Fairness-Regularized Online Optimization with Switching Costs

Pengfei Li
School of Information
Rochester Institute of Technology
pflics@rit.edu

Yuelin Han
Electrical and Computer Engineering
University of California, Riverside
yhan116@ucr.edu

Adam Wierman
Computing & Mathematical Sciences
California Institute of Technology
adamw@caltech.edu

Shaolei Ren
Electrical and Computer Engineering
University of California, Riverside
shaolei@ucr.edu

Abstract

Fairness and action smoothness are two crucial considerations in many online optimization problems, but they have yet to be addressed simultaneously. In this paper, we study a new and challenging setting of fairness-regularized smoothed online convex optimization with switching costs. First, to highlight the fundamental challenges introduced by the long-term fairness regularizer evaluated based on the entire sequence of actions, we prove that even without switching costs, no online algorithms can possibly achieve a sublinear regret or finite competitive ratio compared to the offline optimal algorithm as the problem episode length T increases. Then, we propose `FairOBD` (Fairness-regularized Online Balanced Descent), which reconciles the tension between minimizing the hitting cost, switching cost, and fairness cost. Concretely, `FairOBD` decomposes the long-term fairness cost into a sequence of online costs by introducing an auxiliary variable and then leverages the auxiliary variable to regularize the online actions for fair outcomes. Based on a new approach to account for switching costs, we prove that `FairOBD` offers a worst-case asymptotic competitive ratio against a novel benchmark—the optimal offline algorithm with parameterized constraints— by considering $T \to \infty$. Finally, we run trace-driven experiments of dynamic computing resource provisioning for socially responsible AI inference to empirically evaluate `FairOBD`, showing that `FairOBD` can effectively reduce the total fairness-regularized cost and better promote fair outcomes compared to existing baseline solutions.

1 Introduction

Smoothed decision-making is critical to reducing abrupt and even potentially dangerous large action changes in many online optimization problems, e.g., energy production scheduling in power grids, object tracking, motion planning, and server capacity provisioning in data centers, among others [47, 55, 2]. Mathematically, action smoothness can be effectively achieved by including into the optimization objective a switching cost that penalizes temporal changes in actions. The added switching cost essentially equips the online optimization objective with a (finite) memory of the previous actions and also creates substantial algorithmic challenges. As such, it has received a huge amount of attention in the last decade, with a quickly growing list of algorithms developed under various settings (see [50, 3, 32, 45, 30, 55, 47, 13, 34, 11] and the references therein).

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Additionally, (long-term) fairness is a crucial consideration that must be carefully addressed in a variety of online optimization problems, especially those that can profoundly impact individuals' well-being and social welfare. For example, sequential allocation of social goods to different individuals/groups must be fair without discrimination [6, 49], the surging environmental costs of geographically distributed AI data centers must be fairly distributed across different regions as mirrored in recent repeated calls by various organizations and government agencies [22, 10, 28, 23], and AI model performance and resource provisioning must promote social fairness without adversely impacting certain disadvantaged individuals/groups [43]. If not designed with fairness in mind, online algorithms can further perpetuate or even exacerbate societal biases, which can disproportionately affect certain groups/individuals and amplify the already widening socioeconomic disparities. Therefore, fairness in online optimization is not just a matter of ethical concern but also a necessity for ensuring broader social responsibility.

The incorporation of a long-term fairness regularizer into the objective function can effectively achieve fairness in online optimization by mitigating implicit or explicit biases that could otherwise be reinforced [6, 40, 39, 28]. Despite its effectiveness in promoting fairness, the added fairness regularizer presents significant challenges for online optimization[6, 40]. This arises from the lack of complete future information, which is required to evaluate the fairness regularizer. Intuitively, online decision-makers prioritize immediate cost reduction, reacting to sequentially revealed information, potentially neglecting the long-term consequences of the decision sequence.

Recent research has started to explore fairness in various online problems, such as online resource allocation, using primal-dual techniques [39, 6, 35]. However, a major limitation of current fair online algorithms lies in their reliance on the assumption that per-round objective functions are independent of historical actions. This assumption plays a pivotal role in both algorithm design and theoretical analysis (e.g., [6, 7]). For instance, in a simple resource allocation scenario, while the maximum allowable action at time t and its corresponding reward may depend on past allocation decisions through the remaining budget, the revenue or reward function at time t is assumed to depend only on the current allocation decision and remain independent of these historical decisions. However, these methods fall short in addressing online problems where the per-round objective function is directly impacted by historical actions, such as those involving switching costs, as detailed in Section 2.

Contributions. In this paper, we study fairness-regularized smoothed online optimization with switching costs, a new and challenging setting that has not been considered in the literature to our knowledge. More specifically, we incorporate a fairness regularizer that captures the long-term impacts of online actions, and aim to minimize the sum of a per-round hitting cost, switching cost, and a long-term fairness cost. To simultaneously address the tension between minimizing the hitting cost, switching cost, and fairness cost, we propose `FairOBD` (Fairness-regularized Online Balanced Descent).

Optimizing the long-term cost with a sequentially revealed context sequence is widely recognized as a challenging problem (e.g., [39, 6, 7, 12, 38, 26]). A common strategy to address this challenge involves decomposing the long-term fairness cost into a sequence of online costs by introducing an auxiliary variable (e.g. [8, 41, 35], which effectively converts the fairness cost into an equivalent long-term constraint. However, the key limitation of existing methods is the requirement of independence between per-round objective functions, which is violated by the switching cost. Specifically, given the updated Lagrangian multiplier, the per-round online objective depends on the previous *irrevocable* action due to the switching cost. This dependency is not permitted in prior fairness-regularized online problems [6, 39]. As a result, due to the differing action sequences, both `FairOBD` and the benchmark algorithm (R, δ)-OPT (Definition 4.3) have distinct objective functions at each round, complicating a direct comparison of their costs.

In this work, we rigorously analyze the inherent hardness of incorporating long-term fairness regularization into online optimization. We establish fundamental lower bounds in the dynamic benchmark under two standard performance metrics: regret and competitive ratio. Unlike the static benchmark, where the offline optimal action is constrained to a single fixed action, the dynamic benchmark allows the offline optimal action sequence to adapt over time with full knowledge of the context. While a finite competitive ratio is attainable in online optimization with switching costs, whether a similar finite competitive ratio can be achieved in presence of long-term fairness regularization remains an open question. Intuitively, this is because long-term regularizer depend on the entire trajectory and can only be evaluated at the end of an episode, whereas switching costs can be assessed step by step.

Although prior work has made conjectures about the impossibility of achieving finite performance in similar settings [6], our results provide the first rigorous proof. Our results show that in the adversarial setting, all online algorithms inevitably incur a constant regret gap (leading to linear total regret) and an unbounded competitive ratio that grows at least linearly with the horizon length. This demonstrates that the unconstrained dynamic benchmark is unsuitable for analyzing such problems. Motivated by real-world applications and guided by insights from our proofs, we propose the (R, δ)-benchmark, in which the deviation between piece-wise and episode-wise function averages is bounded.

By developing a novel proof technique that explicitly accounts for the dependency of online objectives on previous actions, we show that `FairOBD` offers a worst-case asymptotic competitive ratio against a (R, δ)-constrained optimal offline algorithm (Theorem 5.2 in Section 5.2), where R and δ specify the total allowed variations of the optimal actions. Importantly, our analysis quantifies the impact of the long-term fairness regularizer on the convergence speed of `FairOBD`'s total cost, which asymptotically matches the best-known results for two existing settings – smoothed online optimization without fairness and fairness-regularized online optimization without switching costs—as special cases. This demonstrates the power and generality of `FairOBD`.

Finally, to empirically evaluate the performance of `FairOBD`, we conduct trace-driven experiments for AI workload shifting with a focus on geographically balanced distribution of public health risks. Our results demonstrate that `FairOBD` can effectively reduce the total fairness-regularized cost, highlighting the empirical advantages of `FairOBD` compared to existing baseline solutions.

In summary, our impossibility results and the proposed (R, δ) benchmark lay a solid foundation for studying online optimization with long-term fairness. We further advance the literature on smoothed online optimization by introducing a novel fairness regularizer that promotes equitable outcomes. Most importantly, we overcome the key technical challenges of entangling switching costs as a type of action memory into the mirror descent update process, which is not allowed by the prior fairness-regularized online algorithms [6, 39] or smoothed online algorithms [18].

Together,

2 Related Works

Smoothed online optimization. Smoothed online optimization is a challenging problem for which a growing list of algorithms have been designed to offer the worst-case performance guarantees in both adversarial and stochastic settings [13, 47, 20, 55, 11]. Recently, it has also been extended in various directions, including decentralized networked systems [33], hitting cost feedback delays [42], bandit cost feedback [2, 50, 14], switching cost constraints [46, 51], future cost predictions [31, 30], and machine learning advice augmentation [13, 3, 44, 27], among others.

By considering a dynamic (constrained) offline optimal algorithm as the benchmark, our work focuses on the worst-case competitive ratio and considers the classic smoothed online convex optimization setting [18], but adds a long-term horizon fairness regularizer. This addition presents significant challenges to our algorithm design and is the key novelty that separates our work from the rich literature on smoothed online optimization. Another study [25] considers smoothed online conversion with a simple metric switching cost and a long-term constraint $\sum_t^T x_t = 1$ where $x_t \in \mathbb{R}^+$ is the scalar action at time t (irrelevant to fairness). Besides different hitting and switching costs, our work is substantially separated from [25] as we consider a general and challenging fairness cost that regularizes the long-term fairness of online actions and necessitates our new algorithm `FairOBD`.

Online optimization with fairness and long-term constraints. Our work is broadly relevant to online optimization with fairness and long-term constraints [49, 36, 40]. One key idea is to properly choose the Lagrangian multipliers that correspond to long-term constraints or fairness. For example, some earlier works [15, 17] study online allocation problems by estimating a fixed Lagrangian multiplier using offline data, while other works design online algorithms by updating (or learning to predict) the Lagrangian multiplier in an online manner [16, 1, 56, 4, 54]. More recently, reinforcement learning has been applied to solve online problems with long-term constraints [24], but it may not provide any worst-case performance guarantees in adversarial settings.

Importantly, some recent studies [53, 5, 7, 6] investigate online allocation problems in which the agent first fully observes the reward or cost function before selecting an action. In contrast, other recent work considers settings where certain components of the reward function are not revealed in advance

[38, 12], providing only bandit feedback or requiring additional cost for revelation. Such limitations preclude direct optimization of the reward via gradient-based methods. Despite these differences, both types of settings can be addressed using dual mirror descent (DMD) or its variants, which update the Lagrangian multipliers based on the available context and the observed action sequence. DMD can recover a family of classic online algorithms, including projected gradient descent [56, 40] and multiplicative weight [4]. Similar algorithms are proposed for online stochastic optimization with distribution information in [21]. While fairness can be converted into long-term constraints with the introduction of an auxiliary variable, it also substantially alters the algorithm designs and is known to be difficult to address, as exemplified in the context of online budget allocation [6]. More crucially, given the updated dual variable at each round, the objective function in these studies is *independent* of the previous actions, but our objective function naturally has a memory due to switching costs and hence requires a novel design of `FairOBD`.

3 Problem Formulation

Consider a smoothed online optimization problem spanning an episode of T time rounds denoted by $[T] = \{1, \cdots, T\}$. At each time $t \in [T]$, the agent observes a (potentially adversarially chosen) cost function $f_t(\cdot) : \mathbb{R}^N \to \mathbb{R}^+$ and a matrix $A_t \in \mathcal{A} \subset \mathbb{R}_+^{M \times N}$, and chooses an irrevocable action $x_t \in \mathcal{X} \subset \mathbb{R}^N$ without knowing the future information. The agent incurs two costs at time t: a hitting cost $f_t(x_t)$ and a switching cost $d(x_t, x_{t-1})$, which captures how well the current cost $f_t(\cdot)$ and penalizes temporal changes in actions, respectively. For notational convenience, we define $\mathcal{Z} = \{Ax | A \in \mathcal{A}, x \in \mathcal{X}\}$. At each round t, the matrix A_t maps the agent's action x_t into an M-dimension cost vector, with each dimension corresponding to an entity for which *fairness* needs to be taken into account. Thus, to regularize the online actions x_t for fairness, we define a long-term horizon fairness cost $g(\cdot) : \mathbb{R}^M \to \mathbb{R}^+$ in terms of $\frac{1}{T}\sum_{t=1}^{T} A_t x_t$.

The overall cost is denoted by $cost(x_{1:T})$ and written as

$$cost(x_{1:T}) = \frac{1}{T}\Big(\sum_{t=1}^{T} f_t(x_t) + \sum_{t=1}^{T} d(x_t, x_{t-1})\Big) + g\Big(\frac{1}{T}\sum_{t=1}^{T} A_t x_t\Big) \tag{1}$$

3.1 Assumptions

We make the following standard assumptions as in the prior literature [18, 40, 47, 55].

Assumption 3.1. For each time $t \in [T]$, the hitting cost $f_t(x_t)$, switching cost $d(x_t, x_{t-1})$ and long-term fairness cost $g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t)$ satisfy the following properties:

- The hitting cost $f_t(x_t)$ is continuous and non-negative within the action set X.
- The switching cost $d(x_t, x_{t-1})$ is the scaled squared L_2-norm, i.e., $d(x_t, x_{t-1}) = \frac{\beta_1}{2}\|x_t - x_{t-1}\|^2$, where x_0 is the initial action before the start of the episode.
- The diameter of the vector set $\mathcal{Z} = \{Ax | A \in \mathcal{A}, x \in \mathcal{X}\}$ is bounded by Z, i.e., $\sup_{x_t, x_t' \in \mathcal{X}, A_t, A_t' \in \mathcal{A}} \|A_t x_t - A_t' x_t'\| \leq Z$, where $\|\cdot\|$ is the l_2-norm unless otherwise noted.
- The fairness cost $g(y)$ is convex and L-Lipschitz, i.e., $g(y) - g(y') \leq L \cdot \|y - y'\|$ for any $y, y' \in \mathcal{Z}$.

These assumptions, e.g., non-negativity, are needed for theoretical analysis and widely considered in the literature [18, 39]. Additionally, it is common to consider a convex L-Lipschitz fairness cost. For example, a well-known fairness cost that prioritizes the minimization of higher costs for disadvantaged entities is the l_p norm, i.e., $g(y) = \|y\|_p$ for $p \geq 1$, as considered in a variety of applications including fair federated learning [29], budget allocation [6], server scheduling [9], and geographical load balancing [28], among many others. As a special case when $p \to \infty$, the l_∞-norm fairness cost addresses the classic minimax fairness.

In numerous online problems, including energy scheduling in power grids, object tracking in robotic-human interactions, server capacity provisioning in data centers, and more in [47, 55, 13], both long-term cost and smoothness are crucial considerations. To better motivate the consideration and make our model more concrete, we provide several application examples in Appendix A.

4 The Hardness of Long-term Fairness Regularizer

Compared with switching cost which can be calculated with the action in the subsequent step, the long-term regularizer can be only evaluated at the end of the episode of T, with the complete action and context sequence. Thus, even intuitively, handling the long-term regularizer is considerably more challenging than managing switching costs. In this subsection, we will provide rigorous analysis on the impact of this inherent difficulty on two commonly considered performance metrics: regret and competitive ratio.

Our theoretical results establish fundamental lower bounds on the performance gap between online and offline algorithms in the presence of such long-term dependencies, highlighting the hardness of the long-term fairness regularizer for online optimization even (in the absence of the switching cost).

Theorem 4.1. *Assume that there is no switching cost (i.e., $\beta_1 = 0$). Let $x^*_{1:T}$ denote the optimal offline action sequence. For any online algorithm that produces an online sequence of $x^\dagger_{1:T}$, the worst-case regret, defined as $\max_{(f_{1:T}, A_{1:T})} \Big[cost(x^\dagger_{1:T}) - cost(x^*_{1:T}) \Big]$, is lower-bounded by a positive constant, i.e. $\Omega(1)$.*

The proof idea of Theorem 4.1 is to construct adversarial future context sequence based on the historical actions of the online algorithm, which can be found in Appendix C.1. Notably, as the hitting and long-term costs are already averaged by the length of episode T, the constant performance gap in Theorem 4.1 can be translated to a linear regret when evaluating the total cost. Furthermore, with the switching cost eliminated ($\beta = 0$), this linear regret is solely attributed to the long-term cost. The lower bound on the cost gap depends on the size of the action and context spaces, specifically quantified by the maximum norms of the action and context matrices (their diameters). Our proof establishes this lower bound by constructing an adversarial context sequence and identifying an offline optimal solution, both within a unit l_2 ball. If the diameters of these spaces further increase, the lower bound on the regret would also increase accordingly through a simple scaling of the adversarial sequence. Thus, if this diameter grows with the episode length T, the cost gap will similarly increase with T, potentially leading to superlinear regret in terms of the total cost.

To conclude, the findings in Theorem 4.1 highlight the fundamental limitation imposed by the lack of future context information, preventing any online algorithm from asymptotically matching the offline optimal cost.

Theorem 4.2. *Assume that there is no switching cost (i.e., $\beta_1 = 0$). Let $x^*_{1:T}$ denote the optimal offline action sequence. For any online algorithm that produces an online sequence of $x^\dagger_{1:T}$, the competitive ratio, defined as $C^\dagger = \max_{(f_{1:T}, A_{1:T})} \frac{cost(x^\dagger_{1:T})}{cost(x^*_{1:T})}$, is lower-bounded by $\Omega(T)$, i.e., $C^\dagger \gtrsim \Omega(T)$.*

In addition to regret, the competitive ratio is a widely used metric for evaluating online algorithms, particularly against a dynamic benchmark. Unlike a static benchmark, the dynamic setting allows the offline optimal action to vary over time, potentially achieving significantly lower costs. In such dynamic scenarios, achieving sublinear regret may be impossible, making the competitive ratio a more suitable performance measure. A finite competitive ratio guarantees that an online algorithm's cost is bounded by a constant multiple of the offline optimal cost. For problems with only switching and hitting costs, [19] achieved the state-of-the-art finite competitive ratio. Given that our problem setting also employs a similar dynamic benchmark, competitive ratio appears to be an appropriate evaluation metric. However, the inclusion of a long-term regularizer fundamentally alters this landscape.

The inherent difficulty introduced by such regularization is a long-standing research question. For example, [6] considers a related online optimization problem focused on maximizing total reward under constraints on both the context sequence and the offline optimal benchmark. They conjecture that no online algorithm could achieve vanishing regret or a finite competitive ratio. Based on our Theorem 4.2 and 4.1, we rigorously confirm that in this setting, neither a finite competitive ratio nor vanishing regret is attainable, even in the absence of switching costs. These theorems highlight the fundamental challenge introduced by the long-term regularizer. To enable meaningful performance comparisons despite this challenge, we slightly constrain the offline optimal benchmark and establish an asymptotic competitive ratio under this modified benchmark.

Definition 4.3 ((R, δ)-optimal algorithm)**.** An action sequence $x^*_{1:T}$ is (R, δ)-optimal if it satisfies:

$$x^*_{1:T} = \arg\min_{x_{1:T}} cost(x_{1:T}), \quad s.t., \quad \sum_{k=1}^{K} \left\| \sum_{t=(k-1)R+1}^{kR} A_t x^*_t - \frac{R}{T} \sum_{t=1}^{T} A_t x^*_t \right\| \leq \delta, \tag{2}$$

where R represents the frame size and K is a positive integer denoting the number of frames in an episode such that $R \cdot K = T$, and $\delta \geq 0$ measures the allowable total deviation between the frame-wise $\sum_{t=(k-1)R+1}^{kR} A_t x^*_t$ and episode-wise average $\frac{R}{T} \sum_{t=1}^{T} A_t x^*_t$ scaled by R.

The constrained benchmark, denoted (R, δ)-OPT, imposes a restriction on the offline optimal solution or the adversarial context sequence. Specifically, the parameter δ limits the total deviation between the frame-wise average fairness vector, $A_t x^*_t$, and its episode-wise counterpart, scaled by the frame length R. The stringency of this (R, δ) constraint is inversely related to the values of R and δ; increasing either parameter makes the constraint less restrictive, and vice versa. This benchmark effectively limits the power of the offline optimum by assuming its total variance is sublinear with respect to the episode length T. If the total variance of the offline optimal action sequence were permitted to grow linearly with T, the environment can continuously shift the average of the optimal long-term cost. In other words, the offline optimal long-term cost is able to maintain a constant distance from the online algorithm's cost, thus explaining the constant cost gap observed in Theorem 4.1. Therefore, we assume δ in the offline optimal benchmark is always sublinear with respect to the episode length T. Moreover, similar constrained offline optimal benchmarks have also been widely considered in the literature, such as the switching-constrained OPT in smoothed online optimization [18] and frame-wise OPT with limited future information [40, 39]. Notably, in many practical applications such as data center scheduling, this is a very natural assumption, where the context A_t exhibits some periodicity as evidenced by the diurnal workloads. In this case even the actions chosen by the unconstrained offline algorithm can exhibit a periodic pattern, thus resulting in a sufficiently small δ for a certain large R.

In the following, we interchangeably refer to (R, δ)-OPT as OPT whenever applicable without ambiguity. Next, we define the asymptotic competitive ratio [7] by considering $T \to \infty$ as follows.

Definition 4.4 (Asymptotic competitive ratio)**.** Given an offline (R, δ)-optimal algorithm OPT, an algorithm ALG is asymptotically C-competitive against OPT if for any problem instance $\gamma = \{f_t, A_t | t = 1, \cdots, T\}$, it satisfies $\sup_\gamma \lim_{T\to\infty} \Big[cost(ALG) - C \cdot cost(OPT) \Big] \leq 0$.

5 FairOBD: Fairness-Regularized Online Balanced Descent

We propose a novel online algorithm FairOBD for fairness-regularized smoothed online optimization defined in Eqn. (1). Compared to the existing smoothed online optimization literature without fairness consideration [47, 18, 42, 27], the crux of FairOBD is to decompose the horizon fairness cost $g\big(\frac{1}{T}\sum_{t=1}^{T} A_t x_t\big)$ into online costs by introducing a new regularizer and an auxiliary variable that carefully balances the hitting cost, switching cost, and fairness cost.

5.1 Algorithm Design

Even in the absence of the fairness cost, the problem of smoothed online optimization is already challenging due to the potential conflicts between minimizing the current hitting cost $f_t(x_t)$ and staying closer to the previous action to reduce the switching cost $d(x_t, x_{t-1})$. The fairness cost $g\big(\frac{1}{T}\sum_{t=1}^{T} A_t x_t\big)$ is defined in terms of the long-term average of $\frac{1}{T}\sum_{t=1}^{T} A_t x_t$ and hence cannot be determined until the end of time T, further creating substantial challenges. FairOBD addresses these challenges by introducing an auxiliary variable and decomposing $g\big(\frac{1}{T}\sum_{t=1}^{T} A_t x_t\big)$ into an online version, and by balancing the hitting cost, switching cost, and fairness cost.

5.1.1 Decomposing the fairness cost

To address the challenges stemming from the horizon fairness cost $g\big(\frac{1}{T}\sum_{t=1}^{T} A_t x_t\big)$, we introduce an auxiliary variable $z_t \in \mathcal{Z} = \{z = Ax | A \in \mathcal{A}, x \in \mathcal{X}\}$. In the long run, we aim to choose the actions

Algorithm 1 `FairOBD`: Fairness-regularized Online Balanced Descent

1: **Input:** Initial κ_1, reference function $h(\cdot)$, and learning rate η
2: **for** $t = 1$ **to** T **do**
3: Receive $f_t(\cdot)$ and A_t
4: Obtain the action x_t and auxiliary variable z_t by solving the following:

$$\min_{x_t \in \mathcal{X}_t, z_t \in \mathcal{Z}} \Big(f_t(x_t) + \lambda_1 d(x_t, x_{t-1}) + \frac{\lambda_2}{2} \|x_t - v_t\|^2 + \kappa_t \cdot A_t x_t + g(z_t) - \kappa_t z_t \Big), \tag{5}$$

where $v_t = \arg\min_{x \in \mathcal{X}} f_t(x)$.
5: Obtain a stochastic subgradient of κ_t: $d_t = z_t - A_t \cdot x_t$
6: Update the dual variable by mirror descent: $\kappa_{t+1} = \arg\min_{\kappa \in \mathbb{R}^M} \langle d_t, \kappa \rangle + \frac{1}{\eta} V_h(\kappa, \kappa_t)$, where $V_h(x, y) = h(x) - h(y) - \nabla h(y)^\top (x - y)$ is the Bregman divergence.
7: **end for**

that *approximately* satisfy the condition $\sum_{t=1}^T A_t x_t = \sum_{t=1}^T z_t$. This ensures that the horizon-wise average of the auxiliary variable can serve as an effective estimator for the horizon fairness cost. Alternatively, we can view z_t as a dynamic "budget" that is allocated to the agent to guide its online actions. Thus, we reformulate the problem (1) as follows:

$$\min_{x_{1:T}, z_{1:T}} \frac{1}{T} \left(\sum_{t=1}^T f_t(x_t) + \sum_{t=1}^T d(x_t, x_{t-1}) \right) + \frac{1}{T} \sum_{t=1}^T g(z_t), \quad s.t. \quad \sum_{t=1}^T A_t x_t = \sum_{t=1}^T z_t, \tag{3}$$

whose optimal actions are denoted by $(\hat{x}_{1:T}, \hat{z}_{1:T})$. Importantly, the offline optimal cost of (3) is the same as that of the original problem in (1). This point can be seen by considering $\hat{z}_1 = \cdots = \hat{z}_T = \frac{1}{T} \sum_{t=1}^T A_t \hat{x}_t$ and $\hat{x}_{1:T} = x^*_{1:T}$ where $x^*_{1:T}$ is the offline optimal solution to (1).

While the reformulated problem (3) decomposes the horizon fairness cost $g\big(\frac{1}{T} \sum_{t=1}^T A_t x_t\big)$ by the introduction of $z_{1:T}$ and has the same optimal cost as (1), it presents a new challenge due to the long-term constraint $\sum_{t=1}^T A_t x_t = \sum_{t=1}^T z_t$ where z_t itself is also an online action. To address this challenge, we note that unlike a real budget constraint that cannot be violated at any round [7], the long-term constraint in (3) only needs to be approximately satisfied. Therefore, we relax the constraint and instead minimize the Lagrangian form of (3) as follows:

$$\min_{x_{1:T}, z_{1:T}} \frac{1}{T} \left(\sum_{t=1}^T f_t(x_t) + \sum_{t=1}^T d(x_t, x_{t-1}) \right) + \frac{1}{T} \sum_{t=1}^T g\left(z_t\right) + \kappa_t \left(\sum_{t=1}^T A_t x_t - \sum_{t=1}^T z_t \right) \tag{4}$$

where $\kappa_t \in \mathbb{R}^M$ is the estimated Lagrangian multiplier at round $t \in [T]$.

Had all the future information $\{f_{1:T}(\cdot), A_{1:T}\}$ been provided in advance, we could easily determine the optimal $\kappa_1 = \cdots = \kappa_T = \kappa^*$. But, for online optimization, we need to update z_t based on the currently available information without knowing the future. Moreover, interestingly, given κ_t at round t, x_t and z_t can be independently solved in (4). Thus, we propose to employ mirror descent to update κ_t online, based on which we subsequently optimize x_t and z_t at each round $t \in [T]$.

While [7, 6, 40] use mirror descent to update the Lagrangian multiplier for online allocation of a fixed given budget, our work has a crucial difference: given κ_t at each round t, our objective includes *memory* in the form of a switching cost $d(x_t, x_{t-1})$, whereas the objective in [7, 6, 40] only depends on the current action x_t. This difference voids the key steps in the proof techniques of [7, 6, 40] and requires additional care to the switching cost.

5.1.2 Balancing the hitting cost, switching cost, and fairness cost

To further reconcile conflicts between minimizing the current hitting cost $f_t(x_t)$ and staying closer to the previous action to reduce the switching cost $d(x_t, x_{t-1})$, we propose `FairOBD` that uses the term $\kappa_t A_t x_t$ as a fairness regularizer while optimizing x_t online. The fairness regularization effect of $\kappa_t A_t x_t$ comes from the fact that it is intended to meet the long-term constraint $\sum_{t=1}^T A_t x_t = \sum_{t=1}^T z_t$ for fairness, while "balanced" is due to two new hyperparameters $\lambda_1 \in (0, 1]$ and $\lambda_2 \in [0, \infty)$ to balance the hitting cost $f_t(x_t)$ and switching cost $d(x_t, x_{t-1})$. The hyperparameters $\lambda_1 \in (0, 1]$

and $\lambda_2 \in [0, \infty)$ can be optimally set to optimize the asymptotic competitive ratio in Theorem 5.2. Furthermore, given κ_t, the auxiliary variable z_t can be easily optimized by solving $g(z_t) - \kappa_t z_t$.

Before completing the design of `FairOBD`, we introduce a reference function $h(\cdot) : \mathbb{R}^M \to \mathbb{R}$ to define the Bregman divergence in our update of κ_t using mirror descent. Specifically, we choose $h(\cdot) : \mathbb{R}^M \to \mathbb{R}$ that is l-strongly convex and β_2-smooth [48]. For example, Line 7 of Algorithm 1 follows the classic additive update rule $\kappa_{t+1} = [\kappa_t - \eta d_t]^+$ where $\eta > 0$ is the learning rate when $h(\kappa) = \frac{\|\kappa\|^2}{2}$, whereas it becomes the multiplicative update when $h(\kappa) = \kappa^T \log(\kappa)$ [7]. The specific choice $h(\cdot)$ does not affect our performance analysis in Theorem 5.1 and 5.2.

Next, we formally describe `FairOBD` in Algorithm 1. In Line 4, we optimize x_t to balance the hitting cost, switching cost, and fairness cost. In Line 5, we optimize z_t as a fairness budget for guiding the optimization of x_t in the future. In Lines 6 and 7, we update κ_{t+1} using mirror descent.

5.2 Performance Analysis

We present the analysis of `FairOBD` in terms of the asymptotic competitive ratio as follows.

Theorem 5.1. *Suppose that $x^*_{1:T}$ is the action produced by the (R, δ)-optimal algorithm OPT. Assume that there is no switching cost (i.e., $\beta_1 = 0$). By setting the learning rate as $\eta = \mathcal{O}(\frac{1}{\sqrt{T}})$ and parameters $\lambda_1 = \lambda_2 = 0$, and initializing $\kappa_1 = \mathcal{O}(\frac{1}{T})$ (or $\kappa_1 = 0$ if $h(\kappa) = \frac{\|\kappa\|^2}{2}$), the cost of `FairOBD` is upper bounded by*

$$\begin{aligned} cost(x_{1:T}) \leq & cost(x^*_{1:T}) + \left[\frac{\eta}{2l} Z^2 R + Z\|\kappa_1\| + \beta_2 L \sqrt{\frac{1}{T}(\frac{Z^2}{l^2} + 2\frac{LZ}{\eta l} + \frac{2}{\eta^2 l} Z\|\kappa_1\|)}\right] + \frac{L\delta}{T} \\ = & cost(x^*_{1:T}) + \mathcal{O}(\frac{1}{\sqrt{T}}) + \frac{L\delta}{T}, \end{aligned} \tag{6}$$

where the parameters m, Z and L are specified in Assumption 3.1, and l and β_2 are the strong convexity and smoothness parameters of the reference function $h(\cdot)$, respectively.

In Theorem 5.1, `FairOBD` achieves an asymptotic competitive ratio of 1 when the total deviation δ is sublinear in T. The total deviation δ quantifies the fundamental difficulty of the online problem itself, independent of the algorithm design. As discussed regarding the hardness of the long-term regularizer, if δ grows linearly with T, no online algorithm can attain sublinear regret. Therefore, Theorem 5.1 demonstrates that `FairOBD` achieves vanishing regret as the episode length approaches infinity, even in worst-case scenarios. This vanishing regret highlights the effectiveness of `FairOBD` in addressing the long-term regularizer, which is achieved by decomposing the long-term cost with a set of auxiliary variables. Furthermore, the dual variable within `FairOBD` dynamically adapts through continuous online updates, which drives the cost of `FairOBD` to eventually converge to the offline optimal benchmark.

Theorem 5.2. *Suppose that $x^*_{1:T}$ is the action produced by the (R, δ)-optimal algorithm OPT and that the hyperparameters $\lambda_1 \in (0, 1]$ and $\lambda_2 \in [0, \infty)$. Assume further the hitting cost $f_t(\cdot)$ is m-strongly convex for any $t = 1, \cdots, T$. By initializing $\kappa_1 = \mathcal{O}(\frac{1}{T})$ (or $\kappa_1 = 0$ if $h(\kappa) = \frac{\|\kappa\|^2}{2}$) and setting the learning rate $\eta = \mathcal{O}(\frac{1}{\sqrt{T}})$, the cost of `FairOBD` is upper bounded by*

$$\begin{aligned} cost(x_{1:T}) \leq & C \cdot cost(x^*_{1:T}) + \frac{1}{\lambda_1}\left[\frac{\eta}{2l} Z^2 R + Z\|\kappa_1\| + \beta_2 L \sqrt{\frac{1}{T}(\frac{Z^2}{l^2} + 2\frac{LZ}{\eta l} + \frac{2}{\eta^2 l} Z\|\kappa_1\|)} + \frac{L\delta}{T}\right] \\ = & C \cdot cost(x^*_{1:T}) + \mathcal{O}(\frac{1}{\sqrt{T}}) + \frac{L\delta}{\lambda_1 T}, \end{aligned} \tag{7}$$

where $C = \max\{\frac{m+\lambda_2}{m\lambda_1}, 1 + \frac{\lambda_1\beta_1}{m+\lambda_2}\}$. The parameters m, Z and L are specified in Assumption 3.1, and l and β_2 are the strong convexity and smoothness parameters of the reference function $h(\cdot)$, respectively. In addition, by considering $\delta = \mathcal{O}(\sqrt{T})$ (or in general sublinear forms in T) and

optimally setting $\lambda_1 \in (0, 1]$ *and* $\lambda_2 \in [0, \infty)$ *such that* $m + \lambda_2 = \frac{\lambda_1 m}{2}\left(1 + \sqrt{1 + \frac{4\beta_1}{m}}\right)$, `FairOBD` *achieves an asymptotic competitive ratio of* $\frac{1}{2}\left(1 + \sqrt{1 + \frac{4\beta_1}{m}}\right)$ *against* (R, δ)*-OPT when* $T \to \infty$.

As demonstrated in Equation (7), by properly setting the learning rate and initial Lagrange multiplier, `FairOBD`'s cost asymptotically converges to within a constant factor of the (R, δ)-constrained offline optimal cost. This convergence guarantees the stability of both the auxiliary variable and multiplier, ensuring robust performance irrespective of the context sequence. In our algorithm, the dual update is conducted using mirror descent, which generalizes the standard gradient descent as a special case when the reference function is chosen as $h(x) = \frac{1}{2}\|x\|^2$. Consequently, Theorem 5.2 and Theorem 5.1 also cover the performance analysis of dual updates with gradient descent by setting $l = \beta_2 = 1$. In addition, a comparison of these two theorems reveals that the constant C in Theorem 5.2 arises solely from the inclusion of the switching cost. Furthermore, by optimally setting (λ_1, λ_2) to minimize C in Equation (7), `FairOBD` asymptotically achieves the competitive ratio of $\frac{1}{2}\left(1 + \sqrt{1 + \frac{4\beta_1}{m}}\right)$. This ratio matches the best-known competitive ratio lower bound [18] for the problem without a fairness cost. This observation highlights the superior performance of `FairOBD` and confirms that C is inevitable due to the switching cost.

Unlike standard online smoothed optimization [19], the long-term fairness cost impacts the total cost of `FairOBD` in two-fold, primarily due to the L-Lipschitz property of the long-term fairness constraint. Generally speaking, larger L implies faster change speed of fairness function, which introduces additional uncertainties when optimizing the auxiliary variables in $g(z_t)$. This imprecise estimation slows down the convergence speed of the dual variable, as reflected in the second last term in Eqn (7). Moreover, since δ represents the discrepancy between the frame-wise and episode-wise averages of $A_t x_t$ in the offline optimal benchmark, a larger L amplifies this difference in `FairOBD`'s final fairness cost, as demonstrated by the last term in Equation (7). Thus, the overall performance of `FairOBD` is significantly influenced by the Lipschitz constant L, which bounds the first-order smoothness of the fairness cost.

The additive gap observed in Theorem 5.1 and Theorem 5.2 is directly influenced by the inherent difficulty associated with the long-term cost. This difficulty is captured by R and δ: as R and/or δ increase, the potential for adversarial context sequences in the long-term cost becomes greater, which is a fundamental challenge for any online algorithm. Thus, R and δ quantify the maximum level of adversarialness in the context sequence. However, many real-world applications are less adversarial, exhibiting predictable periodic patterns (such as the diurnal cycles in AI workload distributions). These patterns significantly reduce δ, the long-term deviations. Consequently, for a reasonably large yet finite R, the total deviation between the online performance and the unconstrained offline optimal benchmark can remain constant or grow sublinearly with T.

In addition to the rigorous insights provided by Theorem 5.2, our proof technique also represents a novel technical contribution. Unlike existing proof techniques that typically rely on the assumption of independence between per-round objective functions , our approach does not require such an assumption. Removing the assumption of temporal independence is crucial in our fairness-regularized smoothed online optimization problem, as the switching cost functions inherently violate this assumption by incorporating historical actions. Consequently, comparing the costs of per-round objective functions between `FairOBD` and the offline optimal benchmark becomes highly challenging, as their cost functions are no longer identical under these conditions. To address this limitation, Lemma D.3 and D.4 establish a theoretical foundation for such a comparison by introducing an intermediate benchmark with per-round objective functions aligned with those of `FairOBD`. This intermediate benchmark effectively bridges the gap between the two policies, each associated with distinct sets of cost functions, enabling a meaningful and rigorous comparison. Though the current result relies on the properties of squared l_2-norm in switching cost, it is promising to relax this assumption to accommodate more general strongly convex and smooth functions. Specifically, in Lemma D.3 and Lemma D.4, the results can be directly extended under strong convexity, since the squared l_2-norm is a special case where convexity and smoothness coincide. Furthermore, when converting the intermediate benchmark to the offline optimal benchmark, the gap between the two can also be bounded by the strong convexity of the switching cost. It is interesting future work to study this more general setting. More broadly, the proof techniques in our approach represent a substantial

development in addressing states influenced by past actions, extending beyond the limitations of existing approaches that are restricted to handling memoryless functions.

6 Empirical Results

Our empirical study investigates the problem of fair resource provisioning for AI inference services in geographically separated data centers, where decisions incur instantaneous hitting costs (energy consumption and workload-imbalance penalties), switching costs (reconfiguration overhead), and a long-term fairness cost quantifying regional public-health impacts from electricity generation pollution. `FairOBD` decomposes and dynamically optimizes all three cost components via auxiliary variables and mirror-descent updates to improve cost efficiency, action smoothness, and long-term fairness simultaneously. In our simulation, we use the one-week trace of normalized LLM inference requests from [37], and route demand across seven selected data centers (Arizona, Iowa, Illinois, Texas, Virginia, Washington, Wyoming) with publicly available electricity prices, PUE values, and health-damage rates from WattTime [52]. We compare `FairOBD` against five benchmarks: the offline optimal (`OPT`), offline fairness-only (`FairOPT`), hitting-cost minimizer (`HITMIN`), Regularized Online Balanced Descent (`ROBD`), and Dual Mirror Descent (`DMD`).

We also test `FairOBD` with different learning rates and different weights for the fairness cost to test its robustness under different settings, with more details in Appendix B. Our numerical results (Table 2 in Appendix B) demonstrate that `FairOBD` achieves the lowest total cost among online methods, with the minimal cost gap to the offline optimal. Moreover, `FairOBD` attains the smallest fairness cost among online baselines, while maintaining low switching overhead. These results confirm that explicit incorporation and dynamic optimization of long-term fairness in `FairOBD` yields superior performance across all cost dimensions.

7 Conclusion

We study a novel and challenging problem of fairness-regularized smoothed online convex optimization. Our main contribution is the proposal of `FairOBD` to reconcile the tension between minimizing the hitting cost, switching cost, and fairness cost. Importantly, `FairOBD` offers a guaranteed worst-case asymptotic competitive ratio against (R, δ)-OPT for finite R and sublinearly growing δ by considering the problem episode length $T \to \infty$. Moreover, our analysis can recover the best-known results for two existing settings (i.e., smoothed online optimization without fairness and fairness-regularized online optimization without switching costs) as special cases. Finally, we run trace-driven experiments of dynamic computing resource provisioning for socially responsible AI inference to demonstrate the empirical advantages of `FairOBD` over existing baseline solutions.

8 Limitation and Impact Statements

While simultaneously addressing fairness and action smoothness in online optimization, `FairOBD` is designed based on a set of assumptions, including non-negative hitting costs and squared l_2-norm switching costs. Therefore, it is interesting future work to overcome these limitations by, e.g., relaxing the strong convexity assumption, considering more general forms of switching costs, and/or incorporating potentially untrusted predictions of future information to further reduce the overall cost. Although our work can potentially increase awareness of fairness in online optimization, we do not foresee negative societal impacts or the need for safeguards due to the theoretical nature of our work.

Acknowledgment

Pengfei Li, Yuelin Han, and Shaolei Ren were supported in part by the NSF under grants CNS-2007115 and CCF-2324941. Adam Wierman was supported by NSF grants CCF-2326609, CNS-2146814, CPS-2136197, CNS-2106403, and NGSDI-2105648 as well as funding from the Resnick Sustainability Institute.

References

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

[2] Idan Amir, Guy Azov, Tomer Koren, and Roi Livni. Better best of both worlds bounds for bandits with switching costs. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 15800–15810. Curran Associates, Inc., 2022.

[3] Antonios Antoniadis, Christian Coester, Marek Elias, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. In *ICML*, 2020.

[4] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[5] Kamiar Asgari and Michael J. Neely. Bregman-style online convex optimization with energy harvesting constraints. *Proc. ACM Meas. Anal. Comput. Syst.*, 4(3), nov 2020.

[6] Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. Regularized online allocation problems: Fairness and beyond, 2021.

[7] Santiago R. Balseiro, Haihao Lu, and Vahab Mirrokni. The best of many worlds: Dual mirror descent for online allocation problems. *Operations Research*, 0(0):null, May 2022.

[8] Santiago R Balseiro, Haihao Lu, and Vahab Mirrokni. The best of many worlds: Dual mirror descent for online allocation problems. *Operations Research*, 71(1):101–119, 2023.

[9] N. Bansal and K. Pruhs. Server scheduling in the l_p norm: a rising tide lifts all boat. In *STOC*, 2003.

[10] California Government Operations Agency. Benefits and risks of generative artificial intelligence report. *State of California Report*, November 2023.

[11] Niangjun Chen, Gautam Goel, and Adam Wierman. Smoothed online convex optimization in high dimensions via online balanced descent. In *COLT*, 2018.

[12] Xi Chen, Jiameng Lyu, Yining Wang, and Yuan Zhou. Network revenue management with demand learning and fair resource-consumption balancing. *Production and Operations Management*, 33(2):494–511, 2024.

[13] Nicolas Christianson, Tinashe Handina, and Adam Wierman. Chasing convex bodies and functions with black-box advice. In *COLT*, 2022.

[14] Ofer Dekel, Jian Ding, Tomer Koren, and Yuval Peres. Bandits with switching costs: $t^{2/3}$ regret. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 459–467, New York, NY, USA, 2014. Association for Computing Machinery.

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

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

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

[18] Gautam Goel, Yiheng Lin, Haoyuan Sun, and Adam Wierman. Beyond online balanced descent: an optimal algorithm for smoothed online optimization. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[19] Gautam Goel, Yiheng Lin, Haoyuan Sun, and Adam Wierman. Beyond online balanced descent: An optimal algorithm for smoothed online optimization. *Advances in Neural Information Processing Systems*, 32:1875–1885, 2019.

[20] Gautam Goel and Adam Wierman. An online algorithm for smoothed online convex optimization. *SIGMETRICS Perform. Eval. Rev.*, 47(2):6–8, December 2019.

[21] Jiashuo Jiang, Xiaocheng Li, and Jiawei Zhang. Online stochastic optimization with wasserstein based non-stationarity. *arXiv preprint arXiv:2012.06961*, 2020.

[22] Amba Kak and Sarah Myers West. AI Now 2023 landscape: Confronting tech power. *AI Now Institute*, April 2023.

[23] Joseph B. Keller, Manann Donoghoe, and Andre M. Perry. The U.S. must balance climate justice challenges in the era of artificial intelligence. *Brookings Institution*, January 2024.

[24] Weiwei Kong, Christopher Liaw, Aranyak Mehta, and D. Sivakumar. A new dog learns old tricks: RL finds classic optimization algorithms. In *ICLR*, 2019.

[25] Adam Lechowicz, Nicolas Christianson, Bo Sun, Noman Bashir, Mohammad Hajiesmaili, Adam Wierman, and Prashant Shenoy. Online conversion with switching costs: Robust and learning-augmented algorithms. In *SIGMETRICS*, 2024.

[26] Pengfei Li, Nicolas Christianson, Jianyi Yang, Adam Wierman, and Shaolei Ren. Learning for sustainable online scheduling with competitive fairness guarantees. In *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*, pages 63–79, 2025.

[27] Pengfei Li, Jianyi Yang, Adam Wierman, and Shaolei Ren. Robust learning for smoothed online convex optimization with feedback delay. In *NeurIPS*, 2023.

[28] Pengfei Li, Jianyi Yang, Adam Wierman, and Shaolei Ren. Towards environmentally equitable ai via geographical load balancing. In *e-Energy*, 2024.

[29] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2020.

[30] Yingying Li, Xin Chen, and Na Li. Online optimal control with linear dynamics and predictions: Algorithms and regret analysis. In *NeurIPS*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[31] Yingying Li and Na Li. Leveraging predictions in smoothed online convex optimization via gradient-based algorithms. In *NeurIPS*, volume 33, 2020.

[32] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *INFOCOM*, 2011.

[33] Yiheng Lin, Judy Gan, Guannan Qu, Yash Kanoria, and Adam Wierman. Decentralized online convex optimization in networked systems. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 13356–13393. PMLR, 17–23 Jul 2022.

[34] Yiheng Lin, Gautam Goel, and Adam Wierman. Online optimization with predictions and non-convex losses. *Proc. ACM Meas. Anal. Comput. Syst.*, 4(1), May 2020.

[35] Wanteng Ma, Ying Cao, Danny HK Tsang, and Dong Xia. Optimal regularized online allocation by adaptive re-solving. *Operations Research*, 73(4):2079–2096, 2025.

[36] Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: Online convex optimization with long term constraints. *J. Mach. Learn. Res.*, 13(1):2503–2528, sep 2012.

[37] Microsoft Azure. Azure Public Dataset. `https://github.com/Azure/AzurePublicDataset`, 2024.

[38] Mathieu Molina, Nicolas Gast, Patrick Loiseau, and Vianney Perchet. Trading-off price for data quality to achieve fair online allocation. *Advances in Neural Information Processing Systems*, 36:40096–40139, 2023.

[39] M. J. Neely. Universal scheduling for networks with arbitrary traffic, channels, and mobility, `http://arxiv.org/abs/1001.0960`.

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

[41] Michael J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

[42] Weici Pan, Guanya Shi, Yiheng Lin, and Adam Wierman. Online optimization with feedback delay and nonlinear switching cost. *Proc. ACM Meas. Anal. Comput. Syst.*, 6(1), Feb 2022.

[43] Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Comput. Surv.*, 55(3), feb 2022.

[44] Daan Rutten, Nicolas Christianson, Debankur Mukherjee, and Adam Wierman. Smoothed online optimization with unreliable predictions. *Proc. ACM Meas. Anal. Comput. Syst.*, 7(1), mar 2023.

[45] Daan Rutten and Debankur Mukherjee. Capacity scaling augmented with unreliable machine learning predictions. *SIGMETRICS Perform. Eval. Rev.*, 49(2):24–26, jan 2022.

[46] Uri Sherman and Tomer Koren. Lazy oco: Online convex optimization on a switching budget. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 3972–3988. PMLR, 15–19 Aug 2021.

[47] Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Online optimization with memory and competitive control. In *NeurIPS*, volume 33. Curran Associates, Inc., 2020.

[48] Zhan Shi, Xinhua Zhang, and Yaoliang Yu. Bregman divergence for stochastic variance reduction: Saddle-point and adversarial prediction. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[49] Sean R. Sinclair, Gauri Jain, Siddhartha Banerjee, and Christina Lee Yu. Sequential fair allocation: Achieving the optimal envy-efficiency trade-off curve. *Oper. Res.*, 71(5):1689–1705, sep 2023.

[50] Juaren Steiger, Bin Li, Bo Ji, and Ning Lu. Constrained bandit learning with switching costs for wireless networks. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, pages 1–10, 2023.

[51] Guanghui Wang, Yuanyu Wan, Tianbao Yang, and Lijun Zhang. Online convex optimization with continuous switching constraint. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[52] WattTime. Data signals for WattTime dataset, 2024.

[53] Jianyi Yang, Pengfei Li, Mohammad J. Islam, and Shaolei Ren. Online allocation with replenishable budgets: Worst case and beyond. In *SIGMETRICS*, 2024.

[54] Jianyi Yang and Shaolei Ren. Learning-assisted algorithm unrolling for online optimization with budget constraints. In *AAAI*, 2023.

[55] Lijun Zhang, Wei Jiang, Shiyin Lu, and Tianbao Yang. Revisiting smoothed online learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

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

Appendix

A Real-world Applications with Long-term Costs

In this section, we present several real-world application that emphasizes long-term fairness along with the switching cost. We provide concrete modeling for these applications and demonstrate how these two applications align closely with our problem formulation.

A.1 Geographical Load Balancing with Balanced Public Health Costs

The exponentially growing demand for AI has been driving the recent surge of data centers around the globe, which are notoriously energy-consuming and thus have huge environmental impacts in terms of the local air pollution and carbon emissions (e.g. for fossil-fuel based power plants). The escalating air pollution resulting from AI computing significantly contributes to various health issues, including lung cancer, heart disease, and cardiovascular problems. Thus, the rise in ambient air pollution leads to substantial public health costs, encompassing lost workdays, increased medication usage, and increased hospitalization rates. More critically, the public health costs of data centers are becoming increasingly unevenly distributed across different regions, disproportionately affecting marginalized communities. To mitigate the uneven distribution of public health impacts from AI computation, an effective approach is to leverage fairness-aware geographical load balancing to fairly distribute the environmental costs across different data center locations [28], and, consequently, public health risks across diverse data center locations.

Specifically, let $x_t \in \mathbb{R}_+^N$ be the number of active servers or computing capacity to process incoming workloads in N different data centers at time t. With a provisioning capacity of x_t, there is an operational cost that includes the energy cost and a penalty term for workload imbalance, which can be modeled as the hitting cost $f_t(x_t)$ [32]. When adjusting computing capacities, a switching cost $d(x_t, x_{t-1})$ arises due to server reallocation, workload migration, or wear-and-tear. Additionally, a health cost $A_t x_t$ is incurred across N locations due to air pollution from electricity generation [59], influenced by the local grid's real-time fossil fuel intensity. Further modeling details can be found in Appendix B.1.

A.2 Socially-fair AI Resource Provisioning

Large AI models often have multiple sizes, each with a distinct performance and resource usage tradeoff. For example, the GPT-3 model family has eight different sizes, ranging from the smallest one with 125 million parameters to the largest one with 175 billion parameters [58]. As a result, the amount of computing resources can directly affect the model selection and inference quality during inference, and needs to be carefully determined based on the number of user requests from different regions. As we become increasingly reliant on AI for acquiring knowledge, insufficient resource provisioning can lead to subpar AI model performance and could disproportionately jeopardize the prospects of users/individuals from certain disadvantaged regions. Thus, the computing resource provisioning for users from different regions must be fair in the long run. In addition, there is a switching cost when adjusting the computing resource capacity (e.g., by activating otherwise hibernating servers).

We consider a discrete-time model for the AI server provisioning problem, where $y_t \in \mathbb{R}^M$ represents the user demand from M different regions at time t and $x_t \in [x_{\min}, x_{\max}]$ denotes the amount of provisioned AI servers to meet the demand. The hitting cost $f(x_t, y_t)$ encompasses both the electricity expenses of the AI servers and a penalty term for any dropped workload demand. The switching cost $d(x_t, x_{t-1})$ models the cost incurred by hardware and software reconfiguration, particularly when server provisioning undergoes frequent or substantial changes. At each time slot, we assume the provisioned AI servers are shared by each region in proportion to their user workload demand. To promote fair distribution of AI resources among the M regions within the horizon, we account for each region's dynamic resource share, which varies based on its fluctuating workload demands. Specifically, we employ a min-max fairness function over the total amount of allocated AI resources for the M regions.

B Experiment

In this section, we run experiments of fair computing resource provisioning for AI inference services. Our results demonstrate that `FairOBD` can effectively reduce the total fairness-regularized cost, highlighting the empirical advantages of `FairOBD` compared to existing baseline solutions.

B.1 Simulation Setup

We investigate the challenge of fair resource provisioning for AI data centers that deliver inference services to dynamically routed AI workloads. At each time step, we dynamically route incoming user demand across geographically dispersed AI data centers and provision server/GPU resources accordingly. This provisioning process incurs a hitting cost (e.g., energy cost, carbon emissions, and penalties for imbalanced workload distribution), as well as a switching cost representing the reconfiguration overhead associated with changes in provisioning decisions. Furthermore, electricity consumption contributes to public health risks within the local community due to air pollution generated during electricity production. We incorporate a long-term fairness cost that quantifies the degree of public health impacts on the most affected region across all AI data centers.

B.1.1 Model

We consider a discrete-time model where each round (or time slot) is one hour. As each round has the same duration, we interchangeably use power and energy wherever applicable without ambiguity. To fulfill the demand for AI inference service at time t, denoted as w_t, we need to determine AI server/GPU provisioning action x_t. The resource provisioning decision is defined as $x_t = [x_{1,t}, \cdots, x_{N,t}]$, where $x_{i,t}$ denotes the provisioned server/GPU capacity for data center $i \in [N]$, subject to constraint $x_{i,t} \leq M_i$ where M_i is its maximum capacity. Additionally, to guarantee the optimal quality of experience for users, we consider all the AI workloads are instantaneously served without further delays, where $\sum_{i=1}^{N} x_{i,t} = w_t$. The provisioning decision incurs a hitting cost (e.g., energy cost and penalty for imbalanced workload distribution), and a switching cost due to changing the provisioning decisions. Furthermore, electricity consumption contributes to public health risks within the local community due to air pollution generated during electricity production. We incorporate a long-term fairness cost that quantifies the degree of public health impacts on the most affected region across all AI data centers. We model these three costs as follows.

The hitting cost encompasses the cost of energy consumption and penalties for imbalanced workload scheduling. Specifically, we model server power consumption as a linear function of provisioned computing resources, $q \cdot x_t$. And the non-IT power consumption (e.g., cooling systems) is modeled by multiplying the power usage efficiency (PUE) γ_i over the power consumption at data center $i \in [N]$. Thus, the total electricity expenses of AI computing can be expressed as $\sum_{i=1}^{N} p_{i,t}^{e} \cdot \gamma_i \cdot q \cdot x_{i,t}$, where $p_{i,t}^{e}$ denotes the time-varying electricity price at location i. Additionally, given the fluctuating AI workload, it's highly imperative to achieve a balanced workload distribution, both spatially and temporally. Within time slot t, as the entire AI workload must be fulfilled collaboratively by N data centers, imbalanced workload distribution can overload several data centers and result in low utilization rate for some data centers. The low utilization may leads to wasted reserved capacity and potentially increased idle power consumption. Additionally, from the perspective of whole horizon T, a smaller temporal variation in the provisioned server capacities is also preferred, considering the same amount of total workload. To this end, we incorporate a regularizer $u_1 \|x_{i,t}\|^2$ into the hitting cost of data center i. When it's summed over the entire planning horizon T, this term represents the squared L_2-norm of the provisioned server capacity vector, $[x_{i,1}, \cdots, x_{i,T}]$.

To sum up, the hitting cost at time t is defined as

$$f_t(x_t) = \sum_{i=1}^{N} p_{i,t}^{e} \cdot \gamma_i \cdot q \cdot x_{i,t} + u_1 \sum_{i=1}^{N} \|x_{i,t}\|^2 \tag{8}$$

Besides, the switching cost $d(x_t, x_{t-1}) = \frac{u_2}{2} \|x_t - x_{t-1}\|^2$ penalizes the frequent or large changes between consecutive actions. Here, u_2 is a hyperparameter that reflects the overhead associated with hardware reconfiguration and communication arising from dynamically routed workloads.

Scaling laws are observed in large AI services [60], such as large language models, demonstrating an exponential increase in computational requirements to achieve continuous performance improvements.

Location	Average Health Price ($/MWh)	Average Electricity Price ($/MWh)	PUE (γ_i)
Arizona	17.29	77.7	1.18
Iowa	62.81	62.6	1.16
Illinois	49.93	82.6	1.35
Texas	48.19	63.0	1.28
Virginia	52.68	87.0	1.14
Washington	17.55	62.0	1.15
Wyoming	34.79	76.1	1.11

Table 1: Information for selected data centers

This rapid escalation in computation translates to substantial energy consumption. Due to the air pollution generated during electricity production, the increased electricity consumption associated with these models contributes to public health risks within local communities. Following the methodology in [52], we denote the health damage rate as $A_t = [A_{1,t}, \cdots, A_{N,t}]$, which reflects the associated air pollution and consequent health risks resulting from the electricity consumption. By strategically aligning AI computation with time periods and locations that are abundant with cleaner energy sources (e.g., solar, wind), the health damage rate can be significantly minimized. At the same time, ensuring fair distribution of public health risks is crucial when dynamically routing AI workloads. We incorporate a long-term fairness cost that quantifies the degree of public health impacts on the most affected region across all AI data centers, formulated as below

$$g\Big(x_{1:T}\Big) = u_3 \left\| \frac{q}{T} \sum_{t=1}^{T} A_t \cdot x_t^\top \right\|_p \tag{9}$$

where the L_p-norm can recover the widely-considered min-max fairness function as p approaches infinity. By regularizing the overall public health risk in the long-term, we can not only promote social fairness and mitigate widening socioeconomic gaps, while maintaining sufficient flexibility for dynamic AI workload routing.

By summing up the hitting, switching and long-term fairness cost, the objective function is defined as below

$$\begin{aligned} &\arg\min_{x_{1:T}} \frac{1}{T} \sum_{t=1}^{T} f_t(x_t) + \frac{1}{T} \sum_{t=1}^{T} d(x_t, x_{t-1}) + g\Big(x_{1:T}\Big) \\ s.t. \quad &\sum_{i=1}^{N} x_{i,t} = w_t, \ \ \forall t \in [1, T] \\ &x_{i,t} \leq M_i, \ \forall i \in [1, N], \ \forall t \in [1, T] \end{aligned} \tag{10}$$

B.1.2 Datasets and parameter settings

We employ a publicly available inference trace dataset for LLM services on Azure [37]. We normalize a sample of coding-related inference requests processed by multiple LLM services within Azure. These traces are collected between May 10th and May 16th, 2024. The dataset provides user demand patterns across different times of the week.

Specifically, we aggregate the number of requests received hourly to simulate the total computational workload across different times of the day.

To meet this workload, we assumed that the total computing demand could be distributed across seven selected data centers located in Arizona, Iowa, Illinois, Texas, Virginia, Washington, and Wyoming (N = 7). The data center information is obtained from [61]. We use the average state industrial electricity price [63], denoted as $p^e_{i,t}$, to calculate the electricity cost for each data center. For γ_i, we use the values reported by [62] to conduct our experiment.

For health impact analysis, air pollutant emissions resulting from electricity consumption contribute to various adverse effects, such as increased hospitalizations, higher medication usage, more frequent emergency room visits, and additional lost school days. These impacts can be quantified in economic

Metrics	OPT	FairOPT	HITMIN	ROBD	DMD	FairOBD		
						$\eta = 10^{-2}$	$\eta = 10^{-3}$	$\eta = 10^{-4}$
Hitting Cost	171.30	177.20	**159.75**	163.63	169.46	170.06	167.80	167.47
Switching Cost	23.27	351.48	43.75	**23.16**	93.53	25.65	27.52	28.17
Fairness Cost	36.36	33.33	140.12	111.85	54.16	**41.36**	51.05	52.68
Total Cost	230.93	562.00	343.62	298.64	317.15	**237.07**	246.36	248.31

Table 2: The average costs of different algorithms in the default setting (i.e. $u_1 = 10$, $u_2 = 1000$ and $u_3 = 3.5$). Minimum costs for the online algorithms are highlighted in bold.

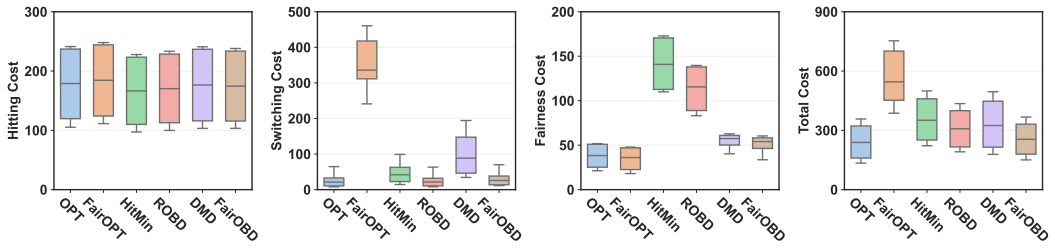

Figure 1: Cost distributions of different algorithms in the default setting (i.e. $u_1 = 10$, $u_2 = 1000$ and $u_3 = 3.5$)

terms using epidemiological and economic research on the associated health outcomes, collectively referred to as health costs. WattTime [52] divides the United States into distinct regions and provides real-time health prices ($/MWh), representing the health cost ($) per unit of energy consumption (MWh) across different regions and time periods. To determine the region in which each data center is located, we utilize the information from [61]. We use the health damage rate to estimate the health cost of energy consumption for each data center. Specifically, we set $A_t = [\gamma_i \cdot p^h_{i,t}]_{i=1}^N$, where $p^h_{i,t}$ represents the hourly average of WattTime's health price. Our online algorithm's ability to dynamically update its dual variable and learn from the online context sequence eliminates the need for prior training on a separate dataset. Therefore, we only build a testing dataset for all baseline algorithms. This testing dataset consists of 97 three-day (72-hour) context sequences, created by applying a sliding window across a one-week (168-hour) context sequence.

To provide readers with a high-level overview of the data centers' operational characteristics, Table 1 summarizes several key metrics including the average electricity price and average health price observed from May 10th to May 16th, 2024, as well as the Power Usage Effectiveness (PUE) and geographic location for each facility.

In the default setting, we set $q = 1$ to map the provisioned computing resource to energy consumption. The weights are $u_1 = 10$ for the regularizer in hitting cost, $u_2 = 1000$ for switching cost and $u_3 = 3.5$ for the long-term fairness cost. We choose the identical maximum capacity for each data center with $M_i = 1$ and normalize maximum workload traces according to the maximum capacity. We use $p = \infty$ for the l_p norm. These values are chosen to ensure that the hitting cost, switching cost, and fairness cost have comparable magnitudes. We set the initial dual variable $\kappa_1 = [3]_{i=1}^N$, λ_1 = 1, λ_2 = 30 for `DMD` and `FairOBD`.

The hyperparameters λ_1 and λ_2 for `ROBD` are optimally selected based on Theorem 4 in [18]. For `DMD`, we use the same set of hyperparameter as `FairOBD`, such as the default learning rate $\eta = 10^{-3}$, except that `DMD` ignores the switching cost. Regardless of the algorithm. The total costs of all the baseline algorithms are calculated with Eqn (10).

Our experiments are conducted on a MacBook Air with an M3 chip and 16 GB of memory. The average execution time per online algorithm over 72 time slots is approximately 1 second.

B.1.3 Baseline Algorithms

We consider the following baseline algorithms for comparison.

- **Optimal Fairness Offline** (`FairOPT`): the offline optimal policy solely minimizing long-term fairness cost.

Metrics	OPT	FairOPT	HITMIN	ROBD	DMD	FairOBD		
						$\eta = 10^{-2}$	$\eta = 10^{-3}$	$\eta = 10^{-4}$
Hitting Cost	168.88	177.20	**159.75**	163.63	169.46	169.39	167.80	167.47
Switching Cost	22.92	351.49	43.75	**23.16**	93.53	25.67	27.52	28.17
Fairness Cost	20.33	16.67	70.06	55.93	27.08	**21.35**	25.52	26.34
Total Cost	212.13	545.35	273.56	242.71	290.07	**216.41**	220.84	221.97

Table 3: Cost comparison of different algorithms with the fairness weight $u_3 = 1.75$.

Metrics	OPT	FairOPT	HITMIN	ROBD	DMD	FairOBD		
						$\eta = 10^{-2}$	$\eta = 10^{-3}$	$\eta = 10^{-4}$
Hitting Cost	171.81	177.20	**159.75**	163.63	169.46	170.52	167.80	167.47
Switching Cost	23.97	351.49	43.75	**23.16**	93.53	25.70	27.52	28.17
Fairness Cost	70.82	66.66	280.24	223.70	108.32	**82.40**	102.10	105.35
Total Cost	266.60	595.35	483.75	410.49	371.31	**278.62**	297.41	300.99

Table 4: Cost comparison of different algorithms with the fairness weight $u_3 = 7$.

- **Hitting Cost Minimizer** (HITMIN): the online algorithm chasing the minimizer of the hitting cost.
- **Regularized Online Balanced Descent** (ROBD): the state-of-the-art online algorithm without accounting the long-term fairness cost.
- **Dual Mirror Descent** (DMD): the online algorithm solely optimizing long-term fairness.
- **Optimal Offline** (OPT): the strongest optimal offline benchmark, corresponding to $(T, 0)$-OPT in our analysis. No online algorithms can possibly outperform OPT.

Among these algorithms, ROBD achieves the lowest competitive ratio for smoothed online convex optimization [18], which is an adapted version of Eqn (1) by removing the fairness cost. Besides, DMD is proposed for online allocation problems without switching costs [6], [57], [39], which updates the dual variable κ_t based on mirror descent techniques. We adapt DMD with a reference function $h(\kappa) = \frac{\|\kappa\|^2}{2}$ in our experiment.

B.2 Results

We summarize the average costs of the online algorithms and their offline optimal benchmark under the default setting in Table 2. The proposed algorithm, FairOBD, maintains robust performance across different learning rates η and consistently outperforms other online baseline algorithms (HITMIN, ROBD, and DMD) in terms of total cost. Moreover, FairOBD attains the minimum fairness cost among all online baselines, demonstrating its effectiveness in minimizing the long-term fairness cost.

To illustrate the minimal achievable long-term fairness cost, we incorporate FairOPT as the offline fairness-optimal baseline, considering access to the complete context sequence. Due to the inherent trade-off between operational cost and fairness cost, the locations with the lowest health impact do not necessarily align with those offering lower electricity prices. Furthermore, exclusively targeting such locations would exacerbate fluctuations in server provisioning across different regions. These two factors explain why FairOPT achieves the lowest fairness cost, despite its hitting and switching costs being among the highest observed, as shown in Figure 1. Conversely, HITMIN operates in the opposite direction, focusing exclusively on minimizing hitting cost while significantly increasing the other two costs. To address this challenge, DMD resolves the tension between long-term fairness cost and instantaneous operational cost by decomposing the long-term costs and optimizing them using mirror descent techniques. This approach enables an effective balance between these two competing metrics.

Without explicitly accounting for smoothness during the load balancing process, the aforementioned three methods minimize fairness and/or hitting costs by aggressively targeting locations with the lowest operational costs. These approach inevitably incur a significantly higher switching cost. For instance, the average switching cost of FairOPT is still around 15 times that of the offline optimal benchmark. In extreme cases, this ratio could be even higher, indicating considerable fluctuations in server provisioning—an undesirable strategy for data centers. It highlights the critical need to explicitly incorporate switching costs into the optimization framework. ROBD strategically balances between the minimizer of the hitting cost and the previous action, achieving the lowest switching

cost while maintaining a reasonably low hitting cost. However, due to the non-separable nature of the long-term cost, ROBD fails to minimize the fairness cost as a result of its algorithm design. By decomposing the long-term cost using auxiliary variables and dynamically optimizing these variables with mirror descent techniques, FairOBD achieves the best trade-off between hitting, switching, and fairness costs. This approach results in the lowest fairness and total costs among all the online algorithms.

To compare the impact of fairness cost, we present additional results by considering $u_3 = 1.75$ and $u_3 = 7$ on the performance of FairOBD, summarized in Tables 3 and 4, respectively. Similar as in Table 2, we see that under three different learning rates η, FairOBD is very close to OPT in terms of the total cost regardless of the hyperparameters, which demonstrates the superior performance of FairOBD. The other baselines (i.e., FairOPT, HITMIN, ROBD, and DMD) all have substantially higher total costs. The other online baseline algorithms (ROBD and HITMIN), which do not explicitly account for long-term fairness cost, consistently exhibit poor fairness performance across different values of the fairness cost parameter u_3. In contrast, FairOBD consistently maintains a relatively low fairness cost while simultaneously minimizing hitting cost and ensuring action smoothness. These results demonstrate that incorporating long-term fairness cost into our algorithm design is not only effective but also robust.

In summary, FairOBD effectively balances action smoothness while maintaining a low hitting cost and outperforms all other online baselines in fairness cost. These results validate the superiority of FairOBD 's explicit incorporation of long-term fairness in fostering socially responsible decision-making.

C Proof of the Lower Bounds

C.1 Proof of Theorem 4.1

Proof. In this proof, we consider hitting, switching and long-term cost as following,

$$cost(x_{1:T}) = \frac{1}{T}\sum_{t=1}^{T}\frac{m_0}{2}\|x_t - c_t\|^2 + \|\frac{1}{T}\sum_{t=1}^{T}A_t x_t\|_p, \tag{11}$$

where the horizon length T is known in advance. For the hitting cost, we choose the constant value m_0 satisfying $m_0 > 1$. For an online policy, the context c_t and A_t are revealed sequentially, and the future context can be chosen adversarially based on the actions from the online optimal policy. The offline policy has the complete information about the context.

Now we construct two selections for the online context sequence

$$\text{Option 1: } \begin{cases} c^*_{1:T} = [\underbrace{1,1,\ldots,1}_{T/2},\underbrace{0,0,\ldots,0}_{T/2}] \\ A^*_{1:T} = [\underbrace{1,1,\ldots,1}_{T/2},\underbrace{0,0,\ldots,0}_{T/2}] \end{cases} \tag{12}$$

$$\text{Option 2: } \begin{cases} c^\diamond_{1:T} = [\underbrace{1,1,\ldots,1}_{T}] \\ A^\diamond_{1:T} = [\underbrace{1,1,\ldots,1}_{T/2},\underbrace{-1,-1,\ldots,-1}_{T/2}] \end{cases} \tag{13}$$

Let $x^\dagger_{1:T}$ denote the action sequence generated by the online algorithm. For the first half of the horizon, $t \in [1, T/2]$, the two context sequences $(c^*_{1:T}, A^*_{1:T})$ and $(c^\diamond_{1:T}, A^\diamond_{1:T})$ are identical. The decision of which context sequence to choose is made at time $T/2 + 1$. Before revealing the context at this time, we examine the historical actions $x^\dagger_{1:T/2}$ of the expert policy.

For the online algorithm, we define the sum of action sequence as

$$\sum_{t=1}^{T/2} x^\dagger_t = \frac{T}{2}(S^\dagger + \epsilon(T)), \tag{14}$$

where $S^\dagger$ is a constant value and $\epsilon(T)$ is a vanishing term satisfying $\lim_{T\to\infty}\epsilon(T)=0$. We consider two cases about $S^\dagger$.

Case 1: $|S^\dagger| < \frac{2\,m_0-1}{2\,m_0}$.

In this case, we choose the second context trace $(c^\diamond_{1:T}, A^\diamond_{1:T})$. And the offline optimal action sequence can be obtained as

$$x^*_{1:T} = c^\diamond_{1:T}. \tag{15}$$

And the total cost is calculated as

$$cost(x^*_{1:T}) = 0 + \big|\sum_{t=1}^{T} A^\diamond_t c^\diamond_t\big| = 0 \tag{16}$$

In this case, for any action sequence $x^\dagger_{T/2+1:T}$ chosen by the online algorithm, the cost of the expert algorithm is lower bounded by

$$\begin{aligned} cost(x^\dagger_{1:T}) &= \frac{1}{T}\sum_{t=1}^{T}\frac{m_0}{2}\|x^\dagger_t - c_t\|^2 + \|\frac{1}{T}\sum_{t=1}^{T} A_t x_t\|_p \\ &\geq \frac{1}{T}\sum_{t=1}^{T/2}\frac{m_0}{2}\|x^\dagger_t - 1\|^2 \\ &\geq \frac{m_0}{2\,T}\frac{T}{2}\left\|\frac{\sum_{t=1}^{T/2} x^\dagger_t - T/2}{T/2}\right\|^2 \\ &= \frac{m_0}{4}\Big(S^\dagger - 1 + \epsilon(T)\Big)^2 \end{aligned} \tag{17}$$

where the first inequality is due to the non-negativity of the hitting and long-term costs, the second inequality is based on RMS–AM inequality. Given $\epsilon(T)$ is a vanishing term, the regret is lower bounded by

$$cost(x^\dagger_{1:T}) - cost(x^*_{1:T}) \gtrsim \frac{m_0}{4}\|S^\dagger - 1\|^2 > \frac{m_0}{4}(\frac{1}{2\,m_0})^2 = \frac{1}{16 m_0} \tag{18}$$

Case 2: $|S^\dagger| \geq \frac{2\,m_0-1}{2\,m_0}$.

In this case, we choose the first context trace $(c^*_{1:T}, A^*_{1:T})$. And one feasible action sequence is defined as

$$x_{1:T} = \frac{m_0 - 1}{m_0}\, c^*_{1:T} \tag{19}$$

We don't need to prove the optimality of this feasible action sequence, as any the offline optimal cost must be upper bounded by the cost of this feasible action sequence, which is

$$\begin{aligned} cost(x^*_{1:T}) &\leq \frac{1}{T}\sum_{t=1}^{T}\frac{m_0}{2}\|x_t - c_t\|^2 + \|\frac{1}{T}\sum_{t=1}^{T} A_t x_t\|_p \\ &= \frac{1}{T}\frac{m_0}{2}\frac{T}{2}\frac{1}{m_0^2} + \frac{1}{T}\frac{T}{2}\frac{m_0-1}{m_0} \\ &= \frac{1}{4m_0} + \frac{m_0-1}{2m_0} = \frac{2m_0-1}{4m_0} \end{aligned} \tag{20}$$

For the online algorithm, we have

$$\begin{aligned} cost(x^\dagger_{1:T}) &= \frac{1}{T}\sum_{t=1}^{T/2}\frac{m_0}{2}\|x^\dagger_t - c_t\|^2 + \|\frac{1}{T}\sum_{t=1}^{T/2}A_t x_t\|_p \\ &\geq \frac{1}{T}\sum_{t=1}^{T/2}\frac{m_0}{2}\|x^\dagger_t - 1\|^2 + \frac{1}{2}\|S^\dagger + \epsilon(T)\|_p \\ &\geq \frac{m_0}{2}\frac{T}{2}\left\|\frac{\sum_{t=1}^{T/2}x^\dagger_t - T/2}{T/2}\right\|^2 + \frac{1}{2}\|S^\dagger + \epsilon(T)\|_p \\ &= \frac{m_0}{4}\|S^\dagger - 1 + \epsilon(T)\|^2 + \frac{1}{2}\|S^\dagger + \epsilon(T)\|_p \\ &\gtrsim \frac{m_0}{4}(S^\dagger - 1)^2 + \frac{1}{2}\|S^\dagger\| \end{aligned} \tag{21}$$

Then the regret of the online algorithm is lower bounded by

$$cost(x^\dagger_{1:T}) - cost(x^*_{1:T}) \gtrsim \frac{m_0}{4}(S^\dagger - 1)^2 + \frac{1}{2}\|S^\dagger\| - \frac{2m_0 - 1}{4m_0} \tag{22}$$

Based on our condition on $S^\dagger$, if $S^\dagger \leq -\frac{2\,m_0-1}{2\,m_0}$, it's obvious that

$$\frac{m_0}{4}(S^\dagger - 1)^2 + \frac{1}{2}\|S^\dagger\| - \frac{2m_0 - 1}{4m_0} \geq \frac{m_0}{4}(1 + \frac{2\,m_0 - 1}{2\,m_0})^2. \tag{23}$$

On the other hand, if $S^\dagger > 0$, the minimizer of $\frac{m_0}{2}(S^\dagger - 1)^2 + \frac{1}{2}\|S^\dagger\| - \frac{2m_0-1}{4m_0}$ is achieved at the critical point $S^\dagger = 1 - \frac{1}{2m_0}$. Therefore, $S^\dagger = 1 - \frac{1}{2\,m_0}$ is the minimizer for the bound as the bound monotonically increases as $S > 0$. So the lower bound in this case can be expressed as

$$cost(x^\dagger_{1:T}) - cost(x^*_{1:T}) \gtrsim \frac{m_0}{4}(\frac{1}{2\,m_0})^2 + \frac{1}{2}\|1 - \frac{1}{2\,m_0}\| - \frac{2m_0 - 1}{4m_0} = \frac{1}{16\,m_0} \tag{24}$$

Given $m_0 > \frac{3}{2}$, the lower bounded for cost gap is obtained by combining the bounds for these two conditions for $S^\dagger$, which is

$$cost(x^\dagger_{1:T}) - cost(x^*_{1:T}) \gtrsim \frac{1}{16m_0}. \tag{25}$$

To sum up, compared with the offline optimal action, the regret of **any online algorithm** is lower bounded by $\Omega(1)$.

□

C.2 Proof of Theorem 4.2

Proof. In this proof, we construct the hitting, switching and long-term cost as following,

$$cost(x_{1:T}) = \frac{1}{T}\sum_{t=1}^{T}\frac{m_0}{2}\|x_t - c_t\|^2 + \|\frac{1}{T}\sum_{t=1}^{T}A_t x_t\|_p, \tag{26}$$

where the horizon length T is known in advance and $m_0 > 1$. For an online policy, the context c_t and A_t are revealed sequentially, and the future context can be chosen adversarially based on the actions from the online policy. The offline policy has the complete information about the context. Our goal is to derive a lower bound for the competitive ratio between online and offline policies under this information asymmetry.

We construct two selections for the online context sequence

$$\text{Option 1: } \begin{cases} c^*_{1:T} = [\underbrace{a, a, \ldots, a}_{T/2}, \underbrace{0, 0, \ldots, 0}_{T/2}] \\ A^*_{1:T} = [\underbrace{1, 1, \ldots, 1}_{T/2}, 1, \underbrace{0, 0, \ldots, 0}_{T/2-1}] \end{cases} \tag{27}$$

$$\text{Option 2:} \begin{cases} c^{\diamond}_{1:T} = [\underbrace{a, a, \ldots, a}_{T}] \\ A^{\diamond}_{1:T} = [\underbrace{1, 1, \ldots, 1}_{T/2}, \underbrace{-1, -1, \ldots, -1}_{T/2}] \end{cases} \tag{28}$$

Let $x^{\dagger}_{1:T}$ denote the action sequence generated by the online algorithm. For the first half of the horizon, $t \in [1, T/2]$, the two context sequences $(c^{*}_{1:T}, A^{*}_{1:T})$ and $(c^{\diamond}_{1:T}, A^{\diamond}_{1:T})$ are identical. The decision of which context sequence to choose is made at time $T/2 + 1$. Before revealing the context at this time, we examine the historical actions $x^{\dagger}_{1:T/2}$ of the online policy.

If there exists a time $t \in [1, T/2]$ such that $x^{\dagger}_{t} \neq a$, we select $(c^{\diamond}_{1:T}, A^{\diamond}_{1:T})$ as the context sequence. In this scenario, the offline optimal action is consistently a, resulting in a zero total cost. However, the expert's cost will be non-zero, leading to an infinite competitive ratio.

Conversely, if the online action sequence is $x^{\dagger}_{1:T/2} = [a, a, \cdots, a]$, we will choose $(c^{*}_{1:T}, A^{*}_{1:T})$ as the context sequence. The best an online algorithm can do is to only optimize the action in $x^{\dagger}_{T/2+1}$ and set all subsequent actions as zero. We define the objective for action at $x^{\dagger}_{T/2+1}$ as

$$\phi_1(x) = \frac{m_0}{2}(x)^2 + \left\| \frac{T}{2} a + x \right\| \tag{29}$$

In our current setting, x is a scalar and the gradient of $g_1(x)$ is defined as

$$g_1'(x) = \begin{cases} m_0 x + 1 & x > -\frac{T\,a}{2} \\ m_0 x - 1 & x < -\frac{T\,a}{2} \\ \text{undefined} & x = -1 \end{cases} \tag{30}$$

In our following proof, we will choose the constant $a \geq \frac{2}{m_0\,T}$ (e.g. $a = \frac{2}{m_0\,T}$). To preserve generality, we will work with a symbolically. Given that $m_0 > 0$ and $T > 2$, for all $x < -\frac{T\,a}{2}$, we can derive the upper bound of the gradient as follows, $g_1'(x) = m_0 x - 1 < -m_0 \frac{T\,a}{2} - 1 < -1$. This implies that $g_1(x)$ is monotonically decreasing as x approaches $-\frac{T\,a}{2}$ from the left. Since $\phi_1(x)$ is continuous, the minimizer cannot be located in the region $x < -\frac{T\,a}{2}$.

On the other hand, as x approaches $-\frac{T\,a}{2}$ from the right ($x > -\frac{T\,a}{2}$), the gradient $g_1'(x) = m_0 x + 1$ approaches $-m_0 \frac{T\,a}{2} + 1 \leq -1 + 1 \leq 0$. Given $\phi_1(x)$ is continuous, this suggests that the minimizer of $\phi_1(x)$ must lie in the region $x \geq -\frac{T\,a}{2}$. To find the critical point in this region, we set $g_1'(x) = 0$ for $x > -1$, which is $m_0 x + 1 = 0$. Solving for x, we obtain the potential minimizer:

$$x^{\dagger,*}_{T/2+1} = -\frac{1}{m_0} \tag{31}$$

And the minimum value of $g_1(x)$ becomes

$$\begin{aligned} g_1(x^{\dagger,*}_{T/2+1}) &= \frac{1}{T}\left[\frac{m_0}{2}(x)^2 + \left\| \frac{T}{2} a + x \right\|\right] \\ &= \frac{1}{T}\left[\frac{1}{2\,m_0} + \frac{T}{2} a - \frac{1}{m_0}\right] \\ &= \frac{1}{2} a - \frac{1}{2 m_0\,T}, \end{aligned} \tag{32}$$

where the second equality is due to our choice of $a \geq \frac{2}{m_0\,T}$. The total cost of **any online algorithm** in this case is lower bounded by

$$cost(x^{\dagger}_{1:T}) \geq \frac{1}{2} a - \frac{1}{2 m_0\,T}. \tag{33}$$

Next, we aim to derive the upper bound for the offline policy with complete context sequence. As we only focus on the scale of the competitive ratio's lower bound, it's not necessary to explicitly solve the true offline optimal trace, which could be very messy. Instead, we choose the following action sequence to serve as a feasible action sequence $x'_{1:T}$, which is

$$x'_t = \begin{cases} 0 & t \in [1, \frac{T}{2}] \\ -a & t = \frac{T}{2} + 1 \\ 0 & t \in [\frac{T}{2} + 1, T] \end{cases} \tag{34}$$

Then the total cost of the true offline optimal cost is bounded by

$$cost(x^*_{1:T}) \leq cost(x'_{1:T}) = \frac{1}{T}\left[\frac{m_0}{2}(\frac{T}{2} + 1)a^2 + a\right] \tag{35}$$

Here we choose $a = \frac{2}{m_0 T}$. The lower bound for the cost of any online algorithm and the upper bound for the offline optimal cost is given by

$$\begin{aligned} cost(x^\dagger_{1:T}) &\geq \frac{1}{2}a - \frac{1}{2m_0 T} = \frac{1}{2m_0 T} \\ cost(x^*_{1:T}) &\leq \frac{1}{T}\left[\frac{m_0}{2}(\frac{T}{2} + 1)\, a^2 + a\right] = \frac{3}{m_0 T^2} + \frac{2}{m_0 T^3}. \end{aligned} \tag{36}$$

The competitive ratio is lower bounded by

$$CR = \frac{cost(x^\dagger_{1:T})}{cost(x^*_{1:T})} \geq \frac{\frac{1}{2m_0}}{\frac{3}{m_0 T} + \frac{2}{m_0 T^2}} = \frac{T}{6 + \frac{4}{T}} \gtrsim \frac{T}{6} \tag{37}$$

In other words, the competitive ratio of **any online algorithm** is lower bounded by $\Omega(T)$. □

D Proof of Asymptotic Competitive Ratio for `FairOBD`

For the ease of presentation, we define the drift of Bregman divergence for the dual variable κ at time t as

$$\Delta(t) = V_h(\kappa_1, \kappa_{t+1}) - V_h(\kappa_1, \kappa_t) \tag{38}$$

To facilitate the proof of the asymptotic competitive ratio for `FairOBD` in different settings, we first present several technical lemmas.

Lemma D.1. *Suppose κ_{t+1} is the solution to the following equation*

$$\kappa_{t+1} = \arg\min_{\kappa \in \mathbb{R}^N} \langle d_t, \kappa \rangle + \frac{1}{\eta} V_h(\kappa, \kappa_t) \tag{39}$$

Then for any $\kappa' \in \mathbb{R}^N$, we have

$$\langle \kappa_{t+1}, d_t \rangle + \frac{1}{\eta} V_h(\kappa_{t+1}, \kappa_t) \leq \langle \kappa', d_t \rangle + \frac{1}{\eta} V_h(\kappa', \kappa_t) - \frac{1}{\eta} V_h(\kappa', \kappa_{t+1}) \tag{40}$$

Proof. For simplicity, we define

$$e(\kappa) = \langle d_t, \kappa \rangle + \frac{1}{\eta} V_h(\kappa, \kappa_t) \tag{41}$$

The gradient of $e(\kappa)$ with respect to κ is defined as

$$\nabla_\kappa e(\kappa) = d_t + \frac{1}{\eta}\big(\nabla h(\kappa) - \nabla h(\kappa_t)\big) \tag{42}$$

Since κ_{t+1} is the minimizer of $e(\kappa)$, then for any $\kappa' \in \mathbb{R}^N$ we must have

$$\langle \kappa' - \kappa_{t+1}, d_t + \frac{1}{\eta}\big(\nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\big)\rangle \geq 0 \tag{43}$$

We prove this by contradiction. Suppose there exist κ', such that $\langle \kappa' - \kappa_{t+1}, \nabla_\kappa e(\kappa)\rangle < 0$, then we construct a function $y(\xi) = e(\kappa_{t+1} + \xi(\kappa' - \kappa_{t+1}))$, where $\xi \in [0, 1]$. The gradient of $y(\xi)$ is

$$\nabla_\xi y(\xi) = \left\langle \kappa' - \kappa_{t+1}, d_t - \frac{1}{\eta}\Big(\nabla h(\kappa_t) - \nabla h\big(\kappa_{t+1} + \xi(\kappa' - \kappa_{t+1})\big)\Big)\right\rangle \tag{44}$$

Then $\nabla_\xi y(0) = \langle \kappa' - \kappa_{t+1}, d_t + \frac{1}{\eta}\big(\nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\big)\rangle < 0$. Therefore $\xi = 0$ is not the minimizer of $y(\xi)$ or in other words, κ_{t+1} is not the minimizer of $e(\kappa)$, which is contradictory to our assumption. So the inequality in Eqn (43) must hold for any $\kappa' \in \mathbb{R}^N$. By organizing the inequality, we have

$$\begin{aligned}\langle \kappa_{t+1}, d_t\rangle \leq& \langle \kappa', d_t\rangle + \frac{1}{\eta}\langle \kappa', \nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\rangle - \frac{1}{\eta}\langle \kappa_{t+1}, \nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\rangle \\ =& \langle \kappa', d_t\rangle + \frac{1}{\eta}\langle \kappa' - \kappa_{t+1}, \nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\rangle\end{aligned} \tag{45}$$

According to the definition of Bregman Divergence, we have

$$\begin{aligned}&\langle \kappa' - \kappa_{t+1}, \nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\rangle \\ =&\langle \kappa' - \kappa_{t+1}, \nabla h(\kappa_{t+1})\rangle - \langle \kappa' - \kappa_{t+1}, \nabla h(\kappa_t)\rangle \\ =&\langle \kappa' - \kappa_{t+1}, \nabla h(\kappa_{t+1})\rangle - \langle \kappa' - \kappa_t, \nabla h(\kappa_t)\rangle + \langle \kappa_{t+1} - \kappa_t, \nabla h(\kappa_t)\rangle \\ =& - h(\kappa') + h(\kappa_{t+1}) + \langle \kappa' - \kappa_{t+1}, \nabla h(\kappa_{t+1})\rangle - h(\kappa_t) + h(\kappa') - \langle \kappa' - \kappa_t, \nabla h(\kappa_t)\rangle \\ &- h(\kappa_{t+1}) + h(\kappa_t) + \langle \kappa_{t+1} - \kappa_t, \nabla h(\kappa_t)\rangle \\ =& - V_h(\kappa', \kappa_{t+1}) + V_h(\kappa', \kappa_t) - V_h(\kappa_{t+1}, \kappa_t)\end{aligned} \tag{46}$$

Then we finish the proof. □

Lemma D.2. *If the reference function $h(\cdot)$ is l-strongly convex and κ_1 is the initial dual variable, then the dual variable κ_t is bounded by*

$$V_h(\kappa_1, \kappa_{T+1}) \leq T(\eta^2 \frac{Z^2}{2l} + \eta L Z) - \kappa_1^\top \cdot (\sum_{t=1}^{T} A_t x_t - z_t) \tag{47}$$

where the constant $Z = \sup_{z_t, x_t, A_t} \|z_t - A_t x_t\|$.

Proof. According to the Line 7 of Algorithm 1 and Lemma D.1, for any $\kappa' \in \mathbb{R}_+^N$ we have

$$\langle \kappa_{t+1}, d_t\rangle + \frac{1}{\eta} V_h(\kappa_{t+1}, \kappa_t) \leq \langle \kappa', d_t\rangle + \frac{1}{\eta} V_h(\kappa', \kappa_t) - \frac{1}{\eta} V_h(\kappa', \kappa_{t+1}) \tag{48}$$

By setting $\kappa' = \kappa_1$ and add $\langle \kappa_t, d_t\rangle$ to both sides of Eqn (48), we have

$$\begin{aligned}\langle \kappa_t - \kappa_1, d_t\rangle \leq& \left(\langle \kappa_t - \kappa_{t+1}, d_t\rangle - \frac{1}{\eta} V_h(\kappa_{t+1}, \kappa_t)\right) \\ &+ \frac{1}{\eta} V_h(\kappa_1, \kappa_t) - \frac{1}{\eta} V_h(\kappa_1, \kappa_{t+1})\end{aligned} \tag{49}$$

Since the reference function $h(a)$ is l-strongly convex. then

$$h(a) \geq h(b) + (a - b)\nabla h(b) + \frac{l}{2}\|a - b\|^2 \tag{50}$$

In other words

$$V_h(\kappa_{t+1}, \kappa_t) \geq \frac{l}{2}\|\kappa_{t+1} - \kappa_t\|^2 \tag{51}$$

Therefore, we have

$$\begin{aligned}&\frac{1}{\eta} V_h(\kappa_{t+1}, \kappa_t) - \langle \kappa_t - \kappa_{t+1}, d_t\rangle \\ =&\langle \kappa_{t+1} - \kappa_t, d_t\rangle + \frac{1}{\eta} V_h(\kappa_{t+1}, \kappa_t) \\ \geq& - \|\kappa_{t+1} - \kappa_t\|\|d_t\| + \frac{l}{2\eta}\|\kappa_{t+1} - \kappa_t\|^2 \\ \geq& - \frac{l}{2\eta}\|\kappa_{t+1} - \kappa_t\|^2 - \frac{\eta}{2l}\|d_t\|^2 + \frac{l}{2\eta}\|\kappa_{t+1} - \kappa_t\|^2 \\ =& - \frac{\eta}{2l}\|d_t\|^2 \geq -\frac{\eta}{2l} Z^2\end{aligned} \tag{52}$$

The first inequality holds by the definition of inner product and the l-strongly convexity assumption of reference function $h(\cdot)$, the second inequality is derived with the AM-GM inequality, and the last inequality is obtained with the assumption $\|d_t\| \leq Z$. According to the definition of $\Delta(t)$, Eqn (49) becomes

$$\langle \kappa_t, z_t - A_t x_t \rangle + \frac{\Delta(t)}{\eta} \leq \frac{\eta}{2l} Z^2 + \langle \kappa_1, z_t - A_t x_t \rangle \tag{53}$$

According to Line 4 and Line 5 of algorithm 1, then for any $x'_t \in \mathcal{X}_t$ and $z'_t \in \mathcal{Z}$, we have

$$\begin{aligned} &f_t(x_t) + \lambda_1 d(x_t, x_{t-1}) + \frac{\lambda_2}{2}\|x_t - v_t\|^2 + g(z_t) + \kappa_t^\top \cdot (A_t x_t - z_t) \\ \leq &f_t(x'_t) + \lambda_1 d(x'_t, x_{t-1}) + \frac{\lambda_2}{2}\|x'_t - v_t\|^2 + g(z'_t) + \kappa_t^\top \cdot (A_t x'_t - z'_t) \end{aligned} \tag{54}$$

By adding $\left(\frac{\Delta(t)}{\eta} - \kappa_t^\top \cdot (A_t x_t - z_t)\right)$ to both sides of the inequality, we have

$$\begin{aligned} &\frac{\Delta(t)}{\eta} + f_t(x_t) + \lambda_1 d(x_t, x_{t-1}) + \frac{\lambda_2}{2}\|x_t - v_t\|^2 + g(z_t) \\ \leq &\left(\frac{\Delta(t)}{\eta} - \kappa_t^\top \cdot (A_t x_t - z_t)\right) + f_t(x'_t) + \lambda_1 d(x'_t, x_{t-1}) + \frac{\lambda_2}{2}\|x'_t - v_t\|^2 + g(z'_t) + \kappa_t^\top \cdot (A_t x'_t - z'_t) \\ \leq &\left(\frac{\eta}{2l} Z^2 - \kappa_1^\top \cdot (A_t x_t - z_t)\right) + f_t(x'_t) + \lambda_1 d(x'_t, x_{t-1}) + \frac{\lambda_2}{2}\|x'_t - v_t\|^2 + g(z'_t) + \kappa_t^\top \cdot (A_t x'_t - z'_t) \end{aligned} \tag{55}$$

Now we choose $x'_t = x_t$ and $z'_t = A_t x'_t$, then we have

$$\frac{\Delta(t)}{\eta} \leq \left(\frac{\eta}{2l} Z^2 - \kappa_1^\top \cdot (A_t x_t - z_t)\right) + g(z'_t) - g(z_t) \leq \frac{\eta}{2l} Z^2 + LZ - \kappa_1^\top \cdot (A_t x_t - z_t) \tag{56}$$

By summing up the difference over the episode T, we have

$$V_h(\kappa_1, \kappa_{T+1}) - V_h(\kappa_1, \kappa_1) \leq \eta T(\frac{\eta}{2l} Z^2 + LZ) - \eta \kappa_1^\top \cdot (\sum_{t=1}^{T} A_t x_t - z_t) \tag{57}$$

□

D.1 Proof of Theorem 5.1

In the problem setting without switching cost, we define the optimization objective as

$$G(x_{1:T}) = \frac{1}{T}\sum_{t=1}^{T}\Big[f_t(x_t)\Big] + g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t). \tag{58}$$

Proof. Since x_t, z_t is the optimal solution for Eqn (5), so we have the following equation for any $x'_t \in \mathcal{X}, z'_t \in \mathcal{Z}$:

$$f_t(x_t) + \kappa_t \cdot A_t x_t \leq f_t(x'_t) + \kappa_t \cdot A_t x'_t \tag{59}$$

$$g(z_t) - \kappa_t z_t \leq g(z'_t) - \kappa_t z'_t \tag{60}$$

By combining the inequalities in Eqn (53), (59) and (60), we have

$$\frac{\Delta(t)}{\eta} + f_t(x_t) + g(z_t) \leq \frac{\eta}{2l} Z^2 + f_t(x'_t) + g(z'_t) + \kappa_t^\top \cdot (A_t x'_t - z'_t) - \kappa_1^\top \cdot (A_t x_t - z_t) \tag{61}$$

For the last term of Eqn (61), since κ_{t+1} is the minimizer of $e(\kappa)$, then for any $\kappa' \in \mathbb{R}^N$ we must have

$$\langle \kappa' - \kappa_{t+1}, d_t + \frac{1}{\eta}\big(\nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\big)\rangle \geq 0 \tag{62}$$

This condition is satisfied only if the gradient is zero, which is

$$d_t + \frac{1}{\eta}\big(\nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\big) = 0. \tag{63}$$

Since the reference function is l-strongly convex, we can bound the dual update by the gradient distance in the following way

$$\|\kappa_{t+1} - \kappa_t\| \leq \frac{\eta}{l}\|\big(\nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\big)\| \leq \frac{\eta\|d_t\|}{l}. \tag{64}$$

Thus, we can have the following inequality for the last term of Eqn (61) by summing it up over the frame R, which is

$$\begin{aligned}\langle \kappa_{t+k} - \kappa_t, A_{t+k}x'_{t+k} - z'_{t+k}\rangle \leq & \Big\| \sum_{j=t}^{t+k-1} \big(\kappa_{j+1} - \kappa_j\big)\Big\| \cdot \big\|A_{t+k}x'_{t+k} - z'_{t+k}\big\| \\ \leq & \frac{\eta}{l} \cdot \Big(\sum_{j=t}^{t+k-1} \big\|A_j x_j - z_j\big\|\Big) \cdot Z \leq (k-1) \cdot \frac{\eta Z^2}{l}.\end{aligned} \tag{65}$$

Then within the frame of size R, we have

$$\left[\sum_{t=kR+1}^{kR+R} \kappa_t \cdot (A_t x'_t - z'_t)\right] \leq \kappa_{kR+1}\left[\sum_{t=kR+1}^{kR+R} (A_t x'_t - z'_t)\right] + \frac{R(R-1)}{2l}\eta Z^2. \tag{66}$$

Then by substituting Eqn (66) to Eqn (61) and summing up over the whole episode T, we have:

$$\begin{aligned}&\sum_{t=1}^{T} \frac{\Delta(t)}{\eta} + \sum_{t=1}^{T}\Big[f_t(x_t) + g(z_t)\Big] = \frac{V_h(\kappa_1, \kappa_{T+1})}{\eta} + \sum_{t=1}^{T}\Big[f_t(x_t) + g(z_t)\Big] \\ &\leq \sum_{t=1}^{T}\Big[f_t(x'_t) + g(z'_t)\Big] + \frac{\eta}{2l}Z^2 TR - \kappa_1^\top \cdot \left[\sum_{t=1}^{T} A_t x_t - z_t\right] + \left[\sum_{k=1}^{K} \kappa_{(k-1)R+1} \sum_{t=(k-1)R+1}^{kR} (A_t x'_t - z'_t)\right]\end{aligned} \tag{67}$$

Now suppose $x^*_{1:T}$ is the (R, δ)-optimal action sequence, then within the k-th frame we manually construct z^*_t as

$$z^*_t = \frac{1}{R}\sum_{i=kR+1}^{kR+R} A_i x^*_i,\ \forall t \in [kR+1, kR+R], 1 \leq k \leq K \tag{68}$$

Based on the L-Lipschitz assumption on function $g(\cdot)$, we have:

$$\begin{aligned}\sum_{t=1}^{T} g(z^*_t) =& \sum_{k=1}^{K} R \cdot g(\frac{1}{R}\sum_{t=(k-1)R+1}^{kR} A_t x^*_t) \\ \leq & R\sum_{k=1}^{K} g(\frac{1}{T}\sum_{t=1}^{T} A_t x^*_t) + L\sum_{k=1}^{K}\|\frac{1}{R}\sum_{t=(k-1)R+1}^{kR} A_t x^*_t - \frac{1}{T}\sum_{t=1}^{T} A_t x^*_t\| \\ \leq & \sum_{k=1}^{K}\Big[R \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x^*_t)\Big] + L\delta \\ = & Tg(\frac{1}{T}\sum_{t=1}^{T} A_t x^*_t) + L\delta\end{aligned} \tag{69}$$

Substituting it back to Eqn (67), we have

$$\begin{aligned}\sum_{t=1}^{T}\Big[f_t(x_t) + g(z_t)\Big] \leq & \sum_{t=1}^{T}\Big[f_t(x^*_t)\Big] + T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x^*_t) + \frac{\eta}{2l}Z^2 TR \\ & + L\delta - \frac{V_h(\kappa_1, \kappa_{T+1})}{\eta} - \kappa_1^\top \cdot \left[\sum_{t=1}^{T} A_t x_t - z_t\right]\end{aligned} \tag{70}$$

where the Bregman divergence $V_h(\kappa_1, \kappa_{T+1})$ is non-negative since the reference function is convex, which can be eliminated later. The last step is to upper bound $g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t)$ with $\frac{1}{T}\sum_{t=1}^{T} g(z_t)$, which is

$$\begin{aligned} & T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t) - \sum_{t=1}^{T} g(z_t) \\ \leq & TL\|\frac{1}{T}\sum_{t=1}^{T} A_t x_t - \frac{1}{T}\sum_{t=1}^{T} z_t\| + T\Big[g(\frac{1}{T}\sum_{t=1}^{T} z_t) - \frac{1}{T}\sum_{t=1}^{T} g(z_t)\Big] \\ \leq & L\|\sum_{t=1}^{T} A_t x_t - \sum_{t=1}^{T} z_t\| \end{aligned} \tag{71}$$

Since κ_{t+1} is the minimizer of $e(\kappa)$, we have

$$d_t + \frac{1}{\eta}\big(\nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\big) = 0. \tag{72}$$

In other words

$$\Big\|\sum_{t=1}^{T}(-d_t)\Big\| = \Big\|\sum_{t=1}^{T}\big(A_t x_t - z_t\big)\Big\| = \frac{\|\nabla h(\kappa_{T+1}) - \nabla h(\kappa_1)\|}{\eta} \tag{73}$$

Since the reference function $h(\cdot)$ is l-strongly convex and β_2-smooth, we have

$$\frac{\|\nabla h(\kappa_{T+1}) - \nabla h(\kappa_1)\|}{\eta} \leq \frac{\beta_2\|\kappa_{T+1} - \kappa_1\|}{\eta} \leq \frac{\beta_2}{\eta}\sqrt{\frac{2}{l}V_h(\kappa_1, \kappa_{T+1})} \tag{74}$$

By substituting the inequality to Eqn (71), we have

$$T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t) - \sum_{t=1}^{T} g(z_t) \leq \frac{\beta_2 L}{\eta}\sqrt{\frac{2}{l}V_h(\kappa_1, \kappa_{T+1})} \tag{75}$$

By substituting Lemma D.2 into the inequality, we have

$$T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t) - \sum_{t=1}^{T} g(z_t) \leq \beta_2 L\sqrt{T(\frac{Z^2}{l^2} + \frac{1}{\eta}\frac{2LZ}{l}) - \frac{2}{\eta^2 l}\kappa_1^\top \cdot (\sum_{t=1}^{T} A_t x_t - z_t)} \tag{76}$$

According to our assumption, we have $\Big|\kappa_1^\top \cdot (\sum_{t=1}^{T} A_t x_t - z_t)\Big| \leq TZ \cdot \|\kappa_1\|$. By substituting the above inequality into Eqn (70), we have

$$\begin{aligned} \sum_{t=1}^{T}\Big[f_t(x_t)\Big] + T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t) \leq & \sum_{t=1}^{T}\Big[f_t(x_t^*)\Big] + T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t^*) + \frac{\eta}{2l}Z^2 TR \\ & + TZ\|\kappa_1\| + L\delta + \beta_2 L\sqrt{T\Big(\frac{Z^2}{l^2} + \frac{1}{\eta}\frac{2LZ}{l} + \frac{2}{\eta^2 l}Z\|\kappa_1\|\Big)} \end{aligned} \tag{77}$$

By dividing both sides by T, we finish the proof. □

D.2 Proof of Theorem 5.2

For the ease of presentation, we define the optimization objective $G(x_{1:T}, \tilde{x}_{0:T-1})$ as

$$G(x_{1:T}, \tilde{x}_{0:T-1}) = \frac{1}{T}\sum_{t=1}^{T}\Big[f_t(x_t) + \lambda_1 d(x_t, \tilde{x}_{t-1}) + \frac{\lambda_2}{2}\|x_t - v_t\|^2\Big] + g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t) \tag{78}$$

To facilitate the proof of the asymptotic competitive ratio for `FairOBD` when the hitting cost is m-strongly convex, we first present several technical lemmas.

Lemma D.3. *Suppose the switching cost is defined as as $d(x_t, x_{t-1}) = \frac{\beta_1}{2}\|x_t - x_{t-1}\|^2$ and the action sequence $\{(x_t, z_t)|\forall t \in [1, T]\}$ is the solution using Algorithm 1, then for any $x(t)' \in \mathcal{X}$, $z(t)' \in \mathcal{Z}$, we have*

$$\begin{aligned}
&\sum_{t=1}^{T}\frac{\Delta(t)}{\eta} + \sum_{t=1}^{T}\Big[f_t(x_t) + \lambda_1 d(x_t, x_{t-1}) + g(z_t)\Big] + \sum_{t=1}^{T}\frac{\lambda_2}{2}\|x_t - v_t\|^2 \\
\leq &\sum_{t=1}^{T}\Big[f_t(x_t') + \lambda_1 d(x_t', x_{t-1}) + \frac{\lambda_2}{2}\|x_t' - v_t\|^2 + g(z_t')\Big] + \left[\sum_{k=1}^{K}\kappa_{(k-1)R+1}\sum_{t=(k-1)R+1}^{kR}(A_t x_t' - z_t')\right] \\
&- \frac{m + \lambda_1\beta_1 + \lambda_2}{2}\sum_{t=1}^{T}\|x_t' - x_t\|^2 + \frac{\eta}{2l}Z^2TR - \kappa_1^\top \cdot \left[\sum_{t=1}^{T}A_t x_t - z_t\right]
\end{aligned} \tag{79}$$

Proof. Since the function $f_t(x)$ is m-strongly convex and switching cost $d(x, x_{t-1})$is also β_1-strongly convex with respect to x by definition, then we have

$$\begin{aligned}
&f_t(x_t) + \lambda_1 d(x_t, x_{t-1}) + \frac{\lambda_2}{2}\|x_t - v_t\|^2 + \kappa_t \cdot A_t x_t + \frac{m + \lambda_1\beta_1 + \lambda_2}{2}\|x_t' - x_t\|^2 \\
\leq &f_t(x_t') + \lambda_1 d(x_t', x_{t-1}) + \frac{\lambda_2}{2}\|x_t' - v_t\|^2 + \kappa_t \cdot A_t x_t'
\end{aligned} \tag{80}$$

where x_t is the optimal solution for Eqn (5). Then similarly, for z_t, we have

$$g(z_t) - \kappa_t z_t \leq g(z_t') - \kappa_t z_t' \tag{81}$$

By combining the inequalities in Eqn (53), (80) and (81), we have

$$\begin{aligned}
&\frac{\Delta(t)}{\eta} + f_t(x_t) + \lambda_1 d(x_t, x_{t-1}) + \frac{\lambda_2}{2}\|x_t - v_t\|^2 + g(z_t) + \frac{m + \lambda_1\beta_1 + \lambda_2}{2}\|x_t' - x_t\|^2 \\
\leq &\frac{\eta}{2l}Z^2 + f_t(x_t') + \lambda_1 d(x_t', x_{t-1}) + \frac{\lambda_2}{2}\|x_t' - v_t\|^2 + g(z_t') + \kappa_t^\top \cdot (A_t x_t' - z_t') - \kappa_1^\top \cdot (A_t x_t - z_t)
\end{aligned} \tag{82}$$

For the last term of Eqn (82), since κ_{t+1} is the minimizer of $e(\kappa)$, then for any $\kappa' \in \mathbb{R}^N$ we must have

$$\langle \kappa' - \kappa_{t+1}, d_t + \frac{1}{\eta}\big(\nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\big)\rangle \geq 0 \tag{83}$$

This condition is satisfied only if the gradient is zero, which is

$$d_t + \frac{1}{\eta}\big(\nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\big) = 0. \tag{84}$$

Since the reference function is l-strongly convex, we can bound the dual update by the gradient distance in the following way

$$\|\kappa_{t+1} - \kappa_t\| \leq \frac{\eta}{l}\|\big(\nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\big)\| \leq \frac{\eta\|d_t\|}{l}. \tag{85}$$

Thus, we can have the following inequality for the last term of Eqn (82) by summing it up over the frame R, which is

$$\begin{aligned}
\langle \kappa_{t+k} - \kappa_t, A_{t+k}x_{t+k}' - z_{t+k}'\rangle \leq &\Big\|\sum_{j=t}^{t+k-1}\big(\kappa_{j+1} - \kappa_j\big)\Big\| \cdot \big\|A_{t+k}x_{t+k}' - z_{t+k}'\big\| \\
\leq &\frac{\eta}{l} \cdot \Big(\sum_{j=t}^{t+k-1}\big\|A_j x_j - z_j\big\|\Big) \cdot Z \leq (k-1) \cdot \frac{\eta Z^2}{l}.
\end{aligned} \tag{86}$$

Then within the frame of size R, we have

$$\left[\sum_{t=kR+1}^{kR+R}\kappa_t \cdot (A_t x_t' - z_t')\right] \leq \kappa_{kR+1}\left[\sum_{t=kR+1}^{kR+R}(A_t x_t' - z_t')\right] + \frac{R(R-1)}{2l}\eta Z^2. \tag{87}$$

Then we finish the proof by substituting Eqn (87) to Eqn (82) and summing up over the whole episode T. □

Lemma D.4. *Given any context sequence $\gamma = \{(f_t(\cdot), A_t)|\forall t \in [1,T]\}$, $x_{1:T}$ and $x^*_{1:T}$ are the action sequence of* `FairOBD` *and the (R,δ)-optimal benchmark, respectively. Suppose the switching cost $d(x_t, x_{t-1}) = \frac{\beta_1}{2}\|x_t - x_{t-1}\|^2$, then the optimization objective of* `FairOBD` *is bounded by*

$$\begin{aligned} G(x_{1:T}, x_{0:T-1}) \leq & G(x^*_{1:T}, x_{0:T-1}) - \frac{m+\lambda_1\beta_1+\lambda_2}{2}\frac{1}{T}\sum_{t=1}^{T}\|x^*_t - x_t\|^2 \\ & + \frac{\eta}{2l}Z^2R + Z\|\kappa_1\| + \frac{L\delta}{T} + \beta_2 L\sqrt{\frac{1}{T}(\frac{Z^2}{l^2} + 2\frac{LZ}{\eta l} + \frac{2}{\eta^2 l}Z\|\kappa_1\|)} \end{aligned} \tag{88}$$

Proof. With Lemma D.2 and Lemma D.3, we are ready to prove the Lemma D.4. Now suppose $x^*_{1:T}$ is the (R,δ)-optimal action sequence, then within the k-th frame we manually construct z^*_t as

$$z^*_t = \frac{1}{R}\sum_{i=kR+1}^{kR+R} A_i x^*_i,\ \forall t \in [kR+1, kR+R], 1 \leq k \leq K \tag{89}$$

Then according to Lemma D.3, we have

$$\begin{aligned} & \frac{V_h(\kappa_1, \kappa_{T+1})}{\eta} + \sum_{t=1}^{T}\Big[f_t(x_t) + \lambda_1 d(x_t, x_{t-1}) + g(z_t)\Big] + \sum_{t=1}^{T}\frac{\lambda_2}{2}\|x_t - v_t\|^2 \\ \leq & \sum_{t=1}^{T}\Big[f_t(x^*_t) + \lambda_1 d(x^*_t, x_{t-1}) + \frac{\lambda_2}{2}\|x^*_t - v_t\|^2 + g(z^*_t)\Big] \\ & - \frac{m+\lambda_1\beta_1+\lambda_2}{2}\sum_{t=1}^{T}\|x^*_t - x_t\|^2 + \frac{\eta}{2l}Z^2TR - \kappa_1^\top \cdot \Big[\sum_{t=1}^{T}A_t x_t - z_t\Big] \end{aligned} \tag{90}$$

Besides, based on the L-Lipschitz assumption on function $g(\cdot)$, we have

$$\begin{aligned} \sum_{t=1}^{T} g(z^*_t) = & \sum_{k=1}^{K} R \cdot g(\frac{1}{R}\sum_{t=(k-1)R+1}^{kR} A_t x^*_t) \\ \leq & R\sum_{k=1}^{K} g(\frac{1}{T}\sum_{t=1}^{T} A_t x^*_t) + L\sum_{k=1}^{K}\|\frac{1}{R}\sum_{t=(k-1)R+1}^{kR} A_t x^*_t - \frac{1}{T}\sum_{t=1}^{T} A_t x^*_t\| \\ \leq & \sum_{k=1}^{K}\Big[R \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x^*_t)\Big] + L\delta \\ = & Tg(\frac{1}{T}\sum_{t=1}^{T} A_t x^*_t) + L\delta \end{aligned} \tag{91}$$

Substituting it back to Eqn (90), we have

$$\begin{aligned} & \sum_{t=1}^{T}\Big[f_t(x_t) + \lambda_1 d(x_t, x_{t-1}) + \frac{\lambda_2}{2}\|x_t - v_t\|^2 + g(z_t)\Big] \\ \leq & \sum_{t=1}^{T}\Big[f_t(x^*_t) + \lambda_1 d(x^*_t, x_{t-1}) + \frac{\lambda_2}{2}\|x^*_t - v_t\|^2\Big] + T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x^*_t) \\ & - \frac{m+\lambda_1\beta_1+\lambda_2}{2}\sum_{t=1}^{T}\|x^*_t - x_t\|^2 + \frac{\eta}{2l}Z^2TR + L\delta - \frac{V_h(\kappa_1, \kappa_{T+1})}{\eta} - \kappa_1^\top \cdot \Big[\sum_{t=1}^{T}A_t x_t - z_t\Big] \end{aligned} \tag{92}$$

where the Bregman divergence $V_h(\kappa_1, \kappa_{T+1})$ is non-negative since the reference function is convex, which can be eliminated later. The last step is to upper bound $g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t)$ with $\frac{1}{T}\sum_{t=1}^{T} g(z_t)$,

which is

$$\begin{aligned}
&T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t) - \sum_{t=1}^{T} g(z_t) \\
\leq & TL\|\frac{1}{T}\sum_{t=1}^{T} A_t x_t - \frac{1}{T}\sum_{t=1}^{T} z_t\| + T\Big[g(\frac{1}{T}\sum_{t=1}^{T} z_t) - \frac{1}{T}\sum_{t=1}^{T} g(z_t)\Big] \\
\leq & L\|\sum_{t=1}^{T} A_t x_t - \sum_{t=1}^{T} z_t\|
\end{aligned} \tag{93}$$

Since κ_{t+1} is the minimizer of $e(\kappa)$, we have

$$d_t + \frac{1}{\eta}\big(\nabla h(\kappa_{t+1}) - \nabla h(\kappa_t)\big) = 0. \tag{94}$$

In other words

$$\Big\|\sum_{t=1}^{T}(-d_t)\Big\| = \Big\|\sum_{t=1}^{T}\big(A_t x_t - z_t\big)\Big\| = \frac{\|\nabla h(\kappa_{T+1}) - \nabla h(\kappa_1)\|}{\eta} \tag{95}$$

Since the reference function $h(\cdot)$ is l-strongly convex and β_2-smooth, we have

$$\frac{\|\nabla h(\kappa_{T+1}) - \nabla h(\kappa_1)\|}{\eta} \leq \frac{\beta_2\|\kappa_{T+1} - \kappa_1\|}{\eta} \leq \frac{\beta_2}{\eta}\sqrt{\frac{2}{l}V_h(\kappa_1, \kappa_{T+1})} \tag{96}$$

By substituting the inequality to Eqn (93), we have

$$T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t) - \sum_{t=1}^{T} g(z_t) \leq \frac{\beta_2 L}{\eta}\sqrt{\frac{2}{l}V_h(\kappa_1, \kappa_{T+1})} \tag{97}$$

By substituting Lemma D.2 into the inequality, we have

$$T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t) - \sum_{t=1}^{T} g(z_t) \leq \beta_2 L\sqrt{T(\frac{Z^2}{l^2} + \frac{1}{\eta}\frac{2LZ}{l}) - \frac{2}{\eta^2 l}\kappa_1^\top \cdot (\sum_{t=1}^{T} A_t x_t - z_t)} \tag{98}$$

According to our assumption, we have $\Big|\kappa_1^\top \cdot (\sum_{t=1}^{T} A_t x_t - z_t)\Big| \leq TZ \cdot \|\kappa_1\|$. By substituting the above inequality into Eqn (92), we have

$$\begin{aligned}
&\sum_{t=1}^{T}\Big[f_t(x_t) + \lambda_1 d(x_t, x_{t-1}) + \frac{\lambda_2}{2}\|x_t - v_t\|^2\Big] + T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t) \\
\leq &\sum_{t=1}^{T}\Big[f_t(x_t^*) + \lambda_1 d(x_t^*, x_{t-1}) + \frac{\lambda_2}{2}\|x_t^* - v_t\|^2\Big] + T \cdot g(\frac{1}{T}\sum_{t=1}^{T} A_t x_t^*) + \frac{\eta}{2l}Z^2 TR \\
&+ TZ\|\kappa_1\| + L\delta - \frac{m + \lambda_1\beta_1 + \lambda_2}{2}\sum_{t=1}^{T}\|x_t^* - x_t\|^2 + \beta_2 L\sqrt{T\Big(\frac{Z^2}{l^2} + \frac{1}{\eta}\frac{2LZ}{l} + \frac{2}{\eta^2 l}Z\|\kappa_1\|\Big)}
\end{aligned} \tag{99}$$

By dividing both sides by T, we finish the proof.

□

Now we are ready to prove Theorem 5.2 as the following.

Proof. According to Lemma D.4, the switching cost in right hand side in Eqn (88) is evaluated on the actual action sequence, while the ultimate goal is to bound it with the switching cost of offline optimal action sequence. In other words, the final step is to convert $G(x_{1:T}^*, x_{0:T-1})$ to $G(x_{1:T}^*, x_{0:T-1}^*)$. More specifically,

$$
\begin{aligned}
&\lambda_1 d(x_t, x_{t-1}) = \frac{\lambda_1\beta_1}{2}\sum_{t=1}^{T}\|x_t^* - x_{t-1}\|^2 \\
\leq &\frac{\lambda_1\beta_1}{2}\sum_{t=1}^{T}\Big(\|x_t^* - x_{t-1}^*\| + \|x_{t-1}^* - x_{t-1}\|\Big)^2 \\
\leq &\frac{\lambda_1\beta_1}{2}(\frac{m+\lambda_2+\lambda_1\beta_1}{m+\lambda_2})\sum_{t=1}^{T}\|x_t^* - x_{t-1}^*\|^2 + \frac{m+\lambda_1\beta_1+\lambda_2}{2}\sum_{t=1}^{T}\|x_{t-1}^* - x_{t-1}\|^2 \\
\leq &\frac{\lambda_1\beta_1}{2}(\frac{m+\lambda_2+\lambda_1\beta_1}{m+\lambda_2})\sum_{t=1}^{T}\|x_t^* - x_{t-1}^*\|^2 + \frac{m+\lambda_1\beta_1+\lambda_2}{2}\sum_{t=1}^{T}\|x_t^* - x_t\|^2
\end{aligned}
\tag{100}
$$

Substitute Eqn (100) into Eqn (88), we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\Big[f_t(x_t) + \lambda_1 d(x_t, x_{t-1}) + \frac{\lambda_2}{2}\|x_t - v_t\|^2\Big] + g(\frac{1}{T}\sum_{t=1}^{T}A_t x_t) \\
\leq &\frac{1}{T}\sum_{t=1}^{T}\Big[f_t(x_t^*) + \lambda_1(\frac{m+\lambda_2+\lambda_1\beta_1}{m+\lambda_2})d(x_t^*, x_{t-1}^*) + \frac{\lambda_2}{2}\|x_t^* - v_t\|^2\Big] + g(\frac{1}{T}\sum_{t=1}^{T}A_t x_t^*) + 0 + \Delta \\
\leq &\frac{1}{T}\sum_{t=1}^{T}\Big[(1+\frac{\lambda_2}{m})f_t(x_t^*) + \lambda_1(\frac{m+\lambda_2+\lambda_1\beta_1}{m+\lambda_2})d(x_t^*, x_{t-1}^*)\Big] + g(\frac{1}{T}\sum_{t=1}^{T}A_t x_t^*) + \Delta
\end{aligned}
\tag{101}
$$

where $\Delta = \frac{\eta}{2l}Z^2R + \beta_2 L\sqrt{\frac{1}{T}(\frac{Z^2}{l^2} + 2\frac{LZ}{\eta l} + \frac{2}{\eta^2 l}Z\|\kappa_1\|)} + Z\|\kappa_1\| + \frac{L\delta}{T}$. Then the overall cost of `FairOBD` is bounded by

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\Big[f_t(x_t) + d(x_t, x_{t-1})\Big] + g(\frac{1}{T}\sum_{t=1}^{T}A_t x_t) \\
\leq &\frac{1}{T\lambda_1}\sum_{t=1}^{T}\Big[f_t(x_t) + \lambda_1 d(x_t, x_{t-1})\Big] + \frac{1}{\lambda_1}g(\frac{1}{T}\sum_{t=1}^{T}A_t x_t) + \frac{1}{\lambda_1}\Delta \\
\leq &\frac{1}{T}\sum_{t=1}^{T}\Big[(\frac{m+\lambda_2}{m\lambda_1})f_t(x_t^*) + (\frac{m+\lambda_2+\lambda_1\beta_1}{m+\lambda_2})d(x_t^*, x_{t-1}^*)\Big] + \frac{1}{\lambda_1}g(\frac{1}{T}\sum_{t=1}^{T}A_t x_t^*) + \frac{1}{\lambda_1}\Delta \\
\leq &C\left[\frac{1}{T}\sum_{t=1}^{T}\Big[f_t(x_t^*) + d(x_t^*, x_{t-1}^*)\Big] + g(\frac{1}{T}\sum_{t=1}^{T}A_t x_t^*)\right] + \frac{1}{\lambda_1}\Delta
\end{aligned}
\tag{102}
$$

where $C = \max\{\frac{m+\lambda_2}{m\lambda_1}, \frac{m+\lambda_2+\lambda_1\beta_1}{m+\lambda_2}\}$ is the competitive ratio. □

Reference

[57] Santiago Balseiro, Haihao Lu, and Vahab Mirrokni. Regularized online allocation problems: Fairness and beyond. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 630–639. PMLR, 18–24 Jul 2021.

[58] 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. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[59] Yuelin Han, Zhifeng Wu, Pengfei Li, Adam Wierman, and Shaolei Ren. The unpaid toll: Quantifying the public health impact of ai. *arXiv preprint arXiv:2412.06288*, 2024.

[60] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[61] Microsoft. Microsoft Local Communities. `https://local.microsoft.com/communities/`.

[62] Microsoft. Microsoft Data Centers Sustainability - Efficiency. `https://datacenters.microsoft.com/sustainability/efficiency/`, 2023.

[63] U.S. Energy Information Administration (EIA). EIA OPEN DATA. `https://www.eia.gov/opendata/`, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include some discussion about the limitations of the work in our paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All the assumptions are clearly listed in the main paper. A complete and correct proof is attached in the appendix and ready for review.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully disclose all the information needed to reproduce the main experimental results of the paper. Due to the page limit, the detailed information can be found in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: In our empirical study, all information is based on publicly available data. We will make the data and code public upon acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: As we don't need to train We provide all the information about the test details

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide the error bars for our experiment in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Due to the theoretical nature of our work, we do not foresee too much negative societal impacts or the need for safeguards. Our discussion mainly focuses on the positive broader impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Due to the theoretical nature of this work, it doesn't have such risks for data or model misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: Due to the theoretical nature of this work, the paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Due to the theoretical nature of this work, the paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Due to the theoretical nature of this work, the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Due to the theoretical nature of this work, the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.