frac{\triangle(\boldsymbol{\lambda}_{t+1})}{\mu} \\
&= \frac{\alpha F^2 T}{2} + ||\bar{\boldsymbol{\lambda}}||V(\{\mathbf{g}_t\}_{t=1}^T) + \frac{\mu G^2 T}{2} + \frac{R\cdot V(\{\tilde{\mathbf{I}}_t^*\}_{t=1}^T)}{\alpha} \\
&\quad + \frac{||\tilde{\mathbf{I}}_1 - \tilde{\mathbf{I}}_0^*||^2}{2\alpha} - \frac{||\boldsymbol{\lambda}_{T+2}||^2}{2\mu} + \frac{||\boldsymbol{\lambda}_2||^2}{2\mu} \overset{(35a)}{\leq} \mathcal{R}_T,
\end{aligned} \tag{35}$$

where inequality (35a) holds because $||\tilde{\mathbf{I}}_1 - \tilde{\mathbf{I}}_0^*||^2$ has been bounded by R according to bounded radius of domain, $||\boldsymbol{\lambda}_{T+2}||^2 \geq 0$, as well as $||\boldsymbol{\lambda}_2||^2 \leq \mu^2 G^2$ if $\boldsymbol{\lambda}_1 = \mathbf{0}$. □

ACKNOWLEDGMENT

This work is partially supported by the National Key R&D Program of China under Grant No. 2017YFB1001801, NSFC under Grant No. 61872175, and the Natural Science Foundation of Jiangsu Province under Grant No. BK20181252. This work is also partially supported by the Fundamental Research Funds for the Central Universities under Grant No. 14380060, Nanjing University Innovation and Creative Program for PhD Candidate under Grant No. CXCY19-25, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] "AWS DeepLens," https://aws.amazon.com/deeplens/, 2020.
[2] C.-J. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jia, B. Jia *et al.*, "Machine learning at facebook: Understanding inference at the edge," in *IEEE HPCA*, 2019.
[3] A. Vulimiri, C. Curino, P. B. Godfrey, T. Jungblut, J. Padhye, and G. Varghese, "Global analytics in the face of bandwidth and regulatory constraints," in *USENIX NSDI*, 2015.
[4] Y. Kim, J. Kim, D. Chae, D. Kim, and J. Kim, "μlayer: Low latency on-device inference using cooperative single-layer acceleration and processor-friendly quantization," in *ACM EuroSys*, 2019.
[5] N. D. Lane, S. Bhattacharya, P. Georgiev, C. Forlivesi, and F. Kawsar, "Accelerated deep learning inference for embedded and wearable devices using deepx," in *ACM MobiSys*, 2016.
[6] E. Li, Z. Zhou, and X. Chen, "Edge intelligence: On-demand deep learning model co-inference with device-edge synergy," in *ACM SIGCOMM Workshops*, 2018.
[7] H. Zhang, G. Ananthanarayanan, P. Bodik, M. Philipose, P. Bahl, and M. J. Freedman, "Live video analytics at scale with approximation and delay-tolerance." in *USENIX NSDI*, 2017.
[8] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *IEEE ICDCS*, 2017.
[9] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," in *ICLR*, 2016.
[10] K. Cai, X. Liu, Y.-Z. J. Chen, and J. C. Lui, "An online learning approach to network application optimization with guarantee," in *IEEE INFOCOM*, 2018.
[11] L. Jiao, L. Pu, L. Wang, X. Lin, and J. Li, "Multiple granularity online control of cloudlet networks for edge computing," in *IEEE SECON*, 2018.
[12] "Kaggle: YouTube Dataset from Users in 2018 about Online View," https://www.kaggle.com/nnqkfdjq/statistics-observation-of-random-youtube-video/, 2018.
[13] "Our open data - Transport for London," https://tfl.gov.uk/info-for/open-data-users/our-open-data/, 2020.
[14] T. Chen and G. B. Giannakis, "Bandit convex optimization for scalable and dynamic iot management," in *IEEE IoT-J*, 2018.
[15] S. S. Ogden and T. Guo, "Modi: Mobile deep inference made efficient by edge computing," in *USENIX HotEdge*, 2018.
[16] C. Hu, W. Bao, D. Wang, and F. Liu, "Dynamic adaptive dnn surgery for inference acceleration on the edge," in *IEEE INFOCOM*, 2019.
[17] G. Gobieski, B. Lucia, and N. Beckmann, "Intelligence beyond the edge: Inference on intermittent embedded systems," in *ACM ASPLOS*, 2019.
[18] S. P. Chinchali, E. Cidon, E. Pergament, T. Chu, and S. Katti, "Neural networks meet physical networks: Distributed inference between edge devices and the cloud," in *ACM HotNets*, 2018.
[19] Z. Jiang, T. Chen, and M. Li, "Efficient deep learning inference on edge devices," in *ACM SysML*, 2018.
[20] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *IEEE INFOCOM*, 2018.
[21] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *IEEE INFOCOM*, 2019.
[22] L. Wang, L. Jiao, J. Li, J. Gedeon, and M. Mühlhäuser, "Moera: Mobility-agnostic online resource allocation for edge computing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1843–1856, 2019.
[23] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," in *IEEE TAC*, 2017.
[24] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," in *JACM*, 2006.
[25] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," in *IEEE TSP*, 2015.
[26] Y. Jin, Z. Qian, S. Guo, S. Zhang, X. Wang, and S. Lu, "ran-gjs: Orchestrating data analytics for heterogeneous geo-distributed edges," in *ACM ICPP*, 2018.
[27] C.-C. Hung, G. Ananthanarayanan, L. Golubchik, M. Yu, and M. Zhang, "Wide-area analytics with multiple resources," in *ACM EuroSys*, 2018.
[28] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," in *IEEE COMST*, 2019.