Online Allocation with Replenishable Budgets: Worst Case and Beyond

JIANYI YANG, University of California, Riverside, United States
PENGFEI LI, University of California, Riverside, United States
MOHAMMAD J. ISLAM, University of California, Riverside, United States
SHAOLEI REN, University of California, Riverside, United States

This paper studies online resource allocation with replenishable budgets, where budgets can be replenished on top of the initial budget and an agent sequentially chooses online allocation decisions without violating the available budget constraint at each round. We propose a novel online algorithm, called OACP (Opportunistic Allocation with Conservative Pricing), that conservatively adjusts dual variables while opportunistically utilizing available resources. OACP achieves a bounded asymptotic competitive ratio in adversarial settings as the number of decision rounds T gets large. Importantly, the asymptotic competitive ratio of OACP is optimal in the absence of additional assumptions on budget replenishment. To further improve the competitive ratio, we make a mild assumption that there is budget replenishment every $T^* \geq 1$ decision rounds and propose OACP+ to dynamically adjust the total budget assignment for online allocation. Next, we move beyond the worst-case and propose LA-OACP (Learning-Augmented OACP/OACP+), a novel learning-augmented algorithm for online allocation with replenishable budgets. We prove that LA-OACP can improve the average utility compared to OACP/OACP+ when the ML predictor is properly trained, while still offering worst-case utility guarantees when the ML predictions are arbitrarily wrong. Finally, we run simulation studies of sustainable AI inference powered by renewables, validating our analysis and demonstrating the empirical benefits of LA-OACP.

CCS Concepts: • **Theory of computation → Online algorithms**.

Additional Key Words and Phrases: Online Allocation, Replenishable Budget, Learning-Augmented Algorithm

ACM Reference Format:
Jianyi Yang, Pengfei Li, Mohammad J. Islam, and Shaolei Ren. 2024. Online Allocation with Replenishable Budgets: Worst Case and Beyond. *Proc. ACM Meas. Anal. Comput. Syst.* 8, 1, Article 4 (March 2024), 34 pages. https://doi.org/10.1145/3639030

1 INTRODUCTION

Online allocation subject to resource (or budget) constraints models a sequential decision-making problem where the agent needs to allocate resources without violating the available budget constraint at each round. It is a central problem of critical importance in numerous applications, such as revenue management, online advertising, computing resource management, among many others. For example, Internet companies need to select advertisements based on online user arrivals subject

Authors' addresses: Jianyi Yang, jyang239@ucr.edu, University of California, Riverside, 900 University Ave., Riverside, California, 92521, United States; Pengfei Li, pli081@ucr.edu, University of California, Riverside, 900 University Ave., Riverside, California, 92521, United States; Mohammad J. Islam, misla056@ucr.edu, University of California, Riverside, 900 University Ave., Riverside, California, 92521, United States; Shaolei Ren, sren@ece.ucr.edu, University of California, Riverside, 900 University Ave., Riverside, California, 92521, United States.

ACM 2476-1249/2024/3-ART4
https://doi.org/10.1145/3639030

to advertisers' budget constraints; cloud operators need to dynamically allocate user requests to available machines subject to resource constraints; and edge devices need to dynamically optimize its battery energy usage while intermittently harvesting energy from the surrounding environment. As such, the problem of online allocation and its variants have received rich attention in the past few decades [9, 11, 31, 40, 55, 59].

Online allocation decisions are temporally coupled due to total budget constraints, thus requiring complete offline information to obtain the optimal solution. Nonetheless, the availability of only online information in practice makes online allocation extremely challenging. To meet budget constraints in online settings, a commonly considered approach is Lagrangian relaxation, which includes weighted budget constraints as a regularizer for online decision making where the weights are dual variables and can be interpreted as the budget/resource price [21, 45, 49, 63]. Consequently, by adjusting the resource price, the agent's budget consumption is also governed so as to meet the budget constraint. For example, there have been a variety of approaches to updating the dual variables online [1, 9, 11, 45, 63].

Despite these efforts and advances in various (relaxed) settings such as stochastic utility functions [9], optimizing the total utility subject to strict budget constraints still remains a challenging problem in *adversarial* settings, where the utility functions can be arbitrarily presented to the agent. In fact, competitive online algorithms for adversarial settings have only been proposed very recently. More concretely, online resource allocation with a single-inventory constraint [41] and a multi-inventory constraint [40] are two of the very few known competitive online algorithms with a finite number of decision rounds under the assumption that the utility functions of each inventory are separable. In [11], an online allocation algorithm that adjusts the dual variable is proposed, achieving a bounded asymptotic competitive ratio in adversarial settings when the length of each problem instance is sufficiently long. Nonetheless, these studies [11, 40, 41] are crucially limited in the following aspects.

- *No budget replenishment.* First and foremost, the total budget constraint is fixed without allowing *replenishment* online [11, 40, 41]. In fact, these algorithms explicitly assume that budgets are *not* replenishable, which would otherwise void their competitive analysis. However, budget replenishment in an online manner is common in practice, e.g., dynamic energy harvesting (see Section 2.3 for additional examples). While some studies [7, 28, 30, 50, 60] have considered budget replenishment, they typically focus on independent and identically distributed budget replenishment. In contrast, arbitrary budget replenishment in adversarial settings naturally provides additional power to the adversary, thus creating significantly more challenges.
- *Worst-case performance only.* Second, the studies [11, 40, 41] only focus on the worst-case performance in terms of the competitive ratio. As a result, the conservativeness needed to address the worst possible problem input significantly limits their average-case performance for most typical problem inputs. Online algorithms based on machine learning (ML) models have been considered for various problems [2, 12, 35, 59], including online resource allocation [23, 24]. Nonetheless, unlike the hand-crafted online algorithms [11, 40, 41], ML-based online optimizers may not offer worst-case performance guarantees and can result in significantly bad results when, for example, the training-testing distribution differs. Even though heuristic techniques such adversarial training can empirically mitigate the lack of performance robustness to some extent, it is still challenging to provably guarantee the worst-case performance of ML models. Thus, it remains an open problem to achieve the *best of both worlds* — improving the average utility while offering the worst-case robustness (in the presence of budget replenishment). In fact, as highlighted above, there even do not exist competitive online algorithms that address budget replenishment in adversarial settings, let alone a learning-augmented algorithm that can improve the average performance while provably offering worst-case performance guarantees.

Algorithm	Budget replenishment	Budget cap	Worst-case robustness	Average utility bound
CR-Pursuit [41]	✘	NA	✔	✘
A&P [40]	✘	NA	✔	✘
DMD [11]	✘	NA	✔	✘
OACP (our work)	✔	✔	✔	✘
OACP+ (our work)	✔	✔	✔	✘
LA-OACP (our work)	✔	✔	✔	✔

Table 1. Comparison between our work and recent online competitive allocation algorithms for adversarial settings. Algorithms for non-adversarial settings are discussed in Section 6 and not shown in the table.

Contributions. In this paper, we address the above points and consider online allocation with replenishable budgets, where the agent receives budget replenishment on the fly and needs to choose irrevocable online decisions to allocate M resources. The goal of the agent is to maximize the total utility over T rounds subject to per-round available budget constraints, where the per-round utility is a function in terms of the online allocation decision.

We first consider an adversarial setting and propose an online algorithm, called OACP (Opportunistic Allocation with Conservative Pricing), that updates the dual variable (i.e., resource pricing) online to regulate the agent's budget allocation and achieves an asymptotic competitive ratio as $T \to \infty$. The key insight of OACP is that we treat the uncertain budget replenishment differently than the initially-assigned fixed budget and set the resource price in a conservative manner, which encourages the agent to be more frugal while still allowing the agent to opportunistically utilize the replenished budgets when applicable. Most importantly, we prove in Theorem 3.1 that OACP achieves the same asymptotic competitive ratio bound as the state-of-the-art optimal bound in [11] that does not address budget replenishment. In our setting with replenishable budgets, the adversary naturally has more power than the setting of a fixed known budget, as it can arbitrarily present budget replenishments to the agent. Therefore, achieving the same asymptotic competitive ratio as that of the state-of-the-art algorithm for fixed budget allocation [11] highlights the benefit of OACP in terms of addressing additional uncertainties of replenished budget.

Next, we propose OACP+ to utilize the budget replenishment more efficiently under a mild assumption that the budget is replenished at least every $T^* \geq 1$ decision rounds. Specifically, OACP+ divides the whole episode of T rounds into K frames of unequal lengths and performs frame-level budget assignment online and a round-level online budget allocation within each frame. To account for the maximum budget cap, a new threshold-based budget assignment strategy is proposed to decide the assigned budget for each frame. Given the assigned budget for each frame, we apply OACP for round-level budget allocation while deferring all the budget replenishment to future frames. We prove that OACP+ achieves a higher asymptotic competitive ratio than OACP if the total budget replenishment is positive in every T^* rounds (Theorem 3.2).

Last but not least, we move beyond the worst-case and aim to maximize the average utility while still offering worst-case utility guarantees. We propose a novel learning-augmented algorithm, called LA-OACP (Learning-Augmented OACP), that integrates a trained ML predictor with OACP. More concretely, LA-OACP utilizes the ML prediction (i.e., online allocation decision by the ML-based optimizer) and expert decision (from OACP or OACP+) as advice, and judiciously combine them. The key novelty of LA-OACP is to introduce a new reservation utility that produces a constrained decision set within which all decisions can meet the worst-case utility constraint (defined with respect to OACP or OACP+). Meanwhile, LA-OACP ensures that the online decisions are chosen from the constrained decision set while being close to ML predictions so as to exploit the benefits of

ML predictions to improve the average utility. We rigorously prove that LA-OACP can improve the average utility compared to OACP when the ML predictor is properly trained, while still offering worst-case utility guarantees (see Theorems 4.1 and 4.2).

Finally, we run simulation studies of sustainable AI inference to maximize the total utility subject to energy constraints with renewable replenishment. Our results validate the analysis of OACP, OACP+ and LA-OACP, demonstrating the empirical advantage of LA-OACP in terms of the average utility over OACP and OACP+ as well as other baseline algorithms.

We highlight the main difference between our algorithms and recent online allocation algorithms that consider adversarial settings in Table 1. *Our major contributions are also summarized as follows.* First, we propose two novel online algorithms OACP and OACP+ that achieve bounded asymptotic competitive ratios for online allocation with replenishable budgets in adversarial settings (**Theorem 3.1 and Theorem 3.2**). To our knowledge, the proposed provably-competitive algorithms advance the existing competitive online algorithms to address budget replenishment in adversarial settings for the first time [11, 40, 41]. Second, we move beyond the worst case and propose a novel learning-augmented algorithm, LA-OACP, that probably improves the average utility compared to OACP or OACP+ (**Theorem 4.2**), while still offering worst-case utility guarantees for online allocation with budget replenishment for any problem instance (**Theorem 4.1**).

2 PROBLEM FORMULATION

In this section, we present the problem formulation for online allocation with replenishable budgets.

Notations: For the convenience of presentation, we first introduce the common notations used throughout the paper. Unless otherwise noted, we use $[N]$ to denote the set $\{1, 2, \cdots, N\}$ for a positive integer N. $\mathbb{E}(\cdot)$ is the expectation operator, $\mathbb{P}$ is a probability measure, $\mathbb{I}(x)$ is an indicator function (i.e., $\mathbb{I}(x) = 1$ if the condition x is true and $\mathbb{I}(x) = 0$ otherwise), and $\mathcal{R}_+^D$ and $\mathcal{R}_{++}^D$ are D-dimensional non-negative and strictly positive real number spaces, respectively. For a vector x, x_j denotes its j-the element and $\|x\|$ is its norm (l_2 norm by default). For two vectors x and y, we use $x \le y$ to denote *element*-wise inequality, i.e., $x_j \le y_j$ for all j and use $x \odot y$ to denote *element*-wise product. $\min(x, y)$ denotes the *element*-wise minimization. We also use $[x]^b = \min(x, b)$ and $[x]^+ = \max(x, 0)$, where the capping and rectifying operators are applied for each element when x is a vector. For a sequence of variables $c_1, \cdots, c_T$, we use $c_{i:j}$ to denote the subsequence $c_i, \cdots, c_j$ for $1 \le i \le j \le T$; we have $c_{i:j} = \emptyset$ if $i > j$.

2.1 Model

We consider an online allocation problem with replenishable resource budgets, where each sequence (a.k.a., problem instance) includes T consecutive rounds and involves sequential allocation of M types of resources based on online information.

At the beginning of a sequence (i.e., round $t = 1$), the decision maker (i.e., agent) is endowed with an initial resource budget $B_1 = [B_{1,1}, \cdots, B_{M,1}] \in \mathcal{R}_{++}^M$, where $B_{m,1} = T\rho_m$ is the initial resource budget for type-m resource, with $\rho_m > 0$ being the per-round average budget initially assigned to the agent, for $m \in [M]$. Moreover, we have $B_1 \le B_{\max}$, where $B_{\max} = [B_{1,\max}, \cdots, B_{M,\max}]$ represents the maximum budget cap. The inclusion of $B_{\max}$ is both practical and general: $B_{\max}$ captures practical constraints such as battery capacity for energy resources, space constraint for product inventory, among others, and the budget cap can be effectively voided when setting a large $B_{m,\max} \to \infty$ for $m \in [M]$, to which case our design also applies.

At the beginning of each round $t \in [T]$, the agent is presented with a utility function $f_t(x) : \mathcal{R}_+^M \to \mathcal{R}_+$, where $x \in \mathcal{X}$ is the allocation decision. Additionally, the agent also receives a potential budget replenishment $\hat{E}_t = \left[\hat{E}_{1,t}, \cdots, \hat{E}_{M,t}\right] \in \mathcal{R}_+^M$, resulting in a total budget of $\min\left(B_t + \hat{E}_t, B_{\max}\right) =$

$B_t + E_t$, available for allocation at round t, where B_t is the remaining budget at the end of round t. In other words, due to the budget cap, the actual budget replenishment is $E_t = \min(\hat{E}_t, B_{\max} - B_t)$ at round t.

The agent's allocation decision for round t is $x_t = [x_{1,t}, \cdots, x_{M,t}] \in \mathcal{X}$, where $\mathcal{X} = \{x \in \mathcal{R}_+^M | 0 \leq x \leq \bar{x}\}$ with $\bar{x} = [\bar{x}_1, \cdots, \bar{x}_M]$ representing the maximum allocation for each resource type at each round. Note that we have $\bar{x} \leq B_{\max}$ since otherwise the maximum budget cap is more stringent while the maximum allocation constraint $\bar{x}$ is never binding.

Given the budget replenishment and the agent's allocation decision, the budget evolves as $B_{t+1} = \min\left(B_t + \hat{E}_t, B_{\max}\right) - x_t = B_t + E_t - x_t$ for round $t+1$. Thus, the information revealed to the agent at the beginning of each round t can be summarized as $y_t = (f_t, \hat{E}_t)$, while all the information for a sequence can be written as $y = [y_1, \cdots, y_T] \in \mathcal{Y}$, where $\mathcal{Y}$ denotes the space of all possible episodic information. When the context is clear, we also use y to denote a sequence. Any remaining budgets at the end of an sequence are wasted without rolling over to the next sequence. If an algorithm π is used to solve the problem with information y, the total the total utility is denoted as $F_T^\pi(y) = \sum_{t=1}^T f_t(x_t)$.

To summarize, for a sequence y, the *offline* problem can be formulated as

$$\max_{x_{1:T} \in \mathcal{X}^T} \sum_{t=1}^{T} f_t(x_t) \tag{1a}$$

$$s.t., \quad x_t \leq B_t + E_t \text{ and } x_t \in \mathcal{X}, \quad \forall t \in [T] \tag{1b}$$

$$B_{t+1} = B_t + E_t - x_t \text{ and } E_t = \min\left(\hat{E}_t, B_{\max} - B_t\right), \quad \forall t \in [T] \tag{1c}$$

Next, we make the following standard assumptions on the utility function $f_t(x)$ for $t \in [T]$.

Assumption 1 (Utility function $f_t(x)$)**.** For any $t \in [T]$, the utility function $f_t(x) : \mathcal{R}_+^M \to \mathcal{R}_+$ is assumed to be non-negative, have subgradients at each point of $x \in \mathcal{X}$. In addition, we assume $f_t(0) = 0$ and $\sup f_t(x) = \bar{f}$ for $t \in [T]$ and $x \in \mathcal{X}$.

The assumptions are standard in the literature on online allocation with budget constraints [9, 11, 40]. Note that we do not require concavity of the utility functions, making our algorithms applicable for a wide range of applications.

2.2 Performance Metrics

With complete information $y = [y_1, \cdots, y_T] \in \mathcal{Y}$ provided to the agent at the beginning of a sequence, the problem in (1) can be efficiently solved via subgradient methods for constrained optimization [13, 16, 27]. If the utility functions are concave, subgradient methods such as the projected subgradient method and the primal-dual subgradient method have provable convergence guarantees [16, 27]. Nonetheless, in practice, the agent only has access to online information $y_{1:t}$ before making its decision x_t at round $t \in [T]$, adding substantial challenges.

Our goal is to design an online algorithm π that maps available online information $y_{1:t}$ to a decision $x_t \in \mathcal{X}$ subject to the budget constraint (1b) at each round $t \in [T]$. To measure the decision quality of an online algorithm π, we use the following metrics that capture the *worst*-case and *average*-case performance, respectively.

Definition 1 (Asymptotic competitive ratio [10, 14])**.** The asymptotic competitive ratio of an online algorithm π is CR^π if $\lim_{T\to\infty} \sup_{y\in\mathcal{Y}} \frac{1}{T}\left(OPT(y) - \frac{1}{CR^\pi} F_T^\pi(y)\right) \leq 0$, where $F_T^\pi(y) = \sum_{t=1}^T f_t(x_t)$ is the total utility of algorithm π and $OPT(y)$ is the optimal utility obtained by the oracle given offline information.

Definition 2 (Average utility)**.** Given an online algorithm π, the average utility is defined as $AVG^{\pi} = \mathbb{E}_{y \in \mathcal{Y}} \left[F_T^{\pi}(y) \right]$, where the expectation is over the sequence information $y \sim \mathbb{P}_y$.

Both competitive ratio and average utility are important in practice, characterizing the robustness of an online algorithm (in terms of its utility ratio to the optimal oracle) and its quality for typical problem instances, respectively. Here, we consider an asymptotic competitive ratio (in the sense of $T \to \infty$) because of the intrinsic hardness of our problem — even for online allocation of a fixed budget without replenishment, only an asymptotic competitive ratio is attainable in the state of the art [11]. We shall design in Section 3 online allocation algorithms to address the worst-case robustness, while we will consider the average performance (subject to a worst-case robustness constraint) in Section 4.

2.3 Application Examples

We now provide a few examples as motivating applications to make our model more concrete.

Online advertising with budget replenishment. Online advertisement serves as a prominent, if not the most prominent, source of revenue for Internet companies [11]. Advertisers need to dynamically set a biding budget, which will then be used by the publisher to maximize profits or the number of impressions for advertisers per their contracts with the publisher. Meanwhile, they can also increase budgets anytime they like. Thus, by viewing the bidding budget as an online decision, this problem fits nicely into the online allocation of replenishable budgets.

Sustainable AI inference. Nowadays, the rapidly increasing demand for artificial intelligence (AI) inference, especially large language models, has resulted in large carbon emissions [43]. To achieve sustainable AI inference, it is important to exploit renewable generation to replenish on-site energy storage. Meanwhile, for the same AI inference service, there often exist multiple models (e.g., eight different GPT-3 models [17]), each having a distinct model size to offer a flexible tradeoff between accuracy performance and energy consumption. However, the renewables are known to be time-varying and unstable. Thus, by viewing the intermittent renewables as replenished budgets, the resource manager needs to schedule an appropriate AI model for inference in an online manner to maximize the utility (e.g. maximizing the accuracy) given available energy constraints [51, 53].

Online inventory management with dynamic replenishment. Manufacturers need to dynamically dispatch available inventory to different distributors based on market demands. Meanwhile, they will also replenish the inventory through newly manufactured products. The goal is to manage the available inventory to maximize the total profit/revenue given dynamic replenishment and environment (e.g., market demands and supply-chain situation), to which our model is well suited.

3 OACP: OPPORTUNISTIC ALLOCATION WITH CONSERVATIVE PRICING

In this section, we address the worst-case robustness in adversarial settings and design an asymptotically competitive online algorithm, called OACP, that conservatively updates the dual variable based on mirror descent and opportunistically allocates replenished budgets. Using a novel technique, OACP provably offers the optimal worst-case performance guarantees for adversarial settings of online allocation with replenishable budgets (Theorem 3.1). Then, by making an additional assumption on the minimum budget replenishment, we extend OACP to OACP+, which offers an improved asymptotic competitive ratio (Theorem 3.2).

To solve the online allocation problem in (1), one can equivalently relax the budget constraints using Lagrangian techniques. More specifically, instead of directly solving (1), we introduce a regularizer and solve $\hat{x}_t = \arg\max_{x \in \mathcal{X}} \{ f_t(x) - \mu_t^{\top} x \}$ where $\mu_t \in \mathcal{R}_+^M$ is the Lagrangian multiplier vector (a.k.a., *dual* variable) with each entry corresponding to one resource budget constraint. The

Algorithm 1 Opportunistic Allocation with Conservative Pricing (OACP)

```
Require: Initialize dual variable $\mu_1$, and budget $B_1 = \rho T$ for $\rho > 0$
 1: for $t = 1$ to $T$ do
 2:     Receive utility function $f_t(x)$ and potential budget replenishment $\hat{E}_t$.
 3:     Get the actual replenished budget $E_t = \min\{\hat{E}_t, B_{\max} - B_t\}$
 4:     Pre-select action $x_t$ based on $\mu_t$: $\hat{x}_t = \arg\max_{x \in \mathcal{X}} \{f_t(x) - \mu_t^\top x\}$
 5:     if $\hat{x}_t \leq B_t + E_t$ then
 6:         $x_t = \hat{x}_t$ and $g_t = -\hat{x}_t + \rho$    //for conservative pricing
 7:     else
 8:         $x_t = 0$ and $g_t = 0$
 9:     end if
10:     Update budget $B_{t+1} = B_t + E_t - x_t$
11:     Dual mirror descent:
        $\mu_{t+1} = \arg\min_{\mu \geq 0} g_t^\top \mu + \frac{1}{\eta} V_h(\mu, \mu_t)$ where $V_h(\mu, \mu_t) = h(\mu) - h(\mu_t) - \nabla h(\mu_t)^\top (\mu - \mu_t)$ is the
        Bregman divergence in which $h(\mu)$ is a $\sigma$-strongly convex reference function (Assumption 2)
12: end for
```

interpretation of μ_t is that it can be viewed as the resource *price* [49, 56]: a higher price encourages resource conservation to meet the budget constraints, and vice versa.

If we were able to optimally set $\mu_t \in \mathcal{R}_+^M$ for $t \in [T]$, we could optimally solve (1) while satisfying the per-round budget constraints. Nonetheless, like in the original problem (1), finding the optimal μ_t for $t \in [T]$ requires the complete offline information $y = [y_1, \cdots, y_T]$ at the beginning of an episode, but this information is clearly lacking for online allocation.

Despite this challenge, the interpretation of the dual variable μ_t as the resource price at round $t \in [T]$ provides us with inspiration for the design of OACP. Specifically, in view of the dynamic budget replenishment E_t, we propose to *conservatively* update the price μ_{t+1} to a higher value for each round $t + 1$ as if E_t does not exist, and then *opportunistically* use the actually available budget $B_t + E_t$. Our algorithm, called OACP, is described in Algorithm 1.

3.1 Competitive Algorithm Design

At each round $t \in [T]$, given μ_t and online information, we solve the following relaxed optimization problem:

$$\hat{x}_t = \arg\max_{x \in \mathcal{X}} \{f_t(x) - \mu_t^\top x\}. \tag{2}$$

Next, we check if $\hat{x}_t$ satisfies the current budget constraint $B_t + E_t$: we set $x_t = \hat{x}_t$ if the budget constraint is satisfied, and $x_t = 0$ otherwise. Then, we update the dual variable based on mirror descent $\mu_{t+1} = \arg\min_{\mu \geq 0} g_t^\top \mu + \frac{1}{\eta} V_h(\mu, \mu_t)$, where $V_h(\mu, \mu_t) = h(\mu) - h(\mu_t) - \nabla h(\mu_t)^\top (\mu - \mu_t)$ is the Bregman divergence defined with respect to a reference function $h(\mu)$.

The goal of mirror descent is to update the dual variable μ_{t+1} such that it can set a resource *price* that reflects the current budget level while staying not too far away from the current dual variable μ_t as regularized by $\frac{1}{\eta} V_h(\mu, \mu_t)$ in terms of Bregman divergence. In particular, the usage of mirror descent to update dual variables for online constrained optimization has begun to be explored recently [7, 9, 11]. Nonetheless, the prior studies on online allocation under adversarial settings have only considered a fixed budget without dynamic budget replenishment [11].

Key insight. The key insight of OACP lies in how we set g_t and choose x_t in Lines 5 and 6 of Algorithm 1. The dual variable μ_t is updated based on $g_t = -\hat{x}_t + \rho$, whose inverse (i.e., $\hat{x}_t - \rho$) measures the overuse of the current allocation compared with a reference per-round budget $\rho = \frac{B_1}{T}$.

When g_t is smaller, the degree of budget overuse is greater, and μ_{t+1} tends to be greater in the mirror descent step, encouraging the agent to use fewer resources at round $t+1$. Under the setting of no budget replenishment, it is natural to set the per-round budget $\rho = \frac{B_1}{T}$ to evaluate the degree of over-consumption for each round. Nonetheless, in the presence of budget replenishment, we cannot simply use $\rho + E_t$ as the reference to incorporate new replenishment E_t in resource pricing. The reason is that the sequence of E_t can be arbitrary and the future replenishment $E_{t+1}, \cdots, E_T$ is unknown. As a result, aggressively using $\tilde{\rho} = \rho + E_t$ as the reference per-round resource consumption can result in an unnecessarily low resource price μ_{t+1}. Instead, OACP still sets the reference per-round budget as $\rho = \frac{B_1}{T}$ as if no budget replenishment were received. Consequently, the resource price μ_{t+1} tends to be higher than using $\rho + E_t$ otherwise, encouraging the agent to be more frugal in resource consumption. On the other hand, the budget replenishment E_t can be still used opportunistically by increasing the actual available budget from B_t to $B_t + E_t$ (Line 5). Thus, by doing so, OACP tends to be *more conservative in resource pricing (i.e., μ_t), while still opportunistically using budget replenishment in actual allocation decisions.*

Next, to make Algorithm 1 self-contained, we specify the following assumptions on the reference function $h(\mu)$ used in the mirror descent step.

Assumption 2 (Reference function $h(\mu)$)**.** The reference function $h(\mu) : \mathcal{R}_+^M \to \mathcal{R}$ is differentiable and σ-strongly convex in $\|\cdot\|_1$-norm in $\mathcal{R}_+^M$, i.e., $h(\mu) - h(\mu') \geq \nabla h(\mu')^\top(\mu - \mu') + \frac{\sigma}{2}\|\mu - \mu'\|_1^2$ for any $\mu, \mu' \in \mathcal{R}_+^M$.

Assumption 2 is standard in the analysis of mirror descent-based algorithms [9, 11]. Along with Assumption 1 on the utility function, it essentially ensures that there is always a unique solution in the mirror descent step in Line 11 of Algorithm 1. Importantly, this step can recover common gradient-based update algorithms by a proper choice of the reference function. For example, with $h(\mu) = \sum_{m=1}^M \mu_m \log(\mu_m)$, the update in Line 11 of Algorithm 1 becomes $\mu_{t+1} = \mu_t \odot \exp(-\eta g_t)$ and captures multiplicative weight updates, where the operator "$\odot$" is the element-wise product [6]; for $h(\mu) = \frac{1}{2}\|\mu\|_2^2$, the update rule becomes $\mu_{t+1} = [\mu_t - \eta g_t]^+$ and recovers online sub-gradient descent method [11].

3.2 Performance Analysis

We proceed with the analysis of OACP in terms of its worst-case performance. Our result highlights that OACP is asymptotically competitive against the offline oracle OPT, generalizing the prior results on the allocation of a fixed budget [11] to replenishable budgets.

Theorem 3.1. *For any episode $y \in \mathcal{Y}$ and $\eta > 0$, by Algorithm 1, the utility of* OACP *satisfies*

$$OPT(y) - \alpha F_T^{\mathsf{OACP}}(y) \leq \alpha \bar{f} + \frac{\alpha\left(\bar{\rho} + \|\bar{x}\|_\infty\right)^2 \eta T}{2\sigma} + \frac{\alpha}{\eta} V_h(\mu, \mu_1), \tag{3}$$

where $\alpha = \max_{m\in[M]} \frac{\bar{x}_m}{\rho_m}$, $\bar{\rho} = \max_{m\in[M]} \rho_m$ is the maximum per-round average budget initially assigned to the agent at round $t = 1$, $\bar{x}$ is the maximum per-round resource allocation constraint, $V_h(\mu, \mu_1)$ is the Bregman divergence between μ and the initial dual variable μ_1 given the σ-strongly convex reference function h, and $\mu = 0$ if Line 5 of Algorithm 1 is always true, and otherwise, $\mu = \frac{\bar{f}}{\alpha\rho_j} e_j$ with $j = \arg\min_{m\in\mathcal{M}_A} V_h(\frac{\bar{f}}{\alpha\rho_m} e_m, \mu_1)$ where $\mathcal{M}_A = \left\{m \mid \exists t \in [T] \text{ such that } \hat{x}_{m,t} > (B_t + E_t)_m\right\}$, e_m is a standard M-dimensional unit vector. Furthermore, by optimally setting $\eta = \frac{1}{\bar{\rho}+\|\bar{x}\|_\infty}\sqrt{2\sigma V_h(\mu, \mu_1)/T}$, we have

$$\lim_{T\to\infty} \sup_{y\in\mathcal{Y}} \frac{1}{T}\left(OPT(y) - \alpha F_T^{\mathsf{OACP}}(y)\right) \leq \lim_{T\to\infty} \frac{1}{T}\left(\alpha\bar{f} + \alpha\left(\bar{\rho} + \|\bar{x}\|_\infty\right)\sqrt{\frac{V_h(\mu, \mu_1)T}{2\sigma}}\right) = 0, \tag{4}$$

i.e., OACP *achieves an asymptotic competitive ratio of* $\frac{1}{\alpha} = \min_{m \in [M]} \frac{\rho_m}{\bar{x}_m}$ *against* OPT.[1]

The proof of Theorem 3.1 is deferred to Appendix A to keep the main body of the paper more concise for better readability. Our proof relies on a technique specifically designed for budget replenishment. Concretely, without budget replenishment, the allocation algorithm (e.g., DMD in [11]) stops allocation whenever any resource type in the initial budget B_1 is exhausted. In contrast, OACP continues allocation until the end of an episode due to new budget replenishment. To account for this, we introduce a group $\mathcal{T}_A$ of rounds that each has a budget violation event, and bound the total utility for rounds that are not in $\mathcal{T}_A$.

Theorem 3.1 can be interpreted as follows. Without optimally setting η, by rearranging the terms in (3), we have $F_T^{\text{OACP}}(y) \geq \frac{1}{\alpha} OPT(y) - \bar{f} - \frac{(\bar{\rho}+\|\bar{x}\|_\infty)^2 \eta T}{2\sigma} - \frac{1}{\eta} V_h(\mu, \mu_1)$. That is, for any episode $y \in \mathcal{Y}$, OACP can obtain a total utility of at least $\frac{1}{\alpha}$ times the optimal oracle's utility, minus per-round utility bound $\bar{f}$ and a term related to the convergence of μ. Moreover, by setting $\eta \sim O(\frac{1}{\sqrt{T}})$, OACP achieves an asymptotic competitive ratio bound of $\frac{1}{\alpha}$ as $T \longrightarrow \infty$. The parameter $\alpha = \max_{m \in [M]} \frac{\bar{x}_m}{\rho_m}$ measures how stringent the initially assigned per-round budget is with respect to the agent's own maximum allocation constraint. Naturally, the larger α (i.e., the initial budget is relatively more limited), a lower competitive ratio bound. Moreover, the asymptotic competitive ratio bound in Theorem 3.1 matches the optimal bound for online allocation of a fixed budget [11].

We also note that, with the added uncertainties due to budget replenishment, the optimal (offline) resource price μ_t^* can also be time-varying, while the optimal resource price μ^* is fixed when without budget replenishment [11]. Consequently, even if we aggressively update the resource price μ_t by directly incorporating replenished budgets at each round, there is still no hope to learn the optimal dynamic resource price μ_t^* with a sublinear regret (or an asymptotic competitive ratio of 1); instead, we can incur additional utility losses due to aggressive but potentially incorrect tracking of μ_t^* in an adversarial setting. Therefore, OACP utilizes the design of conservative pricing while using opportunistic allocation for actual decisions. It adds to the literature by generalizing the state-of-the-art (asymptotically) competitive online algorithm for the setting of a fixed budget [11] to replenishable budgets.

In our setting with replenishable budgets, the adversary naturally has more power than the setting of a fixed budget, as it can adversrially present budget replenishments to the agent. Thus, achieving the same optimal asymptotic competitive ratio as that of state-of-the-art DMD for fixed budget allocation [11] demonstrates the merit of OACP in terms of addressing additional uncertainties of replenished budget.

Importantly, our asymptotic competitive ratio $\frac{1}{\alpha}$ is optimal in the adversarial budget replenishment setting. Specifically, in the adversarial case, it is possible that there is zero budget replenishment, or the budget replenishment only arises in the last decision round and the utility function for this round is chosen as zero by the adversary. As a consequence, the replenished budget cannot be utilized to improve the utility, and our setting essentially reduces to the no budget replenishment setting in the worst case. This means that without further assumptions on the budget replenishment, one cannot find a higher competitive ratio than the optimal bound $\frac{1}{\alpha}$ for online allocation with a fixed budget [11].

3.3 Extension to OACP+ with Minimum Budget Replenishment Assumption

In the unrestricted adversarial budget replenishment case, there can be zero budget replenishment and hence, one cannot expect a higher asymptotic competitive ratio than that of the optimal bound for fixed budget allocation. Next, to avoid the trivial case of no budget replenishment and improve

[1] Throughout the paper, the asymptotic competitive ratio is naturally no greater than 1, i.e., $CR^{\text{OACP}} = \min\{1, \frac{1}{\alpha}\}$.

Algorithm 2 Opportunistic Allocation with Conservative Pricing + (OACP+)

```
Require: Unit frame length $T^*$ and initial budget $B_1 = \rho T$ for $\rho > 0$
 1: for frame $i = 1$ to $K$ do
 2:     Initialize $\mu_{T_{i-1}+1}$, set learning rate $\eta_i > 0$, assign the budget $B^{(i)}_{T_{i-1}+1} = B^{(i)}$ as Eqn. (5) and
        $\hat{\rho}_i = B^{(i)}/(T_i - T_{i-1})$, where $T_i = (2^i - 1)T^*$.
 3:     for $t = T_{i-1} + 1$ to $T_i$ do
 4:         Receive utility function $f_t(x)$.
 5:         Pre-select action $x_t$ based on $\mu_t$: $\hat{x}_t = \arg\max_{x \in \mathcal{X}} \{f_t(x) - \mu_t^\top x\}$
 6:         if $\hat{x}_t \leq B_t^{(i)}$ then
 7:             $x_t = \hat{x}_t$ and $g_t = -\hat{x}_t + \hat{\rho}_i$
 8:         else
 9:             $x_t = 0$ and $g_t = 0$
10:         end if
11:         Update budget $B_{t+1}^{(i)} = B_t^{(i)} - x_t$ and the actual remaining budget $B_{t+1} = B_t + E_t - x_t$
12:         Update dual $\mu_{t+1} = \arg\min_{\mu \geq 0} g_t^\top \mu + \frac{1}{\eta_i} V_h(\mu, \mu_t)$.
13:     end for
14: end for
```

the asymptotic competitive ratio, we make a mild assumption on the minimum budget replenishment every T^* rounds (referred to as a unit frame) and propose a new algorithm called OACP+.

3.3.1 The Design of OACP+. As discussed in the key insight of Algorithm 1, aggressively setting $g_t = -\hat{x}_t + \rho + E_t$ for resource pricing cannot improve the competitive ratio since E_t is arbitrary and $\rho + E_t$ is not a reliable reference per-round budget in the adversarial case. On the other hand, a higher fixed budget means that the online allocator is less starved and hence can increase the competitive ratio [9, 11]. Thus, this provides us with an inspiration to improve the competitive ratio of OACP: *Batching the budget replenishment and allocating it later as if we had a higher fixed budget.*

Concretely, we design a new two-level online allocation algorithm, called OACP+, which divides an entire episode of T rounds into K frames and batches the budget replenishment in frame i for resource allocation in frame $i + 1$. Then, within each frame, OACP+ views the effective budget replenishment (subject to frame-level budget allocation to be specified in Eqn. (5)) in the previous frame as if it were a fixed resource and allocates it online.

OACP+ is described in Algorithm 2, where we introduce a unit frame of length $T^* \geq 1$ rounds during which a minimum amount of budget is replenished (see Definition 3). Note that OACP+ only needs the information of T^*, but does not know the minimum budget replenishment within T^* rounds. Within each frame $i \in [K]$ starting from round $T_{i-1} + 1$ to round T_i, we initialize the dual variable, assign the budget $B^{(i)}$ as the initial budget for frame i, and set the reference per-round budget $\hat{\rho}_i = B^{(i)}/(T_i - T_{i-1})$. Then, by considering that all the budget replenishment in frame i is deferred for allocation in frame $i + 1$ (Line 11 of Algorithm 2), we apply OACP with a fixed assigned frame-level budget $B^{(i)}$ to choose actions for all rounds in frame i. The dual variable is updated based on the reference per-round budget $\hat{\rho}_i$ and learning rate η_i for frame i. Note that in Line 6, we make sure the allocation is not larger than the remaining frame budget $B_t^{(i)}$ which is a part of the fixed assigned frame-level budget $B^{(i)}$. This means that the new budget replenishment in frame i is not incorporated in the resource pricing or used for allocation in frame i. The remaining frame-level budget $B_t^{(i)}$ and the actual remaining budget B_t are updated simultaneously in Line 11.

By batching the budget replenishment in frame i and deferring it for allocation in frame $i+1$, OACP+ can allocate more resources as if it had a higher fixed budget in frame $i+1$.

Nonetheless, to improve the competitive ratio, there are two key challenges in the design of OACP+— frame construction and frame-level budget assignment — which we address as follows.

Frame construction. To defer the budget replenishment in one frame to the next frame and allocate it as fixed budget, it is crucial to appropriately decide the length of each frame, i.e., frame construction. An intuitive way of frame construction is to divide the entire episode of T rounds uniformly into $K = \lceil T/T^* \rceil$ frames, each with $T^* \geq 1$ rounds (which is the length of a unit frame). By doing so, OACP+ incurs an additional term of $O(\sqrt{T^*})$ in the reward bound of each frame by Theorem 3.1 and hence a total additional term of $O\left(\sqrt{T^*}\lceil T/T^* \rceil\right)$, which grows linearly with T. Thus, to avoid the additional linear term $O\left(\sqrt{T^*}\lceil T/T^* \rceil\right)$, OACP+ utilizes a doubling frame construction as follows.

Specifically, the entire episode of T rounds is divided into $K = \lceil \log_2(T/T^*) \rceil$ frames, where $T^* \geq 1$ is the length of a unit frame. The i-th frame starts from round $T_{i-1}+1$ and ends at round T_i, where $T_i = (2^i - 1)T^*$.[2] In other words, assuming the first frame has a length of T^* rounds, the length of frame $i = 2, \cdots, K$ is $2^{i-1}T^*$, doubling the length of its previous frame $i-1$. For each frame i, the additional term incurred by OACP+ is $O(\sqrt{2^{i-1}T^*})$, the sum of which is still sublinear with respect to T, keeping the asymptotic competitive ratio independent of the choice of the initial dual in each frame.

Frame-level budget assignment. It remains to set the frame-level budget $B^{(i)}$ for each frame i given uncertain future budget replenishment. The initial fixed budget $B_1 = T\rho$ is proportionally divided into K frames: the frame-level budget $B^{(i)}$ for each frame i includes a fixed budget $2^{i-1}T^*\rho$, where $2^{i-1}T^*$ is the length of frame i. Additionally, the assigned frame-level budget $B^{(i)}$ also includes an additive budget Ω_i which comes from the budget replenishment and unused budgets assigned in previous frames. Without a maximum budget cap (i.e. $B_{\max} = \infty$), we can directly set Ω_i as the actual budget accumulation $B_{T_{i-1}+1} - (T - T_{i-1})\rho$, where $B_{T_{i-1}+1}$ is the actual remaining budget at the beginning of frame i and $(T - T_{i-1})\rho = (T - (2^{i-1} - 1)T^*)\rho$ is the sum of the fixed budget assignment reserved for the remaining frames (including frame i). Thus, by combining the fixed budget and replenished budget (including unused assignments) from previous frames, the assigned total budget for frame i is $B^{(i)} = B_{T_{i-1}+1} - (T - (2^i - 1)T^*)\rho$.

However, if the maximum budget cap $B_{\max}$ exists, it can restrict the actual budget replenishment. Thus, if we assign all the actually accumulated budget for frame i, it can happen that little additional budgets (other than the fixed budget $2^iT^*\rho$) can be used for frame $i+1$. To further explain this point, consider an online allocation problem with a linear utility function $f_t(x) =< c_t, x >$ (i.e., the inner product of c_t and x). Suppose that the remaining budget B_{T_1+1} at the beginning of the second frame (which has a length $2T^*$ rounds) is as large as $B_{\max}$. This is possible if there is a large budget replenishment during the first frame. As a result, new budget replenishments cannot be accumulated due to the budget cap $B_{\max}$ unless some budgets have been consumed. Assume that the budget replenishment $\hat{E}_t$ and context parameter c_t for the second frame are as follows. In the first T^*+1 rounds of the second frame, the budget replenishment is $\hat{E}_t > 0$ and the context is $c_t = 0$; in the following T^*-1 rounds of the second frame, the budget replenishment for each round is $\hat{E}_t = 0$ and the context parameter c_t is sufficiently large. In this example, OACP+ will not consume any resource during the first T^*+1 rounds, and instead consume all of the assigned budget $B^{(2)}$ during the remaining T^*-1 rounds. As a result, no budget replenishment can be accumulated in this frame

[2] The last frame (i.e., K-th frame) starts from round $(2^{K-1} - 1)T^* + 1$ and ends at the last round T. For the convenience of presentation, we assume $T = (2^K - 1)T^*$ to be consistent with the previous frame's ending round $T_i = (2^i - 1)T^*$.

due to the maximum budget cap. If we still assign the frame budget as $B^{(2)} = B_{T^*+1} - (T - 3T^*)\rho$ as if there were no budget cap, the remaining budget at the beginning of the third frame will be $B_{T_2+1} = (T - 3T^*)\rho$ and the assigned total budget for the third frame will be $4T^*\rho$, resulting in zero additive budget for the third frame ($\Omega_3 = 0$) other than the fixed budget assignment $4T^*\rho$.

To ensure a positive additive budget for each future frame, we need to allocate the budget replenishment Ω_i for frame i more conservatively: the additive budget $\Omega_i, i \in [2, K-1]$ is set as the minimum of the actual budget accumulation $B_{T_{i-1}+1} - (T - (2^{i-1} - 1)T^*)\rho$ and a threshold Γ_i, i.e., $\Omega_i = \min\{B_{T_{i-1}+1} - (T - (2^{i-1} - 1)T^*)\rho, \Gamma_i\}$.

It remains to design a proper threshold Γ_i for frame-level budget assignment. Naturally, if the budget cap $B_{\max}$ becomes larger, the threshold Γ_i should be set higher; also, Γ_i should increase with the length of the frame. We set the threshold as $\Gamma_i = 2^{i-2}T^*\rho_{\max} \odot \beta$ where the operator "$\odot$" is the element-wise product, $\rho_{\max} = \frac{B_{\max}}{T}$ and $\beta \in R_+^M$ is a hyper-parameter indicating the level of conservativeness to balance between the aggressive budget assignment for the next frame and conservative budget reservation for subsequent future frames. Therefore, the assigned total frame-level budget for frame i is the sum of the fixed budget assignment $2^{i-1}T^*\rho$ (where $\rho = \frac{B_1}{T}$) and an additive budget Ω_i, i.e.

$$B^{(i)} = 2^{i-1}T^*\rho + \Omega_i, \tag{5}$$

where $\Omega_1 = 0$, $\Omega_i = \min\left\{B_{T_{i-1}+1} - (T - (2^{i-1} - 1)T^*)\rho, \Gamma_i\right\}$ with $\Gamma_i = 2^{i-2}T^*\rho_{\max} \odot \beta$ for $i \in [2, K-1]$, and $\Omega_K = B_{T_{K-1}+1} - 2^{K-1}T^*\rho$. When $\rho_{\max} = \frac{B_{\max}}{T}$ is sufficiently large such that the threshold Γ_i is not activated, the assigned budget becomes $B^{(i)} = B_{T_{i-1}+1} - (T - (2^i - 1)T^*)\rho$, which reduces to the budget assignment without a maximum budget cap and shows the flexibility of our design of frame-level budget assignment.

3.3.2 *Performance Analysis.*

In this section, we give the asymptotic competitive ratio of OACP+ to highlight the benefits of budget replenishment. To avoid the adversarial case which can reduce to the no budget replenishment setting, we first define the minimum replenishment $E_{\min} \geq 0$ for a unit frame with length T^* and then provide the asymptotic competitive ratio relying on $E_{\min}$. The assumption of minimum budget replenishment in each unit frame is reasonably mild in practice, especially for large T^*. For example, it is reasonable to assume that a minimum amount of solar renewables are replenished each day [7, 11, 51]. Note that $E_{\min}$ is decided by the environment and OACP+ does not need the knowledge of $E_{\min}$.

Definition 3 (Minimum budget replenishment). Given a unit frame of $T^* \geq 1$ rounds, the minimum potential budget replenishment for type-m resource within each unit frame is $E_{\min,m} \geq 0$, i.e., $E_{\min,m} = \inf_j \left\{\sum_{t=(j-1)T^*+1}^{j\cdot T^*} \hat{E}_{t,m}\right\}$, where $\hat{E}_{t,m}$ is the budget that would be replenished at round t if $B_{\max,m} \to \infty$, $j = 0, \cdots, \lceil T/T^* \rceil - 1$ is the index of a unit frame and $E_{\min} = \left[E_{\min,1}, \cdots, E_{\min,M}\right]$.

Theorem 3.2. *If the learning rate for frame i is chosen as* $\eta_i = \frac{1}{\bar{\rho} + \frac{\bar{\beta}}{2}\bar{\rho}_{\max} + \|\bar{x}\|_\infty}\sqrt{2\sigma V_h(\mu, \mu_{T_{i-1}+1})/(2^{i-1}T^*)}$ *with* $\bar{\rho}_{\max} = \max_m \rho_{\max,m}$ *where* $\rho_{\max,m} = \frac{B_{\max,m}}{T}$ *and* $\bar{\beta} = \max_m \beta_m$, *OACP+ achieves an asymptotic competitive ratio against OPT as*

$$CR^{\text{OACP+}} = \min_{m \in [M]} \frac{\rho_m + \Delta\rho_m}{\bar{x}_m}, \tag{6}$$

where $\bar{x}_m$ *is the maximum per-round allocation of type-m resource and* $\Delta\rho_m \geq 0$ *is the improvement due to budget replenishment. Specifically, if* $B_{\max,m} \geq (T + T^*)\rho_m$ *holds for a resource m, we have* $\Delta\rho_m = \min\left\{\frac{E_{\min,m}}{2T^*}, \frac{2B_{\max,m}}{3(T+T^*)} - \frac{\rho_m}{3}\right\}$ *with the optimal choice of* $\beta_m = \frac{4T}{3(T+T^*)} - \frac{2\rho_m}{3\rho_{\max,m}}$*; and if* $B_{\max,m} <$

$(T+T^*)\rho_m$ holds for a resource m, we have $\Delta\rho_m = \min\left\{\frac{E_{\min,m}}{2T^*}, \frac{B_{\max,m}}{6T^*} - \frac{(T-T^*)\rho_m}{6T^*}\right\}$ with the optimal choice of $\beta_m = \frac{T}{3T^*} - \frac{T-T^*}{3T^*}\frac{\rho_m}{\rho_{\max,m}}$. Moreover, without the minimum budget replenishment (i.e., $E_{\min,m} = 0$), we have $\Delta\rho_m = 0$ and the asymptotic competitive ratio $CR^{\mathsf{OACP+}}$ reduces to the one in Theorem 3.1.

The proof of Theorem 3.2 is deferred to Section B. The key challenge is to lower bound the assigned frame-level budget $B^{(i)}$ in Eqn. (5) for frame i and get an effective per-round budget $\hat{\rho} = \rho + \Delta\rho$. To do so, we construct an effective additive budget $\hat{\Omega}_i(\beta)$ for frame i given any $\beta > 0$ in (21) and prove that $\hat{\Omega}_i(\beta)$ is the infimum of the additive budget $\Omega_i(\beta)$ by OACP+ for any $\beta > 0$. Then, by selecting the worst-case per-round effective reference budget $\rho + \hat{\Omega}_i/(2^{i-1}T^*)$ for each frame i and optimizing it by choosing β, we obtain the per-round budget $\hat{\rho} = \rho + \Delta\rho$. At last, by summing up the utility bounds of all the frames, the difference between the optimal utility and OACP+ is bounded as $OPT(y) - \hat{\alpha}F_T^{\mathsf{OACP+}}(y) \le \hat{C}_1 + \hat{C}_2\sqrt{T}$, where $\hat{\alpha} = \max_{m\in[M]} \frac{\bar{x}_m}{\rho_m + \Delta\rho_m}$, and $\hat{C}_1$ and $\hat{C}_2$ are two finite constants in Appendix B. This is then translated to the asymptotic competitive ratio in Theorem 3.2.

Different from the competitive ratio of OACP which relies on the fixed per-round budget ρ, the competitive ratio of OACP+ utilizes the effective per-round budget $\rho + \Delta\rho$, which includes the fixed part ρ and the additional part $\Delta\rho$ due to replenishment (subject to the maximum budget cap $B_{\max}$). Importantly, $\Delta\rho$ is positive if the minimum replenishment over a unit frame $E_{\min} > 0$, resulting in a higher asymptotic competitive ratio than OACP. When $E_{\min} = 0$, there is no guarantee of minimum budget replenishment for each unit frame. Hence, the asymptotic competitive ratio of OACP+ reduces to the one achieved by OACP in the worst case since we cannot rule out the case in which there is no budget replenishment at all. Thus, the improvement of the competitive ratio by OACP+ does not conflict with the optimality of the competitive ratio achieved by OACP for general cases (which includes the case of no budget replenishment).

The insights of the asymptotic competitive ratio of OACP+ are further explained as follows. The improvement of the competitive ratio compared with OACP depends on $\Delta\rho_m$, which is lower bounded by the minimum of two terms. The first term $\frac{E_{\min,m}}{2T^*}$ indicates the effect of the minimum amount of budget replenishment within a unit frame. Naturally, a larger minimum budget replenishment can make the problem less resource-constrained and lead to a higher competitive ratio. The second term in the minimum operation shows the effect of the maximum budget cap $B_{\max,m}$ on constraining the actual budget replenishment following (1c). The second term has a different expression for resource m with $B_{\max,m} < (T+T^*)\rho_m$ because a small $B_{\max,m}$ can result in less space for replenishment. No matter whether $B_{\max,m} \ge (T+T^*)\rho_m$ holds, a higher budget cap $B_{\max}$ allows more budgets to be replenished, thus leading to a higher competitive ratio. If the budget cap $B_{\max}$ is large enough, it does not constrain the budget replenishment any more and the competitive ratio improvement only depends on the minimum budget replenishment $E_{\min}$. In addition, the best choices of threshold hyper-parameter β_m increases with $\rho_{\max,m} = B_{\max,m}/T$. This is consistent with the intuition that with a larger budget cap $B_{\max,m}$, the threshold of the additive budget in Eqn. (5) can be set larger to assign the frame-level budget more aggressively. These observations all confirm the intuition that a larger budget cap can utilize the budget replenishment more effectively, increasing the asymptotic competitive ratio.

4 LA-OACP: LEARNING-AUGMENTED ONLINE ALLOCATION

While OACP and OACP+ have provable worst-case performance guarantees (in terms of asymptotic competitive ratio), they may not perform well on average due to their conservativeness in resource pricing μ_t in order to address the worst-case uncertainties in budget replenishments. In this section, we go beyond the worst-case and propose a novel learning-augmented approach, LA-OACP, that integrates an ML-based online optimizer with OACP (or OACP+) to improve the average performance (Theorem 4.2) while still being able to guarantee the worst-case performance (Theorem 4.1).

4.1 Average Utility Maximization with Worst-Case Utility Constraint

We first present our optimization objective of designing a learning-augmented online algorithm π as follows — maximizing the average utility subject to a worst-case utility constraint. Since the competitive algorithms (i.e., OACP or OACP+) have been proved to ensure the asymptotic competitive ratios, we guarantee the worst-case utility of the learning-augmented online algorithm π by comparing it with the utility of a competitive algorithm. Thus, the objective of our learning-augmented online algorithm is

$$\max_{\pi} \mathbb{E}_y \left[F_T^{\pi}(y)\right] \tag{7a}$$

$$s.t., \quad F_T^{\pi}(y) \geq \lambda F_T^{\pi^\dagger}(y) - R, \quad \forall y \in \mathcal{Y}, \tag{7b}$$

where $F_T^{\pi}(y) = \sum_{t=1}^{T} f(x_t, c_t)$ is the total utility of an online algorithm π, $\lambda \in [0,1]$ represents multiplicative competitiveness of the online algorithm π with respect to the algorithm $\pi^\dagger$ (i.e., OACP or OACP+ in our case) and $R \geq 0$ indicates the additive slackness in the utility constraint. Note that considering a sequence-wise distribution of $y \sim \mathbb{P}_y$ differs from the standard stochastic setting where each online input y_t for $t \in [t]$ is assumed to follow an independent and identically distributed (i.i.d.) distribution (e.g., i.i.d. utility function f_t in [11], or i.i.d. potential replenishment $\hat{E}_t$ in [7]), because y_t for $t \in [t]$ within an sequence can still be arbitrary in our problem (7).

The parameters $\lambda \in [0,1]$ and $R \geq 0$ can be viewed as worst-case robustness requirement with respect to OACP or OACP+ (denoted as $\pi^\dagger$ for the convenience of presentation). Concretely, when $\lambda \in [0,1]$ increases and/or $R \geq 0$ decreases, the online algorithm π is closer to $\pi^\dagger$ in terms of the worst-case utility, and vice versa. Moreover, as $\pi^\dagger$ itself has performance guarantees and is asymptotically competitive against the optimal oracle OPT (Theorem 3.1 and Theorem 3.2), the constraint in (7b) also immediately translates into provable asymptotic competitiveness of the online algorithm π with respect to OPT. That is, given the asymptotic competitive ratio $CR^{\pi^\dagger}$ achieved by $\pi^\dagger$, the constraint (7b) leads to $\lim_{T\to\infty} \sup_y \frac{1}{T}\left(OPT(y) - \frac{1}{\lambda CR^{\pi^\dagger}} F_T^{\pi}(y)\right) \leq \lim_{T\to\infty} \sup_y \frac{1}{T}\left(OPT(y) - \frac{1}{CR^{\pi^\dagger}} F_T^{\pi^\dagger}(y) + \frac{R}{\lambda CR^{\pi^\dagger}}\right) \leq 0$, guaranteeing an asymptotic competitive ratio of $\lambda \cdot CR^{\pi^\dagger}$ for π. In fact, considering a baseline algorithm for worst-case robustness is also a common practice in existing learning-augmented algorithms [19, 39, 52]. Thus, in the following, it suffices to consider (7) to achieve the best of both worlds: maximizing the average utility while bounding the worst-case utility (directly with respect to OACP or OACP+ and also indirectly with respect to OPT).

Unlike OACP or OACP+ that is particularly designed to address the worst-case robustness, an ML model can readily exploit statistical information of $y \in \mathcal{Y}$ based on history instances. Thus, one may want to use a pure ML-based online optimizer to maximize the average utility for solving (7). Nonetheless, ML-based optimizers typically do not have worst-case performance guarantees as hand-crafted algorithms (OACP or OACP+ in our case) due to, e.g., training-testing distributional shifts. In fact, even by assuming perfect ML-based optimizers, maximizing the average utility alone does not necessarily guarantee the worst-case robustness in (7b). The reason is that maximizing the average utility needs to prioritize many typical problem instances, while the worst-case robustness needs to address those rare but possible corner cases. In general, the trade-off between average utility and worst-case robustness is unavoidable and well-known for online optimization problems, thus spurring the emerging field of learning-augmented online algorithms that leverage both

ML predictions and hand-crafted algorithms (see, e.g., [5, 19, 52, 57] for studies in other online problems).

4.2 Algorithm Design

We now present the design of LA-OACP, a novel learning-augmented algorithm for online allocation with replenishable budgets under an additional mild assumption of Lipschitz utility functions.

Assumption 3 (Lipschitz utility). For any $t \in [T]$, the utility function $f_t(x)$ is L-Lipschitz continuous with respect to x, i.e. $\forall x, x' \in \mathcal{X}$, we have $|f_t(x) - f_t(x')| \leq L\|x - x'\|$, where $L > 0$ and $\|\cdot\|$ is a norm operator.

The Lipschitz assumption implies a bounded utility change given a bounded action change, which is reasonable for real applications and commonly assumed in online problems. Remember that to guarantee a competitive ratio, OACP in Algorithm 1 and OACP+ in Algorithm 2 conservatively set their resource prices μ_t in two different conservative manners. Thus, the key goal of LA-OACP is to overcome the conservativeness of competitive algorithms like OACP and OACP+ by exploiting the distribution of $y \in \mathcal{Y}$ to improve the average utility while bounding the worst-case utility loss with respect to OACP/OACP+.[3] Towards this end, LA-OACP utilizes an ML policy/predictor (denoted as $\tilde{\pi}$) as well as a competitive algorithm (denoted as $\pi^\dagger$) that output their decisions as advice, and then judiciously chooses the actual online decisions.

Naturally, always following the decisions of competitive algorithm satisfies the worst-case utility constraint in (7b), but fails to utilize ML for average utility improvement. On the other hand, blindly following the ML policy can potentially improve the average performance but the worst-case utility constraint is not guaranteed.

Thus, a key challenge of learning-augmented online algorithms is how to utilize the decisions of the ML policy and a worst-case robust algorithm (i.e., OACP and OACP+ in our case) as online advice [19, 52]. To address this challenge, given $\tilde{x}_t$ and $x_t^\dagger$ that represent the allocation decisions by the ML policy and the competitive algorithm, respectively, LA-OACP chooses the actual decision x_t using a novel reservation utility which we introduce as follows. In the following, to be consistent with the literature [19], we also refer to the ML policy's decision $\tilde{x}_t$ as ML predictions.

Constrained decision set. To ensure that an online algorithm π satisfies the worst-case utility constraint (7b) for any sequence $y \in \mathcal{Y}$, it might seem sufficient to guarantee $\sum_{\tau=1}^{t} f_t(x_t) \geq \lambda \sum_{\tau=1}^{t} f_t(x_t^\dagger) - R$ for each round $t \in [T]$. Nonetheless, even though the constraint $\sum_{\tau=1}^{t} f_t(x_t) \geq \lambda \sum_{\tau=1}^{t} f_t(x_t^\dagger) - R$ is satisfied for round t, it may not be guaranteed at round $t+1$, thus potentially violating the worst-case utility constraint at the end of the sequence. Let us now consider an illustrative example to explain this point. Suppose that the algorithm π satisfies $\sum_{\tau=1}^{t} f_t(x_t) \geq \lambda \sum_{\tau=1}^{t} f_t(x_t^\dagger) - R$ but allocates more resources than $\pi^\dagger$ up to round t. Then, in future rounds, it is possible that there is very little budget replenishment and the algorithm $\pi^\dagger$ can still allocate resources to gain a higher utility, whereas the algorithm π does not have enough resources to match the utility of $\pi^\dagger$. In other words, if π uses more resources than $\pi^\dagger$ up to round t, the satisfaction of utility constraint by π in terms of $\sum_{\tau=1}^{t} f_t(x_t) \geq \lambda \sum_{\tau=1}^{t} f_t(x_t^\dagger) - R$ is just *temporary* and can still be violated in the future.

To address such uncertainties in the future, we introduce a novel reservation utility $\Delta(x_t) = \lambda L \sum_{m=1}^{M} \left[(B_t^\dagger + E_t^\dagger - x_t^\dagger)_m - (B_t + E_t - x_t)_m\right]^+$ into the utility constraint for each round t, where $(B_t + E_t - x_t)_m$ means the remaining budget for the type-m resource at the end of round t. The

[3] For notational simplicity, we use LA-OACP to represent our learning-augmented algorithm, noting that the competitive algorithm used by LA-OACP can also be OACP+.

Algorithm 3 Learning-Augmented Online Allocation with Replenishable Budgets (LA-OACP)

```
Require: ML policy $\tilde{\pi}$ and the competitive algorithm $\pi^\dagger$ (OACP or OACP+)
for $t = 1$ to $T$ do
    Receive reward function $f_t$, and potential budget replenishment $\hat{E}_t$.
    Get replenished budgets $E_t = \min\{\hat{E}_t, B_{\max} - B_t\}$ for LA-OACP, and $E_t^\dagger = \min\{\hat{E}_t, B_{\max} - B_t^\dagger\}$ for $\pi^\dagger$
    Get ML prediction $\tilde{x}_t$
    Get the action $x_t^\dagger$ of $\pi^\dagger$ based on its own history (by Algorithm 1 or Algorithm 2)
    Choose $x_t$ by solving
        $x_t = \arg\min_{x \in \mathcal{X}} \|x - \tilde{x}_t\|$    (9a)
        $s.t., \sum_{i=1}^{t} f_t(x_i) \geq \lambda \sum_{i=1}^{t} f_t(x_i^\dagger) + \Delta(x_t) - R,$ and $x_t \leq B_t + E_t,$    (9b)
    where $\Delta(x_t) = \lambda L \sum_{m=1}^{M} \left[ (B_t^\dagger + E_t^\dagger - x_t^\dagger)_m - (B_t + E_t - x_t)_m \right]^+$
    Update budgets $B_{t+1} = B_t - x_t + E_t$ for LA-OACP, and $B_{t+1}^\dagger = B_t^\dagger - x_t^\dagger + E_t^\dagger$ for $\pi^\dagger$
end for
```

interpretation of $\Delta(x_t)$ is to bound the maximum potential utility advantage (scaled by $\lambda \in [0,1]$) obtained by $\pi^\dagger$ in future rounds, if $\pi^\dagger$ has more remaining budgets compared to π at the end of round t; on the other hand, if the algorithm π has even more resources available than $\pi^\dagger$, there is no need to add the reservation since π can always roll back to the decision of $\pi^\dagger$ in the future without worrying about budget shortages. Here, we simply use $\Delta(x_t)$ for the convenience of presentation while suppressing its dependency on other terms such as $x_t^\dagger$. Thus, by adding $\Delta(x_t)$, we now have a new constraint on the decision x_t as follows:

$$\sum_{i=1}^{t} f_t(x_i) \geq \lambda \sum_{i=1}^{t} f_t(x_i^\dagger) + \Delta(x_t) - R, \tag{8}$$

which, if satisfied at round t, guarantees the existence of at least one feasible decision that can still satisfy the constraint. In other words, if the decisions x_t are chosen out of the constrained set (8) for round $t \in [T]$, worst-case utility constraint (7b) can be satisfied at the end of any sequence $y \in \mathcal{Y}$. To our knowledge, the design of $\Delta(x_t)$ for constructing a constrained decision set (8) is novel for online allocation with replenishable budgets and also differs from many prior learning-augmented algorithms (e.g., [5] uses a pre-determined threshold for dynamically switching between ML prediction $\tilde{x}_t$ and the worst-case robust action $x_t^\dagger$).

Algorithm. Next, we describe the online optimization process of LA-OACP in Algorithm 3. In LA-OACP, the competitive algorithm (i.e., $\pi^\dagger$) runs independently for the purpose of bounding the worst-case utility constraint (7b), and the ML predictor $\tilde{\pi}$ takes the actual online information $y_{1:t}$ (including the actual remaining budget B_t and replenishment E_t) as its input and generates its prediction $\tilde{x}_t$ as advice to LA-OACP. Then, $\tilde{x}_t$ is projected into a constrained decision set (8) to find the actual decision x_t that guarantees the worst-case utility constraint. The purpose of the projection in LA-OACP is to ensure that x_t is both close to the ML prediction $\tilde{x}_t$ to exploit its potential for improving the average utility, while still staying inside the constrained decision set (8) for worst-case utility constraint (7b).

ML training. Up to this point, we have assumed that the ML predictor/policy $\tilde{\pi}$ has been provided to LA-OACP for online optimization. Next, we discuss how to train $\tilde{\pi}$ used in Algorithm 3.

In the context of online optimization, the ML-based predictor/policy is typically trained offline and then applied online for inference [2, 23, 35, 37]. Here, we adopt this standard practice for LA-OACP. Specifically, we collect a training dataset $\mathcal{S}$ of episodic information $y \in \mathcal{Y}$ based on history data and/or data augmentation techniques, and build an ML model (e.g., a recurrent neural network for online sequential decision making, with each base network parameterized by the same weights [2, 35]).

We train the ML model $\tilde{\pi}$ by optimizing the expected utility obtained by Algorithm 3. Denote LA-OACP ($\tilde{\pi}$) as the algorithm LA-OACP with the ML model $\tilde{\pi}$. The training objective is expressed as

$$\max_{\tilde{\pi}} \frac{1}{|\mathcal{S}|} \sum_{y \in \mathcal{S}} F_T^{\texttt{LA-OACP}(\tilde{\pi})}(y), \tag{10}$$

where $F_T^{\texttt{LA-OACP}(\tilde{\pi})}(y)$ is the total utility of LA-OACP($\tilde{\pi}$) for the sequence y.

To train the ML model, we apply the state-of-the-art backpropagation, while noting that differentiation of the projection operator (which itself is a constrained optimization problem) with respect to the ML prediction $\tilde{x}_t$ is needed and can be performed based on implicit differentiation techniques [3, 4, 34].

4.3 Performance Analysis

We now present the performance analysis of LA-OACP in terms of its worst-case utility as well as its average performance. As formally stated below, our results highlight that LA-OACP guarantees the worst-case utility constraint for any sequence $y \in \mathcal{Y}$ and meanwhile is able to exploit the benefits of ML predictions to improve the average utility.

4.3.1 Worst-Case Utility. We first present the worst-case utility of LA-OACP.

Theorem 4.1. *For any $\lambda \in [0, 1]$ and $R \geq 0$, given any ML predictor $\tilde{\pi}$ and by the design $\Delta(x_t) = \lambda L \sum_{m=1}^{M} \left[(B_t^\dagger + E_t^\dagger - x_t^\dagger)_m - (B_t + E_t - x_t)_m \right]^+$, LA-OACP in Algorithm 3 always guarantees the worst-case utility constraint (7b) for any sequence $y \in \mathcal{Y}$.*

The proof of Theorem 4.1 is available in Appendix C and shows that, based on the design of $\Delta(x_t)$, if (8) is satisfied for round t, then there must always exist a feasible solution satisfying (8) for round $t + 1$.

Theorem 4.1 guarantees that the worst-case utility constraint (7b) is always satisfied for any sequence $y \in \mathcal{Y}$ regardless of how bad the ML predictions are. Thus, even when the training-testing distributions differ and/or the ML predictions are adversarially modified, LA-OACP can still offer worst-case utility guarantees with respect to the competitive algorithm OACP or OACP+.

4.3.2 Average Utility. Besides the robustness guarantee, the performance of a learning-augmented algorithm is often analyzed by considering the *worst*-case competitive ratio (a.k.a., consistency) under the assumption that ML predictions are perfect and offline optimal for any sequence $y \in \mathcal{Y}$ [5, 57]. The consistency metric measures how closely a learning-augmented algorithm can follow the perfect ML predictions in the worst case. However, an ML model in practice is typically not perfect and instead is trained to maximize the average performance in practice. Thus, to measure the capability of LA-OACP for following ML predictions, we directly bound the average utility of LA-OACP and compare it with the average utility of the optimal unconstrained ML model $\tilde{\pi}^* = \arg\max_\pi \mathbb{E}_y \left[F_T^\pi(y) \right]$ that provides the best average performance. As such, given an optimally trained ML model, we measure the *average*-case consistency of LA-OACP with respect to the optimal unconstrained ML model $\tilde{\pi}^*$ in terms of the average utility difference between LA-OACP and $\tilde{\pi}^*$. Our consideration of an optimal ML model essentially parallels the assumption of "perfect ML prediction" for worst-case consistency analysis of learning-augmented algorithms [52, 57], while noting that our optimality is in the average sense subject to our designed constrained decision set (8).

More concretely, we consider an optimal ML predictor that optimizes the average utility of LA-OACP, i.e.

$$\tilde{\pi}^{\circ} = \arg\max_{\tilde{\pi}} \mathbb{E}_y \left[F_T^{\texttt{LA-OACP}(\tilde{\pi})}(y) \right], \tag{11}$$

where LA-OACP$(\tilde{\pi})$ outputs the actions x_t satisfying (8) given the ML prediction $\tilde{x}_t$, and show the average utility bound of LA-OACP in the next theorem.

Theorem 4.2. *For any $\lambda \in [0, 1]$ and $R \geq 0$, the average utility of* LA-OACP *with the optimal ML model $\tilde{\pi}^{\circ}$ is bounded by*

$$\mathbb{E}_y \left[F_T^{\texttt{LA-OACP}(\tilde{\pi}^{\circ})}(y) \right] \geq \max \left\{ \mathbb{E}_y \left[F_T^{\tilde{\pi}^*}(y) \right] - L(1 - \gamma_{\lambda,R}) \mathbb{E}_y \left[\sum_{t=1}^{T} \| x_t^{\dagger} - \tilde{x}_t^* \| \right], \mathbb{E}_y \left[F_T^{\pi^{\dagger}}(y) \right] \right\} \tag{12}$$

where $\gamma_{\lambda,R} = \min \left\{ 1, \frac{R}{2\lambda L \theta} \right\}$, $\theta = \max_y \sum_{t=1}^{T} \| \tilde{x}_t^* - x_t^{\dagger} \|_1$ indicates the maximum cumulative decision difference between the action $x_t^{\dagger}$ of OACP *or* OACP+ *and the action $\tilde{x}_t^*$ of the optimal unconstrained ML predictor $\tilde{\pi}^*$, and L is the Lipschitz constant of the utility function (Assumption 3).*

The proof of Theorem 4.2 is available in Appendix D. The key idea is to translate the constraint (8) into a new distance constraint between x_t and $x_t^{\dagger}$. Thus, if x_t is sufficiently close to $x_t^{\dagger}$ for each round $t \in [T]$, we guarantee the worst-case utility constraint (7b). Meanwhile, by considering the optimal unconstrained ML predictor $\tilde{\pi}^* = \arg\max_{\pi} \mathbb{E}_y \left[F_T^{\pi}(y) \right]$, we find the closest distance between x_t and the ML prediction $\tilde{x}_t^*$ subject to the distance constraint between x_t and $x_t^{\dagger}$, and use this as a feasible online algorithm. The bound of such a closest distance requires an analysis of the remaining budget perturbation depending on the non-linear budget dynamics in (1c) due to the maximum budget cap. Next, by optimality of $\tilde{\pi}^{\circ}$ used by LA-OACP to explicitly maximize the average utility satisfying our constraint (8), we obtain the bound in Theorem 4.2.

Theorem 4.2 shows that the average utility of LA-OACP$(\tilde{\pi}^{\circ})$ with the optimal ML model $\tilde{\pi}^{\circ}$ is no worse than that of the competitive algorithm $\pi^{\dagger}$ (OACP or OACP+) which is the second term in the maximum operator. The reason is that the competitive algorithm $\pi^{\dagger}$ is one of the decision policies with actions in the constrained decision sets (8), whereas LA-OACP$(\tilde{\pi}^{\circ})$ is the optimal policy satisfying the decision constraints (8). This indicates that, while providing the worst-case performance guarantee, LA-OACP can still improve the average utility compared with the competitive algorithm (OACP or OACP+). The improvement relies on the first term in the maximum operator, which bounds the average utility difference between LA-OACP and the optimal-unconstrained ML model $\tilde{\pi}^*$.

The first term in the maximum operator in Theorem 4.2 provides the key insight into the tradeoff between the worst-case performance and average performance. Specifically, with a smaller $\lambda \in [0, 1]$ and/or greater $R \geq 0$, the worst-case utility constraint is less stringent and hence provides more flexibility for LA-OACP to exploit the benefits of ML predictions for higher average utility, and vice versa. In particular, when R is large enough or λ is small enough, the worst-case utility constraint in (7b) is so relaxed that it does not affect average utility maximization. In such cases, LA-OACP approaches the average utility of the optimal unconstrained ML predictor. When the decisions of the optimal-unconstrained ML predictor and the competitive algorithm become more distinct (i.e., increasing θ or $\mathbb{E}_y \left[\sum_{t=1}^{T} \| x_t^{\dagger} - \tilde{x}_t^* \| \right]$ in Theorem 4.2), it is natrually more difficult to follow the ML predictions while still staying close to the competitive algorithm for worst-case utility, unless we lessen the worst-case utility constraint by decreasing $\lambda \in [0, 1]$ and/or increasing $R \geq 0$.

Theorem 4.2 gives the bound of average utility by assuming an optimal ML model $\tilde{\pi}^\circ$ which parallels the assumption of "perfect ML prediction" for the worst-case consistency analysis in existing learning-augmented algorithms [52, 57]. However, if the ML model $\tilde{\pi}$ in LA-OACP is not optimally trained, we can define the ML prediction imperfectness as $\epsilon = \mathbb{E}_y\left[F_T^{\tilde{\pi}^\circ}(y) - F_T^{\tilde{\pi}}(y)\right]$, where $\tilde{\pi}^\circ = \arg\max_{\tilde{\pi}} \mathbb{E}_y\left[F_T^{\texttt{LA-OACP}(\tilde{\pi})}(y)\right]$ is the optimal ML model for LA-OACP. The imperfectness can come from a variety of sources, including finite model capacity and potential training-testing distributional shifts. Then, the average utility bound with respect to $\tilde{\pi}$ can be obtained by subtracting the ML imperfectness ϵ from the average utility bound in Theorem 4.2. Nevertheless, even when $\epsilon \longrightarrow \infty$, the average utility of LA-OACP is always bounded by $\lambda\mathbb{E}_y\left[F_T^{\pi^\dagger}(y)\right] - R$, where $\mathbb{E}_y\left[F_T^{\pi^\dagger}(y)\right]$ is the average utility of the competitive algorithm (OACP or OACP+) used by LA-OACP. This is a natural byproduct of Theorem 4.1, which guarantees the worst-case utility constraint of LA-OACP with respect to the competitive algorithm.

In general, achieving the optimal tradeoff between average utility and the worst-case utility is extremely challenging for learning-augmented algorithms (see, e.g., [19, 52] for discussions on smoothed online convex optimization). Nonetheless, although it remains an open problem to achieve the best tradeoff, our result in Theorem 4.2 provides the first characterization of such a tradeoff in the context of learning-augmented algorithms for online allocation with budget replenishment. In fact, even a competitive online algorithm with budget replenishment is lacking prior to our design of OACP and OACP+.

5 SIMULATION STUDY

In this section, we run a simulation study on sustainable AI inference powered by renewables. First, we present the experimental setup, followed by the comparative analysis of the results from our algorithms with existing baselines. We show that LA-OACP has improved performance in terms of average utility while still being able to offer good worst-case utility.

5.1 Setup

This section presents our problem setting, dataset, baseline algorithms, and ML model architecture.

Problem setting. Edge data centers are becoming a major platform for AI inference thanks to their proximity to end users. To achieve sustainable AI inference on the edge, it is important to exploit renewable generation to replenish on-site energy storage. This can significantly lower the carbon emissions caused by the surging demand for AI inference [43]. For a given AI inference service, multiple models are often available. For instance, there are eight different GPT-3 models [17], each with distinct model sizes, providing a flexible balance between accuracy and energy consumption. However, the renewable sources are known for their time-varying and unstable nature. Thus, we can use intermittent renewables to replenish the energy budgets, and schedule an appropriate AI model for inference in an online manner to maximize the utility given available energy budget constraints [51, 53].

Specifically, we focus on an edge data center with an on-site energy storage unit (e.g., batteries) for AI inference. The initial energy budget is $B_1 = 12kWh$. At each round t, the time-varying renewable energy E_t is replenished to the energy storage subject to the maximum capacity constraint $B_{\max} = 30kWh$. Each problem instance has 120 rounds. If served by the full AI model, the energy consumption for inference is c_t, which also measures the total demand. Nonetheless, the resource manager can decide an AI model at each round t, which consumes energy x_t. If a smaller AI model is chosen, then x_t is also smaller, but the inference accuracy is potentially lower. Here, we use a utility function to denote the reward by consuming x_t energy for serving the demand. Specifically, we model the utility of serving each unit of AI inference demand as $\log(1+\min\{1,\frac{x_t}{c_t}\})$, where the min operator means that over-using energy x_t beyond the maximum demand does not offer additional utility. Next, by using the total demand c_t to scale the demand, we have a utility function of $f_t(x_t)=c_t\log(1+\min\{1,\frac{x_t}{c_t}\})$ at time t. Note that choosing $x_t=0$ means that the inference demand is not processed by the edge (and routed to cloud data centers beyond our scope). The remaining budget in the energy storage is then updated according to (1c). The goal of the resource manager is to maximize $\sum_{t=1}^{T} f_t(x_t)$ subject to the energy budget constraint.

Dataset. In our experiment, the inference demand c_t comes from the GPU power usage of the BLOOM model (a large lanugage model) API running on 16 Nvidia A100 GPUs [43]. The budget replenishment E_t (harvested renewable energy) is constructed based on the renewable dataset from California Independent System Operator [47], which contains hourly solar renewables. The values are scaled down to our setting. We extend the BLOOM trace data by data augmentation to construct a training dataset consisting of 1600 problem instances, each with 120 hours. Then, the entire dataset is divided into training and testing sets with a 3:1 ratio.

Baseline algorithms. To compare our results, we consider the following baseline algorithms.

– *OPT*: OPT is the optimal oracle algorithm that solves (1) based on complete offline information. Thus, OPT has the highest utility for any problem instance.

– *Equal:* Equal uniformly allocates the initial budget, and greedily uses the replenished budget whenever applicable, i.e., $x_t=\min\{\bar{x},\rho+E_t\}$.

– *Greedy:* Greedy allocates as much budget as possible at each round, i.e., $x_t=\min\{\bar{x},B_t+E_t\}$.

– *DMD:* DMD (Dual Mirror Descent) updates the dual variable by mirror descent [11]. With replenishable budgets, DMD updates the dual variable based on subgradient $g_t=\rho+E_t-\hat{x}_t$.

– *ML:* ML uses a standalone ML predictor to yield online allocation decisions subject to per-round budget constraints. Such ML-based online optimizer empirically have superior *average* performance in a variety of online problems (when training-testing distributions are consistent) [2, 23, 35], but cannot guarantee worst-case utility bounds.

The hyperparameters for these algorithms, if applicable, are tuned based on our validation dataset to achieve the maximum utility. While it is not possible to compare our algorithms with all the existing baselines in the literature, our choice of baseline algorithms is representative in the sense that they cover the strongest OPT, naive Greedy, state-of-the-art competitive online algorithm DMD, as well as state-of-the-art ML-based online optimizers. Thus, we do not consider other competitive algorithms than state-of-the-art DMD, or other algorithms that focus on average performance but do not have as empirically good performance as ML. Importantly, our design of OACP or OACP+ is provably-competitive and LA-OACP can provably satisfy the worst-case utility constraint (7b) with respect to any available online algorithm by using it to replace OACP or OACP+ as $\pi^{\dagger}$ in Algorithm 3.

ML model architecture. We implement the ML model based on a neural network with 2 hidden layers each having a width of 10 with ReLu activation. To train the model, we use the Adam optimizer for 100 epochs with a batch size of 20 and a learning rate of 0.001. The same ML architecture is also used in LA-OACP.

5.2 Results

In this section, we present a comparative analysis of different baselines with our proposed algorithms in terms of the average utility and empirical competitive ratio. The average utility is empirically calculated as the average utility of the testing samples and is normalized by the optimal average utility. The competitive ratio is empirically calculated as the minimum ratio of an online algorithm's utility to the optimal utility among the testing samples. Because of the provably better asymptotic

		ML	OACP	OACP+	LA-OACP-0.3	LA-OACP-0.6	DMD	Greedy	Equal
AVG	In	**0.9340**	0.8959	0.9130	0.9311	0.9301	0.8715	0.8574	0.7246
	OOD	0.8975	0.9036	**0.9041**	0.8953	0.8981	0.9016	0.9000	0.7528
CR	In	**0.8645**	0.8481	0.8565	0.8303	0.8223	0.8200	0.8048	0.5650
	OOD	0.7916	0.8234	**0.8411**	0.7916	0.8003	0.8076	0.8048	0.5650

Table 2. Comparison of average utility (AVG) and empirical competitive ratio (CR). LA-OACP-n indicates LA-OACP with $\lambda = n$. "In" and "OOD" mean in-distribution and out-of-distribution, respectively. The average utility is normalized by that of OPT (i.e., 80.2771 and 78.8540 for the in-distribution and out-of-distribution cases, respectively).... Bold values represent the best for the respective metrics.

competitive ratio of OACP+, we use OACP+ in LA-OACP and set $R = 0$ in (7b) by default. All the utility values are normalized with respect to that of OPT.

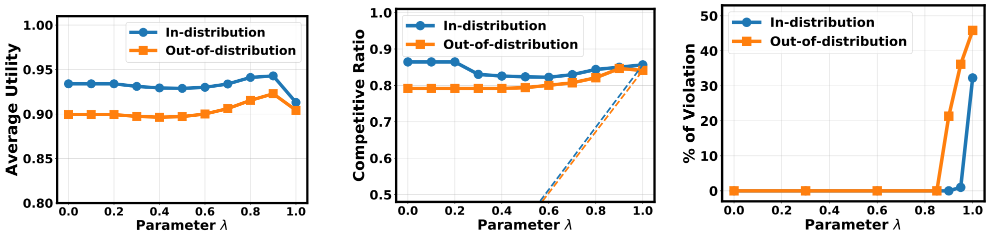
Comparison with baselines. We first compare OACP, OACP+ and LA-OACP with the baseline algorithms in Table 2 under an *in-distribution* case where the training-testing instances are drawn from the same distribution. Our results show that ML has the highest average utility among the considered online algorithms, while LA-OACP, OACP, and OACP+ outperform other baselines (DMD, Greedy and Equal) in terms of the average utility. Importantly, by setting $\lambda = 0.3$ and $\lambda = 0.6$, the average utilities of LA-OACP are both improved compared to OACP and OACP+, and closer to that of ML.

For the in-distribution testing case, the empirical competitive ratio of ML is also the best, although ML does not have a guaranteed competitive ratio. Besides, OACP and OACP+ both have higher competitive ratios than other baselines (DMD, Greedy, Equal), demonstrating their advantages in competitive ratio guarantees. Note that the empirical competitive ratios of OACP are higher than that of DMD which sets its resource price more aggressively, showing the benefit of conservative pricing in OACP. Moreover, while the empirical competitive ratios of LA-OACP are lower than ML, they have provable competitive ratio which is scaled down by λ compared to that of OACP+.

Training-testing distributional shifts. The above results consider that the training and testing instances are drawn from the same distribution. Now, we consider an out-of-distribution (*OOD*) testing case by adding perturbation noises to 30% of the testing instances, and show the results in Table 2. OOD is commonly seen in practice, making ML predictions potentially untrusted. Since the testing distribution shifts compared to the training distribution under the OOD setting, the performances of ML in terms of both average utility and competitive ratio decrease and become worse than OACP and OACP+. Still, OACP and OACP+ outperform the other baselines (DMD, Greedy and Equal) in terms of the empirical competitive ratio, again showing their benefits in the worst-case competitive guarantee. The learning-augmented algorithm LA-OACP improves the competitive ratio of ML with a large λ, showing its effects in providing the ML with guaranteed competitiveness.

Performance under varying λ. Next, we show in Fig. 1(a) the impact of $\lambda \in [0, 1]$ on LA-OACP in terms of the average utility. We see that under the in-distribution setting, when λ increases, the average utility of LA-OACP can decrease due to the increasingly more stringent worst-case robustness constraint (7b). Interestingly, LA-OACP can achieve higher average utility than ML under some λ. This is due to the fact that OACP+ used by LA-OACP can correct the ML predictions for some problem instances in which the original ML predictions do not perform well. For the OOD setting, the average utility of ML is lower due to the distribution shift. By integrating OACP+ with ML, LA-OACP is more beneficial in terms of improving the average utility. This confirms our analysis of LA-OACP in Theorems 4.1 and 4.2.

We show the empirical competitive ratios under different λ in Fig. 1(b). In practice, it is difficult to evaluate the competitive ratio empirically since the adversarial samples for the algorithms under evaluation may not exist in the actual testing dataset under evaluation. As a result, a few



(a) Average utility of LA-OACP

(b) Competitive ratio LA-OACP

(c) Constraint violation rate of ML

Fig. 1. (a) Average utility of LA-OACP with varying $\lambda \in [0, 1]$; (b) Empirical competitive ratio of LA-OACP with varying $\lambda \in [0, 1]$ (dotted lines represent the theoretical competitive ratio bounds); (c) Utility constraint (7b) violation probability by the pure ML predictor.

unfavorable instances can affect the empirical competitive ratio significantly. Our results show that LA-OACP has an empirical competitive ratio higher than the theoretical bound in Theorem 4.1 (dotted line in Fig. 1(b)), which is also very common in practice.

Finally, we show in Fig. 1(c) the worst-case utility constraint violation probability for the pure standalone ML predictor. Naturally, when λ increases, the worst-case utility constraint in (7b) becomes tighter, making the pure ML predictor violate the constraint more frequently. This highlights the lack of worst-case utility guarantees of pure ML, as well as the necessity of LA-OACP to safeguard the ML predictions. Thus, although ML empirically can have a good competitive ratio (against OPT) as shown in Table 2 for the in-distribution case, this empirical advantage is not always guaranteed.

6 RELATED WORKS

Online constrained allocation is a classic problem extensively studied in the last few decades. For example, some earlier works [20, 26] solve online allocation by estimating a fixed Lagrangian multiplier using offline data, while other studies design online algorithms by updating the Lagrangian multiplier or resource price in an online manner [1, 21, 63]. Likewise, online algorithms have also been proposed for online stochastic optimization with distributional information [32]. Online algorithms that allow budget violations are also available [42, 45, 46]. In the context of network optimization, Lyapunov optimization can address various resource constraints by introducing resource queues (equivalent to the Lagrangian multiplier), but the extension to adversarial settings with strict budget constraints is challenging [28, 29, 45, 61].

Online allocation with budget constraints in adversarial settings is very challenging and has not been fully resolved yet. Concretely, for online allocation with inventory constraints, competitive online algorithms are designed by pursuing a pseudo-optimal algorithm, but the utility function either takes a single scalar [41] or is separable over multiple dimensions [40]. A recent study [9] considers online allocation with a more general convex utility function and proposes dual mirror descent (DMD) to update the Lagrangian multiplier given stochastic inputs at each round, while the extension to adversarial settings has been considered more recently in [11] and extension to uncertain time horizons is studied in [8]. Nonetheless, these studies do not apply to online budget replenishment, which we address by proposing provably-competitive OACP and OACP+.

ML predictors/policies have been emerging for exploiting the distributional information of problem inputs and hence improving the average performance of various (online) optimization problems [18, 36, 54]. For example, online scheduling, resource management, and classic secretary problems [12, 35, 54, 62] have all been considered. Nonetheless, a major drawback of these standalone

ML-based optimizers is that they do not have worst-case performance guarantees and may have very high or even unbounded losses in the worst case. As a consequence, they may not be suitable for mission-critical applications. While constrained ML-based policies [22, 25, 33, 58] are available, they focus on orthogonal challenges (i.e., unknown cost/utility functions) and typically focus on the average constraint, rather than worst-case utility constraint for any problem instance.

LA-OACP is relevant to the emerging field of learning-augmented algorithms [15, 15, 19, 38, 44, 57]. The goal of typical learning-augmented algorithms is to improve the worst-case competitive ratio when the ML prediction is perfect, while bounding the worst-case competitive ratio when ML prediction is arbitrarily bad. While it has been considered in a variety of settings, a learning-augmented algorithm for online allocation with replenishable budgets is still lacking. Thus, LA-OACP addresses this gap and is the first learning-augmented algorithm for online allocation with replenishable budgets that offers worst-case utility guarantees for any problem instance.

7 CONCLUSION

In this paper, we study online resource allocation with replenishable budgets, and propose novel competitive algorithms, called OACP and OACP+, that conservatively adjusts dual variables while opportunistically utilizing available resources. We prove, for the first time, that OACP and OACP+ both achieve bounded asymptotic competitive ratios in adversarial settings as the number of decision rounds $T \to \infty$. In particular, under the mild assumption that the budget is replenished every T^* rounds, OACP+ can improve the asymptotic competitive ratio over OACP. Then, to address the conservativeness of OACP, we move beyond the worst-case and propose LA-OACP, a novel learning-augmented algorithm for our problem setting. LA-OACP can provably improve the average utility compared to OACP and OACP+ when the ML predictor is properly trained, while still offering worst-case utility guarantees. Finally, we perform simulation studies using online power allocation with energy harvesting. Our results validate our analysis and demonstrate the empirical benefits of LA-OACP compared to existing baselines.

ACKNOWLEDGMENTS

This work was supported in part by the NSF under grants CNS-1910208 and CNS-2007115.

REFERENCES

[1] Shipra Agrawal and Nikhil R Devanur. 2014. Fast algorithms for online stochastic convex programming. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*. 1405–1424.

[2] Mohammad Ali Alomrani, Reza Moravej, and Elias Boutros Khalil. 2022. Deep Policies for Online Bipartite Matching: A Reinforcement Learning Approach. *Transactions on Machine Learning Research* (2022). https://openreview.net/forum?id=mbwm7NdkpO

[3] Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J. Zico Kolter. 2018. Differentiable MPC for End-to-end Planning and Control. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/ba6d843eb4251a4526ce65d1807a9309-Paper.pdf

[4] Brandon Amos and J. Zico Kolter. 2017. OptNet: Differentiable Optimization as a Layer in Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 136–145.

[5] Antonios Antoniadis, Christian Coester, Marek Elias, Adam Polak, and Bertrand Simon. 2020. Online Metric Algorithms with Untrusted Predictions. In *ICML*.

[6] Sanjeev Arora, Elad Hazan, and Satyen Kale. 2012. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. *Theory of Computing* 8, 6 (2012), 121–164. https://doi.org/10.4086/toc.2012.v008a006

[7] Kamiar Asgari and Michael J. Neely. 2020. Bregman-Style Online Convex Optimization with Energy Harvesting Constraints. *Proc. ACM Meas. Anal. Comput. Syst.* 4, 3, Article 52 (nov 2020), 25 pages. https://doi.org/10.1145/3428337

[8] Santiago Balseiro, Christian Kroer, and Rachitesh Kumar. 2023. Online Resource Allocation under Horizon Uncertainty. In *Abstract Proceedings of the 2023 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer*

Systems (Orlando, Florida, United States) *(SIGMETRICS '23)*. Association for Computing Machinery, New York, NY, USA, 63–64. https://doi.org/10.1145/3578338.3593559

[9] Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. 2020. Dual mirror descent for online allocation problems. In *International Conference on Machine Learning*. PMLR, 613–628.

[10] Santiago R Balseiro and Yonatan Gur. 2019. Learning in repeated auctions with budgets: Regret minimization and equilibrium. *Management Science* 65, 9 (2019), 3952–3968.

[11] Santiago R. Balseiro, Haihao Lu, and Vahab Mirrokni. 2022. The Best of Many Worlds: Dual Mirror Descent for Online Allocation Problems. *Operations Research* 0, 0 (May 2022), null. https://doi.org/10.1287/opre.2021.2242

[12] Thomas Barrett, William Clements, Jakob Foerster, and Alex Lvovsky. 2020. Exploratory combinatorial optimization with reinforcement learning. In *AAAI*.

[13] Dimitri P Bertsekas. 2014. *Constrained optimization and Lagrange multiplier methods.* Academic press.

[14] Allan Borodin and Ran El-Yaniv. 2005. *Online computation and competitive analysis.* cambridge university press.

[15] Joan Boyar, Lene M. Favrholdt, Christian Kudahl, Kim S. Larsen, and Jesper W. Mikkelsen. 2016. Online Algorithms with Advice: A Survey. *SIGACT News* 47, 3 (Aug. 2016), 93–129.

[16] Stephen Boyd, Lin Xiao, and Almir Mutapcic. 2003. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter* 2004, 01 (2003).

[17] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[18] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. 2021. Learning to Optimize: A Primer and A Benchmark. arXiv:2103.12828 [math.OC]

[19] Nicolas Christianson, Tinashe Handina, and Adam Wierman. 2022. Chasing Convex Bodies and Functions with Black-Box Advice. In *COLT*.

[20] Nikhil R Devanur and Thomas P Hayes. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*. 71–78.

[21] Nikhil R Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A Wilkens. 2019. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. *Journal of the ACM (JACM)* 66, 1 (2019), 1–41.

[22] Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic. 2020. Natural policy gradient primal-dual method for constrained markov decision processes. *Advances in Neural Information Processing Systems* 33 (2020), 8378–8390.

[23] Bingqian Du, Zhiyi Huang, and Chuan Wu. 2022. Adversarial Deep Learning for Online Resource Allocation. *ACM Trans. Model. Perform. Eval. Comput. Syst.* 6, 4, Article 13 (feb 2022), 25 pages. https://doi.org/10.1145/3494526

[24] Bingqian Du, Chuan Wu, and Zhiyi Huang. 2019. Learning Resource Allocation and Pricing for Cloud Profit Maximization. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence* (Honolulu, Hawaii, USA) *(AAAI'19/IAAI'19/EAAI'19)*. AAAI Press, Article 929, 8 pages. https://doi.org/10.1609/aaai.v33i01.33017570

[25] Yonathan Efroni, Shie Mannor, and Matteo Pirotta. 2020. Exploration-exploitation in constrained mdps. *arXiv preprint arXiv:2003.02189* (2020).

[26] Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S Mirrokni, and Cliff Stein. 2010. Online stochastic packing applied to display ad allocation. In *European Symposium on Algorithms*. 182–194.

[27] Jean-Louis Goffin. 1977. On convergence rates of subgradient optimization methods. *Mathematical programming* 13 (1977), 329–347.

[28] L. Huang. 2020. Fast-Convergent Learning-Aided Control in Energy Harvesting Networks. *IEEE Transactions on Mobile Computing* 19, 12 (dec 2020), 2793–2803. https://doi.org/10.1109/TMC.2019.2936344

[29] Longbo Huang, Xin Liu, and Xiaohong Hao. 2014. The Power of Online Learning in Stochastic Network Optimization. *SIGMETRICS Perform. Eval. Rev.* 42, 1 (June 2014), 153–165.

[30] Longbo Huang and Michael J. Neely. 2011. Utility Optimal Scheduling in Energy Harvesting Networks. In *MobiHoc*.

[31] Stefanus Jasin and Sunil Kumar. 2012. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research* 37, 2 (2012), 313–345.

[32] Jiashuo Jiang, Xiaocheng Li, and Jiawei Zhang. 2020. Online Stochastic Optimization with Wasserstein Based Non-stationarity. *arXiv preprint arXiv:2012.06961* (2020).

[33] Abbas Kazerouni, Mohammad Ghavamzadeh, Yasin Abbasi Yadkori, and Benjamin Van Roy. 2017. Conservative Contextual Linear Bandits. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/bdc4626aa1d1df8e14d80d345b2a442d-Paper.pdf

[34] Zico Kolter, David Duvenaud, and Matt Johnson. http://implicit-layers-tutorial.org/. Deep Implicit Layers.

[35] Weiwei Kong, Christopher Liaw, Aranyak Mehta, and D. Sivakumar. 2019. A New Dog Learns Old Tricks: RL Finds Classic Optimization Algorithms. In *ICLR*. https://openreview.net/forum?id=rkluJ2R9KQ

[36] Ke Li and Jitendra Malik. 2017. Learning to Optimize. In *ICLR*.

[37] Pengfei Li, Jianyi Yang, and Shaolei Ren. 2022. Expert-Calibrated Learning for Online Optimization with Switching Costs. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 2, Article 28 (Jun 2022), 35 pages.

[38] Pengfei Li, Jianyi Yang, and Shaolei Ren. 2023. Robustified Learning for Online Optimization with Memory Costs. In *INFOCOM*.

[39] Tongxin Li, Ruixiao Yang, Guannan Qu, Guanya Shi, Chenkai Yu, Adam Wierman, and Steven Low. 2022. Robustness and Consistency in Linear Quadratic Control with Untrusted Predictions. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 1, Article 18 (feb 2022), 35 pages. https://doi.org/10.1145/3508038

[40] Qiulin Lin, Yanfang Mo, Junyan Su, and Minghua Chen. 2022. Competitive Online Optimization with Multiple Inventories: A Divide-and-Conquer Approach. *Proc. ACM Meas. Anal. Comput. Syst.* 6, 2, Article 36 (jun 2022), 28 pages. https://doi.org/10.1145/3530902

[41] Qiulin Lin, Hanling Yi, John Pang, Minghua Chen, Adam Wierman, Michael Honig, and Yuanzhang Xiao. 2019. Competitive Online Optimization under Inventory Constraints. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 1, Article 10 (mar 2019), 28 pages. https://doi.org/10.1145/3322205.3311081

[42] Qingsong Liu, Wenfei Wu, Longbo Huang, and Zhixuan Fang. 2021. Simultaneously Achieving Sublinear Regret and Constraint Violations for Online Convex Optimization with Time-Varying Constraints. *Performance Evaluation* 152 (2021), 102240. https://doi.org/10.1016/j.peva.2021.102240

[43] Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2023. Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model. *Journal of Machine Learning Research* 24, 253, 1–15. http://jmlr.org/papers/v24/23-0069.html

[44] Thodoris Lykouris and Sergei Vassilvitskii. 2021. Competitive Caching with Machine Learned Advice. *J. ACM* 68, 4, Article 24 (July 2021), 25 pages.

[45] M. J. Neely. 2010. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool.

[46] Michael J Neely. 2010. Universal scheduling for networks with arbitrary traffic, channels, and mobility. In *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 1822–1829.

[47] California Independent System Operator. 2023. Calfornia Renewable Datasets. https://www.caiso.com/Pages/default.aspx.

[48] Francesco Orabona. 2019. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213* (2019).

[49] D. P. Palomar and M. Chiang. 2007. Alternative Distributed Algorithms for Network Utility Maximization: Framework and Applications. *IEEE Trans. Automatic Control* 52, 12 (Dec. 2007), 2254–2269.

[50] Chengrun Qiu, Yang Hu, Yan Chen, and Bing Zeng. 2018. Lyapunov optimization for energy harvesting wireless sensor communications. *IEEE Internet of Things Journal* 5, 3 (2018), 1947–1956.

[51] Ana Radovanović, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, et al. 2022. Carbon-aware computing for datacenters. *IEEE Transactions on Power Systems* 38, 2 (2022), 1270–1280.

[52] Daan Rutten, Nicolas Christianson, Debankur Mukherjee, and Adam Wierman. 2023. Smoothed Online Optimization with Unreliable Predictions. *Proc. ACM Meas. Anal. Comput. Syst.* 7, 1, Article 12 (mar 2023), 36 pages. https://doi.org/10.1145/3579442

[53] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020. Green ai. *Commun. ACM* 63, 12 (2020), 54–63.

[54] Zhihui Shao, Jianyi Yang, Cong Shen, and Shaolei Ren. 2022. Learning for Robust Combinatorial Optimization: Algorithm and Application. In *INFOCOM*.

[55] Kalyan T Talluri, Garrett Van Ryzin, and Garrett Van Ryzin. 2004. *The theory and practice of revenue management*. Vol. 1. Springer.

[56] H. R. Varian. 1992. *Microeconomic Analysis*. W. W. Norton & Company.

[57] Alexander Wei and Fred Zhang. 2020. Optimal Robustness-Consistency Trade-offs for Learning-Augmented Online Algorithms. In *NeurIPS*.

[58] Yifan Wu, Roshan Shariff, Tor Lattimore, and Csaba Szepesvári. 2016. Conservative bandits. In *International Conference on Machine Learning*. PMLR, 1254–1262.

[59] Jianyi Yang and Shaolei Ren. 2023. Learning-Assisted Algorithm Unrolling for Online Optimization with Budget Constraints. In *AAAI*.

[60] Hao Yu and Michael J Neely. 2019. Learning-aided optimization for energy-harvesting devices with outdated state information. *IEEE/ACM Transactions on Networking* 27, 4 (2019), 1501–1514.

[61] Hao Yu and Michael J. Neely. 2020. A Low Complexity Algorithm with $O(\sqrt{T})$ Regret and $O(1)$ Constraint Violations for Online Convex Optimization with Long Term Constraints. *Journal of Machine Learning Research* 21, 1 (2020), 1–24.

[62] Han Zhang, Wenzhong Li, Shaohua Gao, Xiaoliang Wang, and Baoliu Ye. 2019. ReLeS: A Neural Adaptive Multipath Scheduler based on Deep Reinforcement Learning. In *INFOCOM*.

[63] Martin Zinkevich. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*. 928–936.

APPENDIX

A PROOF OF THEOREM 3.1

We now prove Theorem 3.1 and first restate the convergence lemma of online mirror descent.

Lemma A.1 ([11, 48]). *Let $V_h(x,y) = h(x) - h(y) - \nabla h(y)^\top (x-y)$ be the Bregman divergence based on a σ-strongly convex function h. If $w_t(\mu)$ is a convex function with respect to $\mu \in \mathcal{D}$ where $\mathcal{D}$ is a convex set and its sub-gradient satisfies $\|\partial_\mu w_t(\mu)\|_\infty \leq G$, by updating the variable $\mu_{t+1} = \arg\min_{\mu\in\mathcal{D}} \mu^\top \partial_\mu w_t(\mu) + \frac{1}{\eta} V_h(\mu,\mu_t)$ from some initial variable μ_1, it holds for any $\mu \in \mathcal{D}$ that*

$$\sum_{t=1}^{T} w_t(\mu_t) - w_t(\mu) \leq \frac{G^2\eta}{2\sigma} T + \frac{1}{\eta} V_h(\mu,\mu_1). \tag{13}$$

Proof of Theorem 3.1

Proof. We define $\mathcal{T}_A = \left\{\tau_1, \cdots, \tau_{|\mathcal{T}_A|}\right\}$ as a set of rounds when $\hat{x}_t$ violates the budget constraint, i.e. $\forall \tau \in \mathcal{T}_A$, there exists a dimension m such that $(\hat{x}_\tau)_m > (B_\tau + E_\tau)_m$. By our algorithm design, if $t \in \mathcal{T}_A$, we choose $x_t = 0$ and $g_t = 0$. Define a sequence of functions as

$$w_t(\mu) = \mu_t^\top g_t = \begin{cases} \mu_t^\top (\rho - \hat{x}_t), & t \notin \mathcal{T}_A, \\ 0, & t \in \mathcal{T}_A. \end{cases} \tag{14}$$

By Lemma A.1, we have

$$\sum_{t=1}^{T} w_t(\mu_t) - w_t(\mu) \leq \frac{G^2\eta}{2\sigma} T + \frac{1}{\eta} V_h(\mu,\mu_1), \tag{15}$$

where $G = \sup \|g_t\|_\infty \leq \bar{\rho} + \|\bar{x}\|_\infty$. By our algorithm design, $\forall t \notin \mathcal{T}_A$, the action is chosen as $x_t = \arg\max_{x\in\mathcal{X}}\{f_t(x) - \mu_t^\top x\}$, we have $f_t(x_t^*) \leq f_t(x_t) + \mu_t^\top(x_t^* - x_t)$ and $0 = f_t(0) \leq f_t(x_t) - \mu_t^\top x_t$. Thus we have

$$\begin{aligned} \alpha f_t(x_t) &= f_t(x_t) + (\alpha-1) f_t(x_t) \\ &\geq f_t(x_t^*) - \mu_t^\top x_t^* + \mu_t^\top x_t + (\alpha-1) f_t(x_t) \\ &\geq f_t(x_t^*) - \mu_t^\top x_t^* + \mu_t^\top x_t + (\alpha-1)\mu_t^\top x_t \\ &= f_t(x_t^*) - \alpha\mu_t^\top(\rho - x_t) - \mu_t^\top x_t^* + \alpha\mu_t^\top \rho \\ &\geq f_t(x_t^*) - \alpha w_t(\mu_t), \end{aligned} \tag{16}$$

where the last inequality holds by setting $\alpha = \max_{m\in[M]} \frac{\bar{x}_m}{\rho_m}$.

Then for any $\mu > 0$, we have

$$\begin{aligned}
&OPT(y) - \alpha F_T(y)\\
\leq &\sum_{t=1}^{T} f_t(x_t^*) - \alpha \sum_{t\notin\mathcal{T}_A} f_t(x_t)\\
\leq &\sum_{t=1}^{T} f_t(x_t^*) - \sum_{t\notin\mathcal{T}_A} f_t(x_t^*) + \sum_{t\notin\mathcal{T}_A} \alpha w_t(\mu_t)\\
\leq &\sum_{t\in\mathcal{T}_A} f_t(x_t^*) + \alpha \sum_{t\notin\mathcal{T}_A} w_t(\mu) + \alpha\left(\frac{G^2\eta}{2\sigma}T + \frac{1}{\eta}V_h(\mu,\mu_1)\right)\\
\leq &|\mathcal{T}_A|\bar{f} + \alpha \sum_{t\notin\mathcal{T}_A} \mu^\top(\rho - x_t) + \alpha\left(\frac{G^2\eta}{2\sigma}T + \frac{1}{\eta}V_h(\mu,\mu_1)\right)
\end{aligned} \tag{17}$$

where the first inequality holds because the utility are non-negative, the second inequality holds by (16), the third inequality holds by Lemma A.1, and the last inequality holds by $f_t \leq \bar{f}$.

Now it remains to choose μ to get the bound. If $|\mathcal{T}_A| = 0$, set $\mu = 0$, and the bound holds. Otherwise, we choose μ as follows. Define $\mathcal{M}_A$ is the set of resources of which the corresponding constraints are violated, i.e. for $m \in \mathcal{M}_A$, $\exists t \in [T]$ such that $\hat{x}_{m,t} > (B_t + E_t)_m$. Since the consumed resource plus $\hat{x}_{t,m}$ is larger than the initial budget $B_{1,m}$ when the constraint resource m is violated and $\hat{x}_{t,m} \leq \bar{x}_m$, it holds for resource $m \in \mathcal{M}_A$ that

$$\sum_{t\notin\mathcal{T}_A} x_{t,m} + \bar{x}_m \geq B_{1,m} = \rho_m T. \tag{18}$$

We choose one resource $j \in \mathcal{M}_A$ and set $\mu = \frac{\bar{f}}{\alpha\rho_j} e_j$ where e_j is a unit vector with jth entry being one and other entries being zero, it holds that

$$\begin{aligned}
&\alpha \sum_{t\notin\mathcal{T}_A} \mu^\top(\rho - x_t)\\
=&\alpha \sum_{t\notin\mathcal{T}_A} \mu_j(\rho_j - x_{t,j})\\
\leq&\alpha(T - |\mathcal{T}_A|)\mu_j\rho_j - \alpha\mu_j(T\rho_j - \bar{x}_j)\\
\leq& - \alpha|\mathcal{T}_A|\mu_j\rho_j + \alpha\mu_j\bar{x}_j\\
\leq& - |\mathcal{T}_A|\bar{f} + \alpha\bar{f},
\end{aligned} \tag{19}$$

where the first inequality holds by (18), and the last inequality holds by the choice of μ.

Substituting (19) into (17), we get the bound as

$$OPT(y) - \alpha R_T^{DMD}(y) \leq \alpha\bar{f} + \frac{\alpha G^2\eta T}{2\sigma} + \frac{\alpha}{\eta}V_h(\mu,\mu_1), \tag{20}$$

where $\alpha = \sup_{m\in[M]} \frac{\bar{x}_m}{\rho_m}$, and $\mu = 0$ if $\mathcal{M}_A = \emptyset$. Otherwise, $\mu = \frac{\bar{f}}{\alpha\rho_j} e_j$, $j = \arg\min_m V_h(\frac{\bar{f}}{\alpha\rho_m} e_m, \mu_1), m \in \mathcal{M}_A\}$. Thus, we complete the proof. □

B PROOF OF THEOREM 3.2

Lemma B.1. *If a fixed budget $B^{(i)} = 2^{i-1}T^*\rho + \Omega_i$ where $\Omega_i = \min\{B_{T_{i-1}+1} - (T - (2^{i-1} - 1)T^*)\rho, 2^{i-2}T^*\rho_{\max} \odot \beta\}$ where $\rho_{\max} = B_{\max}/T$ is assigned to each frame $i, 1 \leq i \leq K$ with $2^{i-1}T^*$*

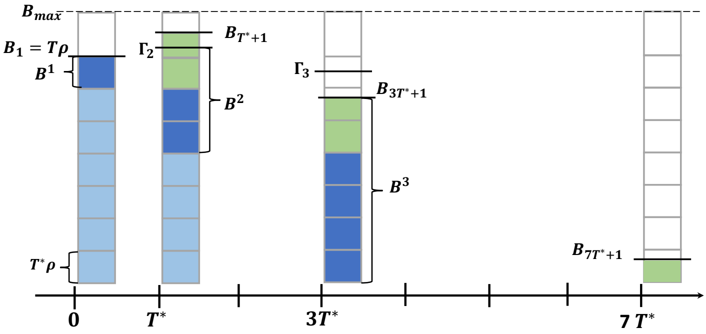


Fig. 2. An example of budget assignment with $T = 7T^*$. Colored rectangles indicate the amount of remained budget and white rectangles are the spaces in the storage. Dark blue rectangles indicate permanent budgets $2^{i-1}T^*\rho$ for the current frame. Light blue rectangles indicate permanent budgets for the future frames $(T - (2^{(i)} - 1)T^*)\rho$. Green rectangles indicate the budget accumulation $\min\{B_{T_{i-1}+1} - (T - (2^{i-1} - 1)T^*)\rho, 2^{i-2}T^*\rho_{\max} \odot \beta\}$.

rounds, the additive budget Ω_i is greater or equal to equivalent additive budget $\hat{\Omega}_i, 1 \le i \le K - 1$ which is expressed as

$$\hat{\Omega}_1 = 0 \tag{21a}$$

$$\hat{\Omega}_2 = \min\{T\rho_{\max} - T\rho, E'_{\min}\} \tag{21b}$$

$$\hat{\Omega}_i = \min\{T\rho_{\max} - 2^{i-3}T^*\rho_{\max} \odot \beta - (T - (2^{i-2} - 1)T^*)\rho, 2^{i-2}E'_{\min}\}, 3 \le i \le K \tag{21c}$$

where $E'_{\min} = \min\{E_{\min}, T^*\rho_{\max} \odot \beta\}$.

Proof. We prove that the equivalent additive budget $\hat{\Omega}_i$ does not exceed the true additive budget Ω_i for any frame i.

For the first frame, it is obvious that $\hat{\Omega}_1 \le \Omega_1 = 0$ holds. For the second frame, we discuss the value of Ω_2 in the following cases.

Firstly, if for a resource $m \in [M]$, $B_{T^*+1,m} - (T - T^*)\rho_m \le T^*\beta_m\rho_{\max,m}$, the additive budget $\Omega_{2,m}$ is $B_{T^*+1,m} - (T - T^*)\rho_m$, and it comprises the replenishment in the first frame $\sum_{t=1}^{T_1} E_{t,m}$ and the unconsumed budget in the first frame $B_m^1 - \sum_{t=1}^{T_1} x_{t,m}$. We can bound the replenishment in the first frame as

$$\sum_{t=1}^{T_1} E_{t,m} \ge \min\{B_{\max,m} - T\rho_m, E_{\min,m}\} \ge \min\{B_{\max,m} - T\rho, E'_{\min,m}\} = \hat{\Omega}_{2,m}. \tag{22}$$

The reason is that if the truly replenished budget of resource m at each round of the first frame is not constrained by $B_{\max,m}$, i.e. $E_{t,m} = \hat{E}_{t,m}, \forall t \in [1, T_1]$, we have $\sum_{t=1}^{T_1} E_{t,m} = \sum_{t=1}^{T_1} \hat{E}_{t,m} \ge E_{\min,m} \ge E'_{\min,m}$. Otherwise, we must have $\sum_{t=1}^{T_1} E_{t,m} \ge B_{\max,m} - T\rho_m$ since $B_{\max,m} - T\rho_m$ is the minimum replenished budget such that the replenishment is constrained by the budget cap $B_{\max,m}$. Therefore for the first case, we always have for the resource m, $\hat{\Omega}_{2,m} \le \sum_{t=1}^{T_1} E_{t,m} \le \Omega_{2,m}$.

For the second case when $B_{T^*+1,m} - (T - T^*)\rho_m > T^*\beta_m\rho_{\max,m}$ for resource m, we have $\Omega_{2,m} = T^*\beta_m\rho_{\max,m}$. Thus, we still have $\hat{\Omega}_{2,m} \le E'_{\min,m} \le T^*\beta_m\rho_{\max,m} = \Omega_{2,m}$.

Since the inequality holds for all the resources m, we have $\hat{\Omega}_2 \le \Omega_2$.

For the ith ($3 \le i \le K$) frame, we discuss for the value of Ω_i in the following cases.

Firstly, if for a resource m, $B_{T_{i-1}+1,m} - (T - (2^{i-1} - 1)T^*)\rho_m \le 2^{i-2}T^*\beta_m\rho_{\max,m}$, then the additive budget $\Omega_{i,m}$ includes the replenishment in the $(i-1)$th frame $\sum_{t=T_{i-2}+1}^{T_{i-1}} E_{t,m}$, the unconsumed assigned budget in the $(i-1)$th frame $B_m^{i-1} - \sum_{t=T_{i-2}+1}^{T_{i-1}} x_{t,m}$, and the possibly saved budget $[B_{T_{i-2}+1,m} - (T - (2^{i-2} - 1)T^*)\rho_m - 2^{i-3}T^*\beta_m\rho_{\max,m}]^+$ at the beginning of $(i-1)$th frame. The truly replenished budget in the $(i-1)$th frame can be bounded as

$$\sum_{t=T_{i-2}+1}^{T_{i-1}} E_{t,m} \ge \min\{B_{\max,m} - B_{T_{i-2}+1,m}, 2^{i-2}E'_{\min,m}\}. \tag{23}$$

The reason is that if the replenishment at each round of the $(i-1)$th frame is not constrained by $B_{\max,m}$, i.e. $E_{t,m} = \hat{E}_{t,m}, \forall t \in [T_{i-2}+1, T_{i-1}]$, we have $\sum_{t=T_{i-2}+1}^{T_{i-1}} E_{t,m} = \sum_{t=T_{i-2}+1}^{T_{i-1}} \hat{E}_{t,m} \ge 2^{i-2}E_{\min,m} \ge 2^{i-2}E'_{\min,m}$. Otherwise, we must have $\sum_{t=T_{i-2}+1}^{T_{i-1}} E_{t,m} \ge B_{\max,m} - B_{T_{i-2}+1,m}$ since $B_{\max,m} - B_{T_{i-2}+1,m}$ is the minimum replenished budget such that the replenishment is constrained by the budget cap $B_{\max,m}$.

If it holds at the beginning of the $(i-1)$th frame that $B_{T_{i-2}+1,m} \le (T - (2^{i-2} - 1)T^*)\rho_m + 2^{i-3}T^*\beta_m\rho_{\max,m}$, we further have

$$\Omega_{i,m} \ge \sum_{t=T_{i-2}+1}^{T_{i-1}} E_{t,m} \ge \min\{B_{\max,m} - 2^{i-3}T^*\beta_m\rho_{\max,m} - (T - (2^{i-2} - 1)T^*)\rho_m, 2^{i-2}E'_{\min,m}\} = \hat{\Omega}_{i,m}. \tag{24}$$

Otherwise, $[B_{T_{i-2}+1,m} - (T - (2^{i-2} - 1)T^*)\rho_m - 2^{i-3}T^*\beta_m\rho_{\max,m}]^+$ is positive and is included in $\Omega_{i,m}$. Under such a case, we have

$$\begin{aligned}\hat{\Omega}_{i,m} &= \min\{(T - 2^{i-3}T^*\beta_m)\rho_{\max,m} - (T - (2^{i-2} - 1)T^*)\rho_m, 2^{i-2}E'_{\min,m}\} \\ &\le \min\{B_{\max,m} - B_{T_{i-2}+1,m}, 2^{i-2}E'_{\min,m}\} + B_{T_{i-2}+1,m} - (T - (2^{i-2} - 1)T^*)\rho_m - 2^{i-3}T^*\beta_m\rho_{\max,m} \\ &\le \sum_{t=T_{i-2}+1}^{T_{i-1}} E_{t,m} + B_{T_{i-2}+1,m} - (T - (2^{i-2} - 1)T^*)\rho_m - 2^{i-3}T^*\beta_m\rho_{\max,m} \le \Omega_{i,m},\end{aligned} \tag{25}$$

where the first inequality holds because $\min\{A + B, C\} \le \min\{A, C\} + B$ for $A, B, C \ge 0$, the second inequality holds by (23), and the last inequality holds since $\sum_{t=T_{i-2}+1}^{T_{i-1}} E_{t,m}$ and $[B_{T_{i-2}+1,m} - (T - (2^{i-2} - 1)T^*)\rho_m - 2^{i-3}T^*\beta_m\rho_{\max,m}]^+$ are both included in $\Omega_{i,m}$.

Secondly, if $B_{T_{i-1}+1,m} - (T - (2^{i-1} - 1)T^*)\rho_m > 2^{i-2}T^*\beta_m\rho_{\max,m}$, the additive budget $\Omega_{i,m} = 2^{i-2}T^*\beta_m\rho_{\max,m}$, and we have $\hat{\Omega}_{i,m} \le 2^{i-2}E'_{\min,m} \le 2^{i-2}T^*\beta_m\rho_{\max,m} = \Omega_{i,m}$.

Since the inequality holds for all the resources m, we have $\hat{\Omega}_i \le \Omega_i$ for $3 \le i \le K$. □

Proof of Theorem 3.2

Proof. Since dual mirror descent is applied to each frame, using similar techniques as the proof of Theorem 3.1, we can prove that within each frame $i, i \in [K]$, given the choice of η and μ, it holds that

$$\sum_{t=T_{i-1}}^{T_i} f_t(x_t^*) - \alpha_i f_t(x_t) \le \alpha_i \bar{f} + \alpha_i(\bar{\rho}^{(i)} + \|\bar{x}\|_\infty)\sqrt{\frac{V_h(\mu, \mu_1)(2^{i-1}T^*)}{2\sigma}}, \tag{26}$$

where x_t^* is the offline-optimal solution for the whole episode with length T, $\alpha_i = \sup_{m\in[M]} \frac{\bar{x}_m}{\rho_m^{(i)}}$, and $\bar{\rho}^{(i)} = \sup_{m\in[M]} \rho_m^{(i)}$.

To use the doubling trick, we need to bound $\rho^{(i)} = \frac{B^{(i)}}{2^{i-1}T^*}$. By Lemma B.1, we have $\rho^1 = \rho$, $\rho^2 = \frac{2T^*\rho+\Omega_i}{2T^*} \geq \frac{2T^*\rho+\min\{T\rho_{\max}-T\rho, E'_{\min}\}}{2T^*} = \rho + \min\{\frac{T\rho_{\max}-T\rho}{2T^*}, \frac{\rho_{\max}\odot\beta}{2}, \frac{E_{\min}}{2T^*}\}$, and for $3 \leq i \leq K$, we have

$$\begin{aligned}\rho^{(i)} =& B^{(i)}/(2^{i-1}T^*) = \frac{2^{i-1}T^*\rho + \Omega_i}{2^{i-1}T^*} \geq \frac{2^{i-1}T^*\rho + \hat{\Omega}_i}{2^{i-1}T^*} \\ =&\rho + \frac{\min\left\{T\rho_{\max} - 2^{i-3}T^*\rho_{\max}\odot\beta - (T-(2^{i-2}-1)T^*)\rho, 2^{i-2}E'_{\min}\right\}}{2^{i-1}T^*} \\ =&\rho + \min\left\{\frac{1}{2^{i-1}}\left(\frac{T\rho_{\max}}{T^*} - \frac{T+T^*}{T^*}\rho\right) + \frac{\rho}{2} - \frac{\rho_{\max}\odot\beta}{4}, \frac{\rho_{\max}\odot\beta}{2}, \frac{E_{\min}}{2T^*}\right\},\end{aligned} \tag{27}$$

where the first inequality holds since $\Omega_i \geq \hat{\Omega}_i$, and the last equality holds since $E'_{\min} = \min\{E_{\min}, T^*\rho_{\max}\odot\beta\}$. If it holds for a resource m that $B_{\max,m} < (T+T^*)\rho_m$, we have $\frac{1}{2^{i-1}}\left(\frac{T\rho_{\max,m}}{T^*} - \frac{T+T^*}{T^*}\rho_m\right) + \frac{\rho_m}{2} - \frac{\beta_m\rho_{\max,m}}{4} \geq \frac{T\rho_{\max,m}-(T+T^*)\rho_m}{4T^*} + \frac{\rho_m}{2} - \frac{\beta\rho_{\max,m}}{4}$. By optimally choosing $\beta_m = \frac{T}{3T^*} - \frac{T-T^*}{3T^*}\frac{\rho_m}{\rho_{\max,m}}$, we have

$$\begin{aligned}\rho_m^{(i)} \geq& \rho_m + \min\left\{\frac{T\rho_{\max,m} - (T+T^*)\rho_m}{4T^*} + \frac{\rho_m}{2} - \frac{\beta_m\rho_{\max,m}}{4}, \frac{\beta_m\rho_{\max,m}}{2}, \frac{E_{\min,m}}{2T^*}\right\} \\ =&\rho_m + \min\left\{\frac{T\rho_{\max,m}}{6T^*} - \frac{(T-T^*)\rho_m}{6T^*}, \frac{E_{\min,m}}{2T^*}\right\}.\end{aligned} \tag{28}$$

If it holds for a resource m that $B_{\max,m} \geq (T+T^*)\rho_m$, we have $\frac{1}{2^{i-1}}\left(\frac{T\rho_{\max,m}}{T^*} - \frac{T+T^*}{T^*}\rho_m\right) + \frac{\rho_m}{2} - \frac{\beta_m\rho_{\max,m}}{4} \geq \frac{1}{2^{K-1}}\left(\frac{T\rho_{\max,m}}{T^*} - \frac{T+T^*}{T^*}\rho_m\right) + \frac{\rho_m}{2} - \frac{\beta_m\rho_{\max,m}}{4} \geq \frac{T\rho_{\max,m}-(T+T^*)\rho_m}{T+T^*} + \frac{\rho_m}{2} - \frac{\beta_m\rho_{\max,m}}{4}$ given that $T \geq (2^{K-1}-1)T^*$. By optimally choosing $\beta_m = \frac{4T}{3(T+T^*)} - \frac{2\rho_m}{3\rho_{\max,m}}$, we have

$$\begin{aligned}\rho_m^{(i)} \geq& \rho_m + \min\left\{\frac{T\rho_{\max,m} - (T+T^*)\rho_m}{T+T^*} + \frac{\rho_m}{2} - \frac{\beta_m\rho_{\max,m}}{4}, \frac{\beta_m\rho_{\max,m}}{2}, \frac{E_{\min,m}}{2T^*}\right\} \\ =&\rho_m + \min\left\{\frac{2T\rho_{\max,m}}{3(T+T^*)} - \frac{\rho_m}{3}, \frac{E_{\min,m}}{2T^*}\right\}.\end{aligned} \tag{29}$$

Therefore, we can bound $\rho_m^{(i)}$ as $\rho_m^{(i)} \geq \rho_m + \min\left\{\frac{2T\rho_{\max,m}}{3(T+T^*)} - \frac{\rho_m}{3}, \frac{E_{\min,m}}{2T^*}\right\}$ when $B_{\max,m} \geq (T+T^*)\rho_m$ and $\rho_m^{(i)} \geq \rho_m + \min\left\{\frac{T\rho_{\max,m}}{6T^*} - \frac{(T-T^*)\rho_m}{6T^*}, \frac{E_{\min,m}}{2T^*}\right\}$ when $B_{\max,m} < (T+T^*)\rho_m$. We define $\Delta\rho_m = \min\{\frac{2T\rho_{\max,m}}{3(T+T^*)} - \frac{\rho_m}{3}, \frac{E_{\min,m}}{2T^*}\}$ when $B_{\max,m} \geq (T+T^*)\rho_m$ and $\Delta\rho_m = \min\left\{\frac{T\rho_{\max,m}}{6T^*} - \frac{(T-T^*)\rho_m}{6T^*}, \frac{E_{\min,m}}{2T^*}\right\}$ when $B_{\max,m} < (T+T^*)\rho_m$. Thus, we have $\rho_m^{(i)} \geq \rho_m + \Delta\rho_m$.

Also, we can get the upper bound of $\rho^{(i)}$ for $i \in [2, K]$ as $\rho^{(i)} \leq \frac{2^{i-1}T^*\rho + 2^{i-2}T^*\rho_{\max}\odot\beta}{2^{i-1}T^*} = \rho + \frac{\rho_{\max}\odot\beta}{2}$, where the inequality holds because $\Omega_i \leq 2^{i-2}T^*(\beta\odot\rho_{\max})$. Thus we have $\bar{\rho}^{(i)} = \sup_{m\in[M]} \rho_m^{(i)} \leq \bar{\rho} + \frac{\bar{\beta}}{2}\bar{\rho}_{\max}$, where $\bar{\rho}_{\max} = \max_m \rho_{\max,m}$, $\bar{\beta} = \max_m \beta_m$. When $B_{\max,m} \geq (T+T^*)\rho_m$, the optimal $\rho_m \leq \frac{4}{3}$ as $T \longrightarrow \infty$. When $B_{\max,m} < (T+T^*)\rho_m$, the optimal $\rho_m \leq \frac{2}{3}$ as $T \longrightarrow \infty$ since $\frac{T}{T+T^*} < \frac{\rho_m}{\rho_{\max,m}} \leq 1$.

Define $\hat{\alpha} = \min_{m \in [M]} \frac{\bar{x}_m}{\rho_m + \Delta\rho_m}$. By summing up frames with the lower and upper bounds of $\rho^{(i)}$, we get

$$
\begin{aligned}
\sum_{t=1}^{T} f_t(x_t^*) - \hat{\alpha} f_t(x_t) &\leq \sum_{t=1}^{3T^*} f_t(x_t^*) - \hat{\alpha} f_t(x_t) + \sum_{i=3}^{K} \sum_{t=T_{i-1}}^{T_i} f_t(x_t^*) - \alpha_i f_t(x_t) \\
&\leq 3\bar{f}T^* + \sum_{i=3}^{K} \alpha_i \bar{f} + \alpha_i (\bar{\rho} + \frac{\bar{\beta}}{2} \bar{\rho}_{\max} + \|\bar{x}\|_\infty) \sqrt{\frac{V_h(\mu, \mu_1)(2^{i-1}T^*)}{2\sigma}} \\
&\leq 3\bar{f}T^* + \hat{\alpha} K \bar{f} + \hat{\alpha} (\bar{\rho} + \frac{\bar{\beta}}{2} \bar{\rho}_{\max} + \|\bar{x}\|_\infty) \sqrt{\frac{V_h(\mu, \mu_1)}{2\sigma}} \sum_{i=3}^{K} \sqrt{(2^{i-1}T^*)} \\
&\leq 3\bar{f}T^* + \hat{\alpha} K \bar{f} + \hat{\alpha} (\bar{\rho} + \frac{\bar{\beta}}{2} \bar{\rho}_{\max} + \|\bar{x}\|_\infty) \sqrt{\frac{V_h(\mu, \mu_1)}{2\sigma}} (1 + \sqrt{2}) \sqrt{T},
\end{aligned} \tag{30}
$$

where the second inequality holds by (26) and the third inequality holds due to the fact that $\hat{\alpha} \geq \alpha_i$ for any $i \in [3, K]$.

Since $K = \lceil \log_2(T/T^*) \rceil = O(\log(T))$, it holds for any sequence y that

$$
\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} f_t(x_t^*) - \hat{\alpha} f_t(x_t) \leq 0, \tag{31}
$$

indicating an asymptotic competitive ratio of $CR^{\text{OACP+}} = \frac{1}{\hat{\alpha}} = \min_{m \in [M]} \frac{\rho_m + \Delta\rho_m}{\bar{x}_m}$. □

C PROOF OF THEOREM 4.1

Proof. To prove the wost-case robustness of LA-OACP, we need to prove that there exists at least one feasible action in each round. We prove by induction that $\check{x}_t = \min\{x_t^\dagger, B_t + E_t\}$ is always feasible for constraint (8).

When $t = 1$, $x_t^\dagger$ is obviously a feasible solution of (8). Let $F_t = \sum_{\tau=1}^{t} f_\tau(x_\tau)$ for any $t \in [T]$. Assume that at round $t-1$, $F_{t-1} - \Delta(x_{t-1}) + R \geq \lambda F_{t-1}^\dagger$. At round t, we have

$$
\begin{aligned}
& F_t - \Delta(x_t) + R - \lambda F_t^\dagger \\
=& F_{t-1} - \lambda F_{t-1}^\dagger - \Delta(x_t) + R + f_t(x_t) - \lambda f_t(x_t^\dagger) \\
\geq& \left(\Delta(x_{t-1}) - \Delta(x_t)\right) + f_t(x_t) - \lambda f_t(x_t^\dagger) \\
=& \lambda L \left(\sum_{m=1}^{M} |B_{m,t}^\dagger - B_{m,t}|^+ - |B_{m,t+1}^\dagger - B_{m,t+1}|^+ \right) + f_t(x_t) - \lambda f_t(x_t^\dagger),
\end{aligned} \tag{32}
$$

where $B_{m,t+1} = B_{m,t} + E_{m,t} - x_{m,t}$ and $B_{m,t+1}^\dagger = B_{m,t}^\dagger + E_{m,t}^\dagger - x_{m,t}^\dagger$ by the budget dynamics.

Next, we prove $x_t = \check{x}_t$ is always a feasible solution for constraint (8). If $x_t = \check{x}_t$, we have $B_{t+1} = B_t + E_t - \check{x}_t$. If $B_{m,t} + E_{m,t} \geq x_{m,t}^\dagger$ holds for m, then $\check{x}_{m,t} = x_{m,t}^\dagger$ and we have

$$
\begin{aligned}
|B_{m,t+1}^\dagger - B_{m,t+1}|^+ &= |B_{m,t}^\dagger + E_{m,t}^\dagger - B_{m,t} - E_{m,t}|^+ \\
&= |\min\{B_{m,t}^\dagger + \hat{E}_{m,t}, B_{\max}\} - \min\{B_{m,t} + \hat{E}_{m,t}, B_{\max}\}|^+ \\
&\leq |B_{m,t}^\dagger - B_{m,t}|^+,
\end{aligned} \tag{33}
$$

where the last inequality holds by 1-Lipschitz of the function $\min\{\cdot, B_{\max}\}$. On the other hand, if $B_{m,t} + E_{m,t} < x^{\dagger}_{m,t}$ holds for m, then $\check{x}_{m,t} = B_{m,t} + E_t$ holds for m. Thus

$$\begin{aligned}
&|B^{\dagger}_{m,t} - B_{m,t}|^{+} - |B^{\dagger}_{m,t+1} - B_{m,t+1}|^{+} \\
=&(B^{\dagger}_{m,t} - B_{m,t}) - |B^{\dagger}_{m,t} + E^{\dagger}_{m,t} - x^{\dagger}_{m,t} - B_{m,t+1}|^{+} \\
=&- B_{m,t} - E^{\dagger}_{m,t} + x^{\dagger}_{m,t} \\
\geq& x^{\dagger}_{m,t} - \check{x}_{m,t},
\end{aligned} \tag{34}$$

where the first equality holds because $\min\{B_{m,t} + \hat{E}_{m,t}, B_{\max}\} = B_{m,t} + E_{m,t} < x^{\dagger}_{m,t} \leq B^{\dagger}_{m,t} + E^{\dagger}_{m,t} = \min\{B^{\dagger}_{m,t} + \hat{E}_{m,t}, B_{\max}\}$, so $B_{m,t} \leq B^{\dagger}_{m,t}$, the second equality holds because $B_{m,t+1} = B_{m,t} + E_{m,t} - \check{x}_{m,t} = 0$, and the inequality holds because $E_{m,t} \geq E^{\dagger}_{m,t}$ given $B_{m,t} \leq B^{\dagger}_{m,t}$. Thus we have for any $m \in [M]$,

$$L\left(\sum_{m=1}^{M} |B^{\dagger}_{m,t} - B_{m,t}|^{+} - |B^{\dagger}_{m,t+1} - B_{m,t+1}|^{+}\right) \geq L(x^{\dagger}_{m,t} - \bar{x}_{m,t}). \tag{35}$$

Thus, by the Lipschiz continuity of f, we have

$$\begin{aligned}
&f_t(x^{\dagger}_t) - f_t(\check{x}_t) \\
\leq& \sum_{m=1}^{M} L|x^{\dagger}_{m,t} - \check{x}_{m,t}| \\
\leq& \sum_{m=1}^{M} L(|B^{\dagger}_{m,t} - B_{m,t}|^{+} - |B^{\dagger}_{m,t+1} - B_{m,t+1}|^{+}).
\end{aligned} \tag{36}$$

Continuing with (32), when $x_t = \check{x}_t$, since $\lambda \in [0, 1]$, we have

$$F_t - \Delta(x_t) + R - \lambda F^{\dagger}_t \geq (1 - \lambda) f_t(\check{x}_t) \geq 0. \tag{37}$$

Thus we prove that there always exists $\check{x}_t = \min\{x^{\dagger}_t, B_t + E_t\}$ such that $F_t - \Delta(x_t) + R \leq \lambda F^{\dagger}_t$ holds for each round t. Since $\Delta(x_t) \geq 0$, if (8) holds for each round, we have (8) holds for the last round, thus satisfying the worst-case utility constraint (7b). □

D PROOF OF THEOREM 4.2

Proof. The ML policy optimally trained aware of the projection for worst-case utility constraint is the policy that optimizes the average utility that satisfies (8) for each round. Thus we bound the average utility by bounding the average utility of the policy π° based on the optimal unconstrained ML policy $\tilde{\pi}^*$ and `OACP` $\pi^{\dagger}$, i.e. $\pi^{\circ} = \gamma\tilde{\pi}^* + (1 - \gamma)\pi^{\dagger}$. The constructed policy π° gives the action $x^{\circ}_t = \gamma\tilde{x}^*_t + (1 - \gamma)x^{\dagger}_t$ where $\tilde{x}^*_T$ is the output of ML policy $\tilde{\pi}^*$ and $x^{\dagger}_t$ is the output of $\pi^{\dagger}$.

We first prove that x°_t is always a feasible action for the budget constraints. To show this, we prove by induction that the remaining budget B°_t of π° at each round is no less than a linear combination of the remaining budget $\tilde{B}^*$ of $\tilde{\pi}^*$ and the remaining budget $B^{\dagger}$ of $\pi^{\dagger}$. At the first round, it holds that

$$\begin{aligned}
B^{\circ}_2 &= \min\{B_1 + \hat{E}_1, B_{\max}\} - x^{\circ}_1 \\
&= \gamma\tilde{B}^*_2 + (1 - \gamma)B^{\dagger}_2.
\end{aligned} \tag{38}$$

Assume for the round $t, t > 2$, we have $B^\circ_t \geq \gamma \tilde{B}^*_t + (1-\gamma)B^\dagger_t$. Then we have

$$\begin{aligned} B^\circ_{t+1} &= \min\{B^\circ_t + \hat{E}_t, B_{\max}\} - x^\circ_t \\ &\geq \min\{\gamma \tilde{B}^*_t + (1-\gamma)B^\dagger_t + \hat{E}_t, B_{\max}\} - \gamma \tilde{x}^*_t - (1-\gamma)x^\dagger_t \\ &\geq \gamma \left(\min\{\tilde{B}^*_t + \hat{E}_t, B_{\max}\} - \tilde{x}^*_t\right) + (1-\gamma)\left(\min\{B^\dagger_t + \hat{E}_t, B_{\max}\} - x^\dagger_t\right) \\ &= \gamma \tilde{B}^*_{t+1} + (1-\gamma)B^\dagger_{t+1}, \end{aligned} \tag{39}$$

where the second inequality holds because $\min\{\cdot, B_{\max}\}$ is a concave function. Thus, for any round $t \in [T]$, we have $B^\circ_t \geq \gamma \tilde{B}^*_t + (1-\gamma)B^\dagger_t$. Since the ML policy and OACP both guarantee that $\tilde{B}^*_t \geq 0$ and $B^\dagger_t \geq 0$, we have $B^\circ_t \geq 0$ which means x°_t is a feasible action for budget constraints.

Next, we need to find an γ such that the policy π° satisfy the robustness constraints. By the robust algorithm design, we need to satisfy the robust constraint for each step t which can be expressed as

$$\sum_{i=1}^t f_i(x^\circ_i) \geq \lambda \sum_{i=1}^t f_i(x^\dagger_i) + \lambda L \sum_{m=1}^M |B^\dagger_{m,t+1} - B^\circ_{m,t+1}|^+ - R \tag{40}$$

By Lipschitz continuity of f_t, we have $f_i(x^\dagger_i) \leq f_i(x^\circ_i) + L\|x^\dagger_i - x^\circ_i\|_1$ (We can use L^1-norm since it returns the largest value among L^p-norms ($p \geq 1$).), and thus get a sufficient condition for the robust constriant (40) as

$$-\lambda L \sum_{i=1}^t \|x^\circ_i - x^\dagger_i\|_1 - \lambda L \sum_{m=1}^M |B^\dagger_{m,t+1} - B^\circ_{m,t+1}|^+ \geq (\lambda - 1)\sum_{i=1}^t f_i(x^\circ_i) - R. \tag{41}$$

By (39) and the monotonicity of ReLU operation, we have

$$|B^\dagger_{m,t+1} - B^\circ_{m,t+1}|^+ \leq |B^\dagger_{m,t+1} - \gamma \tilde{B}^*_{m,t+1} - (1-\gamma)B^\dagger_{m,t+1}|^+ = \gamma |B^\dagger_{m,t+1} - \tilde{B}^*_{m,t+1}|^+. \tag{42}$$

Substituting the expressions of x°_t and (42) into the inequality, the sufficient condition for the robust constraint (40) becomes

$$-\gamma\lambda L \sum_{i=1}^t \|\tilde{x}^*_i - x^\dagger_i\|_1 - \gamma\lambda L \sum_{m=1}^M |B^\dagger_{m,t+1} - \tilde{B}^*_{m,t+1}|^+ \geq (\lambda - 1)\sum_{i=1}^t f_i(x^\circ_i) - R. \tag{43}$$

By the definition of $B^\dagger_t$ and $\tilde{B}^*_t$, we have

$$\begin{aligned} &\sum_{m=1}^M |B^\dagger_{m,t+1} - \tilde{B}^*_{m,t+1}|^+ \\ =&\sum_{m=1}^M \left|\left(\min\{B^\dagger_{m,t} + \hat{E}_{m,t}, B_{\max}\} - x^\dagger_{m,t}\right) - \min\{\tilde{B}^*_{m,t} + \hat{E}_{m,t}, B_{\max}\} - \tilde{x}^*_{m,t}\right|^+ \\ \leq&\sum_{m=1}^M |B^\dagger_{m,t} - \tilde{B}^*_{m,t}|^+ + |x^\dagger_{m,t} - \tilde{x}^*_{m,t}|^+ \leq \sum_{i=1}^t \sum_{m=1}^M |x^\dagger_{m,t} - \tilde{x}^*_{m,t}|^+ \leq \sum_{i=1}^t \|x^\dagger_t - \tilde{x}^*_t\|_1, \end{aligned} \tag{44}$$

where the second inequality holds by 1-Lipschitz of $\min\{\cdot, B_{\max}\}$, and the second inequality holds by iteratively applying the first inequality. Thus, the sufficient condition for the robust constraint (40) becomes

$$2\gamma\lambda L \sum_{i=1}^t \|\tilde{x}^*_i - x^\dagger_i\|_1 \leq (1-\lambda)\sum_{i=1}^t f_i(x^\circ_i) + R. \tag{45}$$

Since $(1-\lambda)\sum_{i=1}^{t} f_i(x_i^\circ) \geq 0$, if $\gamma \in [0,1]$ satisfies

$$\gamma \leq \min\left\{1, \frac{R}{2\lambda L \sum_{i=1}^{t} \|\tilde{x}_i^* - x_i^\dagger\|_1}\right\}, \tag{46}$$

then x_t° satisfies the robust constraint (40) for each round t.

Thus, by the definition of $\theta = \max_y \sum_{t=1}^{T} \|\tilde{x}_t^* - x_t^\dagger\|_1$, we further have the sufficient condition that $\hat{x}_t$ satisfies the robust constraint is $\gamma \in [0,1]$ satisfies

$$\gamma \leq \min\left\{1, \frac{R}{2\lambda L \theta}\right\} := \gamma_{\lambda,R}, \tag{47}$$

Next, we can bound the average utility of $\pi^\circ = \gamma_{\lambda,R}\tilde{\pi}^* + (1-\gamma_{\lambda,R})\pi^\dagger$ which is also the bound of the average utility of the proposed policy. Since the function f is $L-$Lipschitz continuous, then we have

$$\begin{aligned}
\mathbb{E}_y\left[F_T^{\pi^\circ}(y)\right] &= \mathbb{E}_y\left[F_T^{\tilde{\pi}^*}(y)\right] - \mathbb{E}_y\left[\left|F_T^{\pi^\circ}(y) - F_T^{\tilde{\pi}^*}(y)\right|\right] \\
&\geq \mathbb{E}_y\left[F_T^{\tilde{\pi}^*}(y)\right] - L\mathbb{E}_y\left[\sum_{t=1}^{T} \|x_t^\circ - \tilde{x}_t^*\|\right] \\
&= \mathbb{E}_y\left[F_T^{\tilde{\pi}^*}(y)\right] - L(1-\gamma_{\lambda,R})\mathbb{E}_y\left[\sum_{t=1}^{T} \|x_t^\dagger - \tilde{x}_t^*\|\right] \\
&= \mathbb{E}_y\left[F_T^{\tilde{\pi}^*}(y)\right] - L\max\{0, 1-\frac{R}{2\lambda L\theta}\}\mathbb{E}_y\left[\sum_{t=1}^{T} \|x_t^\dagger - \tilde{x}_t^*\|\right],
\end{aligned} \tag{48}$$

where the inequality holds by the Lipschitz continuity of reward functions, and the second equality holds since $x_t^\circ - \tilde{x}_t^* = (1-\gamma_{\lambda,R})(x_t^\dagger - \tilde{x}_t^*)$. Since the ML policy $\tilde{\pi}^\circ$ is optimally trained under the constraint (40), we have $\mathbb{E}_y\left[F_T^{\mathsf{LA\text{-}OACP}(\tilde{\pi}^\circ)}(y)\right] \geq \mathbb{E}_y\left[F_T^{\pi^\circ}(y)\right]$, so we prove the average bound in our theorem. □

Received February 2023; revised January 2024; accepted January 2024