Smoothed Online Optimization for Target Tracking: Robust and Learning-Augmented Algorithms

Ali Zeynali
University of Massachusetts Amherst
azeynali@cs.umass.edu

Mahsa Sahebdel
University of Massachusetts Amherst
msahebdelala@umass.edu

Qingsong Liu
University of Massachusetts Amherst
qingsongliu@umass.edu

Mohammad Hajiesmaili
University of Massachusetts Amherst
hajiesmaili@cs.umass.edu

Ramesh K. Sitaraman
University of Massachusetts Amherst & Akamai Technologies
ramesh@cs.umass.edu

Abstract

We introduce the *Smoothed Online Optimization for Target Tracking* (SOOTT) problem, a new framework that integrates three key objectives in online decision-making under uncertainty: (1) *tracking cost* for following a dynamically moving target, (2) *adversarial perturbation cost* for withstanding unpredictable disturbances, and (3) *switching cost* for penalizing abrupt changes in decisions. This formulation captures real-world scenarios such as elastic and inelastic workload scheduling in AI clusters, where operators must balance long-term service-level agreements (e.g., LLM training) against sudden demand spikes (e.g., real-time inference). We first present BEST, a robust algorithm with provable competitive guarantees for SOOTT. To enhance practical performance, we introduce CoRT, a learning-augmented variant that incorporates untrusted black-box predictions (e.g., from ML models) into its decision process. Our theoretical analysis shows that CoRT strictly improves over BEST when predictions are accurate, while maintaining robustness under arbitrary prediction errors. We validate our approach through a case study on workload scheduling, demonstrating that both algorithms effectively balance trajectory tracking, decision smoothness, and resilience to external disturbances.

1 Introduction

This paper introduces and studies the *Smoothed Online Optimization for Target Tracking* problem (SOOTT), a new framework that captures three interacting objectives in online target tracking. At each round, an agent selects an action, evaluated based on the alignment of a windowed average of its recent actions with a dynamically moving target. The agent incurs three types of costs: (1) a tracking cost penalizing deviations between the agent's time-averaged action and a desired but dynamically moving target, (2) an adversarial perturbation cost reflecting external disturbances that is unpredictable and arriving online, and (3) a switching cost that penalizes abrupt changes in the agent's decisions. Together, these components form a composite loss that challenges conventional online optimization techniques by introducing dependencies on both historical behavior and adversarial adjustments. Effective minimization of this loss requires algorithms capable of balancing smooth trajectory alignment, smoothness in decision-making, and resilience to adversarial disturbances.

Preprint. Under review.

A key motivational application for studying SOOTT arises from the need to efficiently manage the scheduling of *elastic* workloads (e.g., AI training) and *inelastic* workloads (e.g., AI inference) in large-scale cloud and AI clusters [6, 27]. . In such environments, operators must simultaneously maintain target processing rates for long-running elastic jobs to meet the customer service-level agreements (SLAs) while accommodating unpredictable spikes in latency-sensitive inelastic workload. This dual demand requires dynamic, on-the-fly resource reallocation, where elastic jobs (e.g., LLM training, finance analysis, software maintenance, and upgrade) can be paused or throttled to prioritize inelastic jobs (e.g., real-time AI inference). At each decision epoch and *before the realization of the inelastic demand*, operators must judiciously determine what fraction of resources to allocate to inelastic jobs, leaving the remainder for elastic ones. Over-allocating (allocating resources more than the realized demand) to inelastic jobs risks leaving resources idle and failing to meet the SLA of elastic jobs, while under-allocating can result in unserved inference requests [40, 19]. Additionally, this flexibility in resource allocation between elastic and inelastic workload comes at a cost: frequent pause and resuming multi-hundred-gigabyte training workloads impose heavy checkpoint-and-restore penalties, making reckless preemption highly counterproductive [25, 26], and SOOTT captures this by adding the switching cost terms. This motivates our SOOTT framework, which captures these trade-offs explicitly: (1) the sliding window tracking term that models long-term SLA requirement for elastic jobs over time, (2) the adversarial perturbation term represents bursty or unpredictable demands in inelastic jobs that can only be observed *after* resource allocation; and (3) the switching cost accounts for the overhead of frequent changes in resource (re)allocation.

Beyond the main case study of the elastic and inelastic workload scheduling, SOOTT is a general framework that is well-suited for broader range of applications, e.g., server maintenance scheduling [22, 32], where consistent service requires regular interventions over a time window; image-based object tracking [8, 44, 15], where predictions must remain coherent across frames, online control [53, 55] where system stability and performance depend on sequences of past inputs, and resource pooling in shared infrastructures, e.g., in multi-tenant cloud platforms and shared Electric Vehicle (EV) charging platforms [38]. In resource pooling of shared infrastructures, operators dynamically allocate shared resources—such as compute, bandwidth, or energy—among multiple users or applications with varying demand profiles and SLA requirements. The challenge is to maintain fair and efficient resource distribution in the presence of unpredictable and non-stochastic workload. Our model naturally captures the need for smooth adjustments while mitigating abrupt disruptions in the allocations.

On the theory front, our framework brings together two well-studied strands of online optimization for target tracking that have so far evolved largely in parallel: (1) memory-based online tracking, where past actions influence current tracking cost [34, 35, 53]; and (2) smoothed online optimization [5, 43, 53] which penalizes abrupt changes in decision-making. We provide a comprehensive review of the related literature in the Appendix §A and highlight how existing algorithms fail to solve our problem holistically. Specifically, existing methods either neglect the memory-based dynamics introduced by the sliding window tracking term or significantly simplify them, or overlook the role of the smoothness component. In this paper, we develop algorithms for SOOTT under competitive worst-case analysis (i.e., without assuming any predictions of adversarial perturbations and dynamic target) and aim to develop algorithms that achieve a solid constant *competitive ratio*, defined as the worst-case ratio between the cost of an online algorithm and the offline optimum [7, 39].

While worst-case guarantees offer robustness, they may be overly conservative or cautious. In recent years, learning-augmented online algorithms [37, 42] have emerged to use potentially imperfect predictions to achieve two goals: performing near-optimally when the predictions are accurate (i.e., *consistency*) and retaining worst-case guarantees when predictions are misleading (i.e., *robustness*). These algorithms bridge the gap between worst-case guarantees and practical performance by incorporating untrusted predictions. However, applying this to our setting introduces unique challenges. Unlike classical online models where predictions are straightforward (e.g., demand or price forecasts), the interplay between sliding-window tracking, adversarial perturbation, and switching costs creates complex interdependencies. As a second goal of this paper, we aim to propose learning-augmented algorithms that effectively integrate machine-learned advice to enhance practical performance while retaining robustness against erroneous predictions.

Main contributions. We study the problem of smoothed online optimization for target tracking, denoted as SOOTT, where the objective is to minimize a cost function including three components: (1) tracking cost, (2) adversarial cost, and (3) switching cost. We provide both robust and learning-augmented algorithms for SOOTT and the key technical contributions are summarized below.

Competitive analysis. We begin by presenting IGA, a semi-online algorithm that has access to the adversary's exact target for the current time step, but not for future. Through a competitive analysis, we establish a constant upper bound on its competitive ratio. Building on this, we propose BEST, a fully online algorithm for SOOTT, and analyze its worst-case performance by bounding its cost relative to that of IGA. Furthermore, we demonstrate the tightness of our competitive guarantees by showing that it recovers the best-known results in relevant special cases [43, 16].

Learning-augmented analysis. To improve performance beyond worst-case guarantees, we consider the learning-augmented setting. We begin with a natural baseline algorithm that fully trusts predictions; while it performs near-optimally with perfect predictions, it is fragile under adversarial noise and lacks robustness in such cases. To address the lack of robustness, we propose CoRT, a robust learning-augmented algorithm that leverages predictions to improve over BEST when they are accurate, while still retaining competitive guarantees under worst-case conditions. Our analysis reveals a fundamental trade-off in CoRT between its consistency and robustness, which can be tuned via a controllable algorithmic parameter.

Case study. Using real-world traces from Google Cloud [17], we empirically evaluate our algorithms through a case study on dynamic resource allocation for both elastic and inelastic workloads. Notably, our experiments demonstrate that the performance of the CoRT algorithm closely approaches that of IGA —our proposed semi-online but impractical algorithm that assumes perfect knowledge of online inputs—while also maintaining robustness against arbitrarily inaccurate predictions.

Technical novelty. Our analysis builds on two new ingredients: (1) we leverage the fact that the auxiliary objective $g_t(u)$(Lemma B.3) is strongly convex, and together with a Lipschitz-stability result for the windowed minimiser (Lemma B.2), to achieve a *two-level contraction* that simultaneously reduces the prediction gap and the accumulated history error. (2) We develop a bespoke sliding-window Cauchy–Schwarz lemma (Lemma B.4) to convert convoluted memory sums into point-wise norms while preserving tight constants. These tools drive the tight bounds for BEST and the consistency–robustness guarantee of our learning-augmented CoRT algorithm.

2 Problem Formulation

Model and problem statement. We consider the problem of *smoothed online optimization for target tracking* (SOOTT) where an agent chooses an action at each time step under an adversarial perturbation setting. At each time $t \in \mathbb{N}$, a trajectory target point $\tau_t \in \mathbb{R}^d$ and a time-varying adversarial perturbation function $f_t : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ are revealed to the agent. Meanwhile, the adversary selects a hidden target $u_t \in \mathbb{R}^d$, which is disclosed only after the agent has chosen action $x_t \in \mathbb{D} \subset \mathbb{R}^d$, where $\mathbb{D}$ is a compact set representing the domain of valid actions. The agent then incurs the following cost:

$$\text{Cost}_t(x_t, h_t) = \underbrace{\left\| \frac{x_t + h_t}{w+1} - \tau_t \right\|^2}_{\text{tracking cost}} + \underbrace{\lambda_1 f_t(x_t - u_t)}_{\text{adversarial cost}} + \underbrace{\lambda_2 \|x_t - x_{t-1}\|^2}_{\text{switching cost}}, \tag{1}$$

where $h_t = \sum_{i=1}^{w} x_{t-i}$ represents the aggregation of the agent's past w actions, and $\lambda_1, \lambda_2 > 0$ are fixed weighting parameters. The goal of the agent is to select actions that minimize the cumulative cost over T time steps: $\sum_{t=1}^{T} \text{Cost}_t(x_t, h_t)$.

This cost function captures three competing objectives. The first term penalizes deviations between the agent's time-averaged action over the current and past w rounds and a desired trajectory target τ_t, thereby encouraging tracking the moving target. The second term, $\lambda_1 f_t(x_t - u_t)$, reflects the adversarial influence and penalizes discrepancies between the agent's action and the (hidden) target of the adversary, u_t, through a function f_t. The third term, $\lambda_2 \|x_t - x_{t-1}\|^2$, imposes a regularization that discourages abrupt changes in the agent's behavior over consecutive slots, promoting smoothness in the sequence of actions. The trade-offs between these objectives are governed by the parameters λ_1 and λ_2.

To enable tractable analysis, we impose the following standard structural assumptions on the adversarial perturbation and initialization:

Assumption 1 (Adversarial Minimum)**.** *The time-dependent adversarial perturbation function $f_t(\cdot)$ is non-negative and minimized at the origin without loss of generality, i.e.,* $\arg\min f_t(x) = \mathbf{0}$.

Assumption 2 (Adversarial Convexity)**.** *Function $f_t(\cdot)$ is m-strongly convex for some $m > 0$.*

Assumption 3 (Adversarial Smoothness)**.** *Function $f_t(\cdot)$ is ℓ-smooth, meaning its gradient is ℓ-Lipschitz continuous for some $\ell > 0$.*

Assumption 4 (Initial Convergence)**.** *For all $t \leq 0$, the agent's actions and the adversary's targets are both initialized at the origin. Additionally, the online algorithm coincides with the offline optimal strategy over this initial period.*

These assumptions are standard online optimization literature and allow for meaningful theoretical analysis [43, 51, 31, 56]. Assumption 1 ensures that the adversarial cost cannot reward the agent through negative values and is minimized when the agent exactly matches the target of the adversary. Assumptions 2 and 3 impose structure to the adversarial perturbations, enabling gradient-based analysis. Finally, Assumption 4 provides a synchronized and consistent initialization for the online optimization process.

Challenges. A major key challenge in SOOTT arises from the presence of memory, i.e., term h_t, and x_{t-1}, that includes the historical actions, in the cost function, which introduces temporal coupling across decisions. Specifically, the agent's current cost depends not only on its present action but also on a window of past actions. Prior work [43, 16, 9] has demonstrated that, even in an idealized setting where the agent has full knowledge of the target of the adversary, u_t, before committing to an action, identifying the optimal decision remains nontrivial due to this memory dependency. Notably, when the influence of memory is limited—e.g., when the memory window w and the smoothness regularization coefficient λ_2 are sufficiently small—the problem becomes effectively myopic. In such cases, a greedy strategy that minimizes the instantaneous cost can closely approximate the optimal offline policy which has a full knowledge of future input, i.e., $\{\tau_t,\ u_t\}_{t=1}^T$, and adversarial cost functions $\{f_t\}_{t=1}^T$. The second challenge in SOOTT stems from the fact that the target of the adversary u_t is revealed only after the agent has selected x_t. Thus, the adversarial cost term is not directly observable at the time of decision, which complicates the design of algorithms with guaranteed performance.

Competitive analysis. Our goal is to design an online algorithm that guarantees a small competitive ratio [7, 39] which guarantees performing near optimal offline algorithm. Formally, for an online algorithm ALG and an input instance $\mathcal{I}$, the competitive ratio is defined as: $\texttt{CR}(ALG) = \sup_{\mathcal{I}} \text{Cost}(ALG, \mathcal{I})/\text{Cost}(OPT, \mathcal{I})$, where Cost$(ALG, \mathcal{I})$, and Cost$(OPT, \mathcal{I})$ denote the cost of algorithm ALG and offline optimum on instance $\mathcal{I}$. In addition, to further simplify the presentation of theoretical bounds, in this paper we use the *degradation factor* (DF) metric, introduced in [50], to bound the worst-case ratio between the performance of two online algorithms. Specifically, the degradation factor of algorithm ALG_1 relative to another algorithm ALG_2 is defined as: $\texttt{DF}(ALG_1, ALG_2) = \sup_{\mathcal{I}} \text{Cost}(ALG_1, \mathcal{I})/\text{Cost}(ALG_2, \mathcal{I})$, which also implies an upper bound on the competitive ratio of ALG_1 in terms of that of ALG_2: $\texttt{CR}(ALG_1) \leq \texttt{DF}(ALG_1, ALG_2) \cdot \texttt{CR}(ALG_2)$. When ALG_2 is the optimal offline algorithm, the degradation factor coincides with the competitive ratio of ALG_1.

3 Robust Online Algorithms for SOOTT

In this section, we introduce IGA, a semi-online benchmark algorithm that relaxes the uncertainty of u_t by assuming that the adversary's target u_t is known at the current time step, but remains unknown for future time steps. Although this assumption is unrealistic in most practical scenarios, IGA plays an important analytical role, serving as a performance baseline against which we compare more practical algorithms that do not have access to this information. Then, in Section 3.2, we present BEST, a fully online algorithm that operates without knowledge of the adversary's target and analyze its performance by bounding its degradation factor relative to IGA.

3.1 IGA: A Semi-online Benchmark Algorithm with Exact Knowledge of u_t

This section introduces Informed Greedy Algorithm (IGA), which known u_t when taking its action. Given this additional information, IGA selects actions that greedily minimize the cost function at each time step. This setting captures an idealized scenario in which the adversary's intention is perfectly predictable and the cost structure is fully known in advance. Although such assumptions may not hold in practice, the performance of IGA offers a meaningful baseline to assess the quality of practical online algorithms.

At each time step t, IGA observes the target of the adversary u_t and chooses an action x_t that minimizes the total cost over the current time step, balancing target tracking, adversarial deviation, and switching penalties. The pseudo-code of IGA is presented in Algorithm 1.

Algorithm 1: The Informed Greedy Algorithm (IGA)

Data: $\hat{x}_{t-w:t-1}$, u_t, τ_t
Result: $\hat{x}_t$: action of the IGA at time t

$\hat{x}_t \leftarrow \text{argmin}_x \, ||\frac{x+\hat{h}_t}{w+1} - \tau_t||^2 + \lambda_1 \, f_t(x - u_t) + \lambda_2||x - \hat{x}_{t-1}||^2$

Output: $\hat{x}_t$

Since IGA has the full knowledge of the cost function at time step t, it can select the action that minimizes the cost at that time step, given the current history h_t. However, as it lacks foresight into future target values τ_t and target of the adversary u_t, its chosen actions may still diverge from those of the optimal offline solution. The following result establishes a performance guarantee for IGA in terms of its competitive ratio, evaluated against the cost incurred by the optimal offline strategy.

Theorem 3.1. *If $2w^2 < 2 + m\lambda_1(w+1)^2$, the competitive ratio of* IGA *is upper bounded by*

$$\texttt{CR}(\texttt{IGA}) \leq 1 + \frac{2(\lambda_2 \, (w+1)^2 + w^2)}{m\lambda_1(w+1)^2 + 2 - 2w^2}. \tag{2}$$

The proof of Theorem 3.1 is given in Appendix §B.1. When both λ_1 and λ_2 are large, the setting reduces to standard smoothed online convex optimization [16, 43, 28], and the competitive ratio of IGA converges to $1 + \frac{2\lambda_2}{m\lambda_1}$, consistent with results in the literature [43, 16]. When $w > 0$, the optimal offline algorithm considers future consequences of current actions, while IGA makes locally optimal decisions using perfect knowledge of u_t. In such cases, when λ_1 is small or f_t is weakly convex, the impact of adversarial cost is diminished, and IGA may perform arbitrarily worse than the offline optimum—hence the necessity of the condition in Theorem 3.1 to ensure bounded competitive ratio.

In the remainder of the paper, we develop online algorithms without perfect information of u_t and assess their performance using the *degradation factor* metric with respect to IGA. This allows us to derive meaningful performance guarantees relative to the offline optimum by combining the bounds on the degradation factor with the result of Theorem 3.1.

3.2 BEST: A Robust Algorithm for SOOTT

We present Backward Evaluation for Sequential Targeting (BEST), an online algorithm for SOOTT that does not require any knowledge of u_t in the current and future time step. Since online algorithms lack exact information about the adversary's target, a naive greedy approach (which is also blind to u_t) that minimizes the cost at each time step independently can diverge significantly from the behavior of the IGA, leading to substantially higher cumulative costs. Our algorithm is designed to keep its actions close to the actions of the IGA by considering the history of IGA during its action selection process. BEST ignores the adversarial cost term in its own historical actions, and selects its action based on the history of IGA. Note that the history of IGA is accessible to BEST since, after observing the target of the adversary in each time step, one could exactly calculate the corresponding action of IGA. We present the pseudo-code of BEST in Algorithm 2.

Algorithm 2: Backward Evaluation for Sequential Targeting (BEST)

Data: u_{t-1}, τ_t, $\hat{x}_{t-w-1:t-2}$: history of actions taken by IGA
Result: x_t: action of the agent at time t

1 $\hat{x}_{t-1} \leftarrow$ action of IGA at time step $(t-1)$

2 $x_t \quad \leftarrow \text{argmin}_x \, ||\frac{x+\hat{h}_t}{w+1} - \tau_t||^2 + \lambda_2||x - \hat{x}_{t-1}||^2$

3 **Output:** x_t

In Line 1, BEST finds the action of IGA in the previous time step, $\hat{x}_{t-1}$, since the most recent target of the adversary has already been revealed. It keeps track of the history of action taken by IGA and evaluates $\hat{x}_{t-1}$, and $\hat{h}_t$ based on IGA's past actions. Next, in Line 2, BEST observes the current trajectory target τ_t and selects its action by ignoring the adversarial cost term and assuming its history matches that of IGA. Note that if the target of the adversary at time step t, is different from x_t, the action of BEST, x_t, and the action of IGA, $\hat{x}_t$ will be different. Due to this difference, the cost value incurred by IGA at time step t would be proportional to $||x_t - \hat{x}_t||^2$ since all terms in the cost function are convex and smooth. This insight shows how BEST keeps its cost values close to the cost of IGA algorithm which formally analyzed in the following Theorem. The following Theorem shows that BEST achieves a bounded degradation factor with respect to IGA proving its worst-case performance guarantee when combined with the result of Theorem 3.1.

Theorem 3.2. *The degradation factor of* BEST *with respect to* IGA *is bounded as follows:*

$$\texttt{DF}(\texttt{BEST}, \texttt{IGA}) \leq 1 + \frac{\ell}{m} \cdot \frac{\eta^2 + 2\lambda_1 \ell (1 + \lambda_2)}{\eta(\eta - m\lambda_1)}. \tag{3}$$

where $\eta = 2/(w+1)^2 + m\,\lambda_1 + 2\lambda_2$.

The complete proof of Theorem 3.2 is provided in Appendix §B.2. As a sketch of the proof, we define an auxiliary function $g_t(u)$ that represents the cost incurred by IGA at time step t, assuming the adversary's target is u. We show that $g_t(u)$ is strongly convex, with its unique minimizer corresponding to the action selected by BEST. This structural property allows us to bound the cost difference between BEST and IGA in terms of the cost of IGA and problem-specific constants. Then, we choose the hyperparameters introduced in the analysis, to ensure that the additive constant term in the bound is negative, which guarantees that BEST achieves a constant degradation factor.

Remark 3.1. *The degradation factor of* BEST *relative to* IGA *grows at most as* $\mathcal{O}(m\lambda_1)$ *with respect to* λ_1 *and* m*. This is intuitive, as increasing either parameter linearly amplifies the influence of the adversarial cost term in its objective. Since* BEST *does not account for this adversarial term in its action selection policy, its performance gap relative to* IGA *increases linearly with* λ_1 *and* m*.*

4 Learning-Augmented Algorithms for SOOTT

Learning-augmented online algorithms incorporate machine-learned predictions of future inputs to enhance classical online decision-making [37, 42]. In SOOTT, the algorithm receives a prediction of the adversary's target for the upcoming time step and integrates this estimate into its action selection strategy. While accurate predictions can significantly improve performance, blindly trusting erroneous predictions may lead to degraded outcomes, especially under high noise. To account for this, the performance of learning-augmented algorithms is typically evaluated using two complementary metrics: *consistency*, which captures performance under accurate predictions, and *robustness*, which measures resilience against arbitrary prediction errors. Achieving both objectives simultaneously is challenging, as improving consistency often comes at the expense of robustness, necessitating careful algorithmic design to manage this trade-off [42].

Let $\hat{u}_t$ denote the prediction of the adversary's target for time step t. As discussed in Section 3, we use IGA as a baseline to evaluate the performance of online algorithms. Based on this, we define the notions of consistency and robustness for the SOOTT setting as follows:

Definition 4.1. *A learning-augmented algorithm for* SOOTT *is* α*-consistent if its degradation factor relative to* IGA *is at most* α *under perfect predictions, i.e., when* $\hat{u}_t = u_t$ *for all t.*

Definition 4.2. *A learning-augmented algorithm for* SOOTT *is said to be* β*-robust if its degradation factor relative to* IGA *is at most* β *for any predicted sequence* $\{\hat{u}_t\}_{t=1}^T$.

In the following sections, we demonstrate that a greedy algorithm which selects actions to minimize the immediate cost achieves optimal consistency but lacks robustness. To overcome this limitation, we introduce a learning-augmented algorithm that incorporates a tunable parameter θ, allowing us to control the trade-off between consistency and robustness.

4.1 PGA: An Algorithm with Full Trust on the Prediction

In this section, we present the Prediction-based Greedy Algorithm (PGA), which greedily finds the action that is predicted to minimize the cost value during time step t. Given the predicted adversary's target $\hat{u}_t$ at time step t, PGA fully trusts the prediction and chooses the action that leads to the lowest cost for the current time step. The detail of PGA is provided in Algorithm 3.

Algorithm 3: Prediction-based Greedy Algorithm (PGA)

Data: $\tilde{x}_{t-w:t-1}$, $\hat{u}_t$, τ_t
Result: $\tilde{x}_t$: action of the agent at time t
$\tilde{x}_t \leftarrow \text{argmin}_x \, ||\frac{x+\tilde{h}_t}{w+1} - \tau_t||^2 + \lambda_1 \cdot f_t(x - \hat{u}_t) + \lambda_2 ||x - \tilde{x}_{t-1}||^2$
Output: $\tilde{x}_t$

Since PGA selects its actions by fully trusting the predicted adversary's target, its performance is highly sensitive to prediction errors. When the prediction is perfect, PGA takes the same actions as IGA, thereby achieving optimal consistency. However, with prediction errors, the cost incurred by PGA can deviate significantly from that of the optimal offline solution. The following theorem provides a lower bound on the degradation factor of PGA as a function of the prediction error in $\hat{u}_t$.

Theorem 4.3. *The degradation factor of* PGA *with respect to* IGA *is lower bounded as follows:*

$$\texttt{DF}(\texttt{PGA}, \texttt{IGA}) \geq \frac{m}{2 \, \max_t f_t(\boldsymbol{0})} \sum_{t=1}^{T} \frac{\|u_t - \hat{u}_t\|^2}{T}.$$

The full proof of Theorem 4.3 is provided in Appendix §B.3.

Remark 4.1. *Since* $\max_t f_t(\mathbf{0})$ *can be arbitrarily close to zero, and the prediction error* $\|u_t - \hat{u}_t\|$ *is unbounded, the cost of* PGA *can become arbitrarily large relative to* IGA *in the worst case. This demonstrates that* PGA *lacks robustness when faced with inaccurate predictions.*

Motivated by the lack of robustness in PGA, in what follows, we aim to design a learning-augmented algorithm that not only enhances the performance of BEST under perfect predictions but also maintains provable robustness guarantees under noisy or adversarial prediction errors.

4.2 CoRT: A Consistent and Robust Learning-Augmented Algorithm for SOOTT

We propose the Consistent and Robust Tracking algorithm (CoRT), which incorporates predictions of the adversary's target $\hat{u}_t$ while providing provable robustness guarantees (see Algorithm 4 for the pseudo-code). Like BEST, CoRT selects actions using the history of IGA. However, it accounts for the adversarial cost term by estimating it through a controlled target $\tilde{u}_t$, computed from $\hat{u}_t$ and constrained to lie within a distance of at most θD_t from BEST's action (Lines 2–5). Here, θ is a tunable algorithm parameter, and D_t is a dynamically adjusted bound. The algorithm initializes with $D_1 = 0$ and updates D_t based on its previous value, the deviation between u_t and BEST 's action, and the discrepancy between that action and $\tilde{u}_t$ (Line 7). Intuitively, CoRT adapts D_t to reflect the observed deviation of the actual adversary's target from BEST 's action, thereby bounding the cumulative deviation of $\tilde{u}_t$ from BEST 's action. See Figure 1 for an illustration. In the limiting case, CoRT recovers BEST as $\theta \to 0$.

Algorithm 4: Consistent and Robust Tracking Algorithm (CoRT)

Data: $\hat{u}_t$, τ_t, D_t, parameter θ, $\hat{x}_{t-w-1:t-2}$: history of actions taken by IGA
Result: $\tilde{x}_t$: action of the agent at time t
1 $x_t \leftarrow$ action of BEST at time t
2 $\tilde{u}_t \leftarrow \hat{u}_t$
3 **if** $||\hat{u}_t - x_t|| \geq \theta D_t$ **then**
4 $\quad \tilde{u}_t \leftarrow x_t + \theta D_t \cdot \frac{(\hat{u}_t - x_t)}{||\hat{u}_t - x_t||}$
5 **end**
6 $\tilde{x}_t \leftarrow \text{argmin}_x \left\|\frac{x+\hat{h}_t}{w+1} - \tau_t\right\|^2 + \lambda_1 f_t(x - \tilde{u}_t) + \lambda_2 \left\|x - \hat{x}_{t-1}\right\|^2$
7 $D_{t+1}^2 \leftarrow D_t^2 + ||u_t - x_t||^2 - \theta^{-2} ||\tilde{u}_t - x_t||^2$
8 **Output:** $\tilde{x}_t$

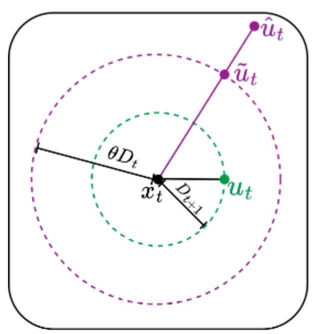


Figure 1: Actual vs. predicted targets for a time step, and the corresponding update of D_t.

Theorem 4.4. *Given parameter θ,* CoRT *is* $\texttt{DF}(\texttt{BEST}, \texttt{IGA})(1 + \theta^2 \mathcal{O}(1))$*-robust and* $\mathcal{C}$*-consistent where:*

$$\mathcal{C} \leq \psi(\theta) + (1 - \psi(\theta)) \cdot \texttt{DF}(\texttt{BEST}, \texttt{IGA}) + \frac{2\lambda_1 \lambda_2 \ell^2}{m\eta(\eta - m\lambda_1)} \cdot \frac{\theta^2}{1 + \theta^2}, \tag{4}$$

and $\psi(\theta) : \mathbb{R}_{\geq 0} \to [0, 1]$ is an increasing function satisfying $\psi(0) = 0$ and $\psi(\infty) = 1$.

The full proof of Theorem 4.4 is provided in Appendix §B.4. As a sketch, we first show that the cost incurred by CoRT deviates from that of BEST by at most a linear function of the aggregate deviation $\sum_t \|x_t - \tilde{x}_t\|^2$, where x_t and $\tilde{x}_t$ denote the actions of BEST and CoRT at time t, respectively. We then prove that this deviation is upper bounded by a factor proportional to θ^2 times the cost of BEST, establishing the robustness guarantee. Furthermore, under perfect predictions, to construct a challenging instance, an adversary must increase the separation between its own action and that of BEST over time. Also, in such condition, the action of CoRT at each time step is a convex combination of the actions of BEST and IGA. This linear relationship among the actions allows us to show that the cost of CoRT is a convex combination of the costs of BEST and IGA, up to a bounded additive error, which yields the consistency bound.

Remark 4.2. *Theorem 4.4 illustrates a trade-off in* CoRT *between its consistency and robustness, governed by the parameter θ. As θ increases, robustness improves at most quadratically, while the consistency decreases. In the limit as $\theta \to \infty$,* CoRT *achieves its best possible consistency but completely sacrifices robustness.*

5 Case Study: Resource Allocation for Elastic and Inelastic Workloads

We consider a case study involving resource allocation in cloud computing platforms handling both elastic and inelastic workloads. In this setting, we evaluate our proposed algorithms for SOOTT and compare them in average and adversarial scenarios.

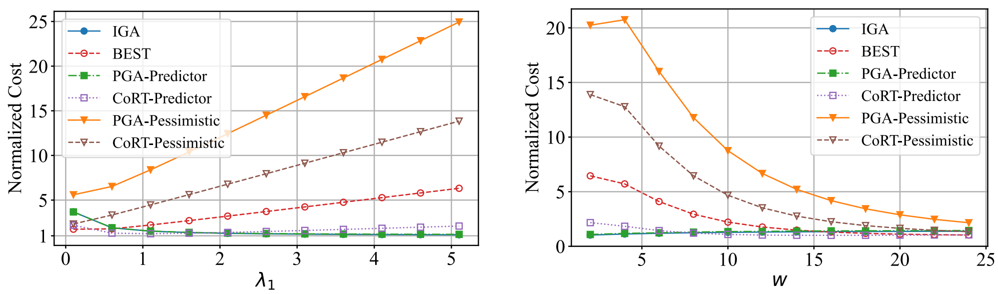
Experimental setup. We model a cloud computing platform comprising multiple independent resources (e.g., processing units such as CPUs or GPUs), serving two categories of jobs. The first category, *inelastic* jobs, consists of online job requests that require immediate allocation of resources, which remain occupied until the job is completed. The second category, *elastic* jobs, comprises predefined jobs that can be paused and resumed over time.

The platform dynamically allocates a subset of resources to elastic workloads, while the remaining units are used to process inelastic workloads. The goal is to maintain long-term SLA requirements close to predefined targets, while serving as many inelastic jobs as possible. These inelastic workloads may vary over time (e.g., due to hourly or daily patterns), making future demand difficult to predict. At each time step, the system must decide what fraction of processing units to allocate to elastic jobs, leaving the remainder for inelastic requests.

Constructing the SOOTT instance. We construct instances of SOOTT as follows: the platform acts as the decision-making agent. At time t, the agent selects an action x_t, representing the fraction of available resources allocated to inelastic jobs ($\mathbb{D} = [0, 1]$). The target for the processing rate of elastic jobs is denoted by τ_t, defined over a moving window of size w. In addition, $1 - u_t$ shows the workload demand of inelastic jobs during the next processing interval. In this setting, the *tracking cost* captures deviations from the target elastic processing rate, while the *adversarial cost* measures the gap between the actual allocation to elastic jobs and the maximum feasible allocation that would still satisfy all inelastic job requests.

Workload dataset and parameter settings. We use CPU utilization traces from the Google Cluster dataset (GCD) [17], which contains utilization records from a total of 1,600 virtual machines. The dataset provides CPU and memory utilization measurements at five-minute intervals. We divide each day into three workload periods: 8 PM–4 AM (off-peak; low demand), 4 AM–12 PM (mid-peak; medium demand), and 12 PM–8 PM (on-peak; high demand). Accordingly, we set $\tau_t = 0.4$ during low-demand hours, $\tau_t = 0.3$ during medium-demand hours, and $\tau_t = 0.2$ during high-demand hours. These thresholds result in an average allocation of approximately 30% across the day. (we have also evaluated other daily averages of τ_t; results are provided in Appendix §C) At each time step t, we extract the inelastic job utilization from the GDC dataset and define u_t as one minus this utilization. To model adversarial behavior, we use a standard convex cost function, $f_t(x) = \|x\|^2$ which is commonly used in the literature [43, 3] We vary λ_1, w, θ, λ_2, and the daily average of τ_t to evaluate their influence on the performance of online algorithms. When analyzing each parameter, we fix the others as follows: $\lambda_1 = 1$ (equal weight on elastic and inelastic jobs), $\lambda_2 = 0.1$ (to assign a 10% weight to job-switching costs), $w = 12$ (corresponding to a one-hour history window), and $\theta = 0.5$.

Prediction models. Since both CoRT and PGA rely on predictions of u_t, we evaluate three prediction models in our analysis: (1) *Predictor:* We employ an LSTM-based model [21] to forecast u_t based



(a) Impact of parameter λ_1 (b) Impact of history's length w

Figure 2: Impact of λ_1 (a) and w (b) on the cost of different algorithms. Increasing λ_1 or decreasing w amplifies the influence of the adversarial cost term, leading to higher overall costs.

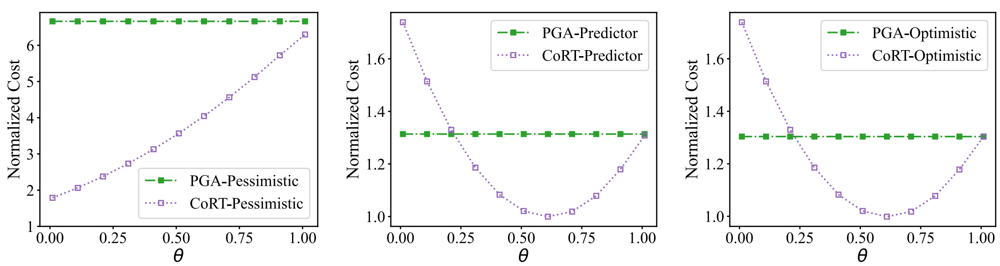


Figure 3: Comparison of the cost of PGA and CoRT as a function of θ under three prediction scenarios: pessimistic prediction (left), LSTM-based prediction (center), and perfect prediction (right).

on its historical values (see Appendix §C for details). (2) *Pessimistic:* We define the prediction as $\hat{u}_t = x_t + (x_t - u_t)$, where x_t is the action taken by BEST at time t. This formulation reflects u_t across x_t, resulting in a prediction that is deliberately misaligned with the true value, simulating an adversarial scenario. (3) *Optimistic:* This model assumes perfect prediction scenario, i.e., $\hat{u}_t = u_t$.

Experimental results. Figure 2 illustrates the impact of varying the parameter λ_1 and the history length w on the cost of online algorithms. As shown, increasing λ_1 magnifies the influence of the adversarial cost component within the overall cost function. Consequently, the cost of online algorithms such as BEST, PGA-Pessimistic, and CoRT-Pessimistic—each lacking foresight into the adversary's future targets—increases almost linearly with respect to λ_1, confirming the trend described in Remark 3.1. Notably, the increase in cost for BEST is significantly smaller than that of PGA-Pessimistic and CoRT-Pessimistic, indicating its stronger robustness. Additionally, we observe that as the history length w increases, the importance of action smoothness becomes more prominent, making the problem easier for online algorithms. In such settings, the gap between the cost of online algorithms and the optimal offline algorithm tends to narrow. Finally, we observe that the costs incurred by PGA-Predictor and CoRT-Predictor are close to those of IGA, which is due to the high accuracy of the *Predictor* model in predicting u_t (see Appendix §C for additional details).

Another observation is that in certain problem instances, algorithms such as CoRT-Predictor and BEST can achieve a lower cost than IGA. This may seem counterintuitive, as IGA has full knowledge of the adversary's target at the current time step. However, it still lacks information about future trajectory targets and future adversarial behavior. As a result, BEST—by leveraging history more effectively—can outperform it in some cases, but not certainly in worst-case as indicated in Theorem 3.2.

Figure 3 illustrates the impact of the parameter θ on the cost of CoRT under three prediction models: *Optimistic*, *Predictor*, and *Pessimistic*. The results show that when CoRT is provided with an adversarial prediction of u_t (i.e., the *Pessimistic* model), its cost increases almost quadratically with respect to θ, confirming the theoretical result in Theorem 4.4. This suggests that, to ensure strong robustness, smaller values of θ should be chosen. Conversely, the analysis using the *Optimistic* predictor indicates that increasing θ can lead to lower costs, thereby improving consistency. Together, these results highlight a fundamental trade-off between consistency and robustness in the performance of CoRT, governed by the choice of the parameter θ. We conducted further experimental analyses, the details of which are presented in Appendix §C.

6 Concluding Remarks

We introduced a new framework for Smoothed Online Optimization in target tracking, which unifies tracking of a dynamic target, robustness to adversarial perturbations, and switching costs, into a single principled formulation. Our proposed algorithms, BEST and its learning-augmented counterpart CoRT, offer both theoretical guarantees and strong empirical performance in applications such as elastic and inelastic workload scheduling. A promising direction is to design robust and competitive algorithms that relax the convexity and smoothness assumptions, thereby extending applicability to a broader range of practical settings. On the learning-augmented front, an interesting future work is to develop risk-aware learning-augmented algorithms that can dynamically adjust their reliance on predictions based on uncertainty quantification models [45, 12].

7 Acknowledgment

This research was supported in part by NSF grants CAREER 2045641, CPS-2136199, CNS-2106299, CNS-2102963, CSR-1763617, CNS-2106463, and CNS-1901137. We acknowledge their financial assistance in making this project possible.

References

[1] Soheil Abbasloo, Chen-Yu Yen, and H Jonathan Chao. Classic meets modern: A pragmatic learning-based congestion control for the internet. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 632–647, 2020.

[2] Farnaz Adib Yaghmaie and Hamidreza Modares. Online optimal tracking of linear systems with adversarial disturbances. *Transactions on Machine Learning Research*, (04), 2023.

[3] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.

[4] Manuel Amersdorfer, Jens Kappey, and Thomas Meurer. Real-time freeform surface and path tracking for force controlled robotic tooling applications. *Robotics and Computer-Integrated Manufacturing*, 65:101955, 2020.

[5] Oren Anava, Elad Hazan, and Shie Mannor. Online learning for adversaries with memory: price of past mistakes. *Advances in Neural Information Processing Systems*, 28, 2015.

[6] Benjamin Berg, Mor Harchol-Balter, Benjamin Moseley, Weina Wang, and Justin Whitehouse. Optimal resource allocation for elastic and inelastic jobs. In *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures*, pages 75–87, 2020.

[7] Allan Borodin, Nathan Linial, and Michael E Saks. An optimal on-line algorithm for metrical task system. *Journal of the ACM (JACM)*, 39(4):745–763, 1992.

[8] Jiarui Cai, Mingze Xu, Wei Li, Yuanjun Xiong, Wei Xia, Zhuowen Tu, and Stefano Soatto. Memot: Multi-object tracking with memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8090–8100, 2022.

[9] Jiho Cha. Bandit convex optimization with unbounded memory. 2024.

[10] Hao Chen and Chen Lv. Online learning-informed feedforward-feedback controller synthesis for path tracking of autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 8(4):2759–2769, 2022.

[11] Niangjun Chen, Anish Agarwal, Adam Wierman, Siddharth Barman, and Lachlan LH Andrew. Online convex optimization using predictions. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 191–204, 2015.

[12] Nicolas Christianson, Bo Sun, Steven Low, and Adam Wierman. Risk-sensitive online algorithms. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1140–1141. PMLR, 2024.

[13] Ashok Cutkosky and Kwabena A Boahen. Stochastic and adversarial online learning without hyperparameters. *Advances in neural information processing systems*, 30, 2017.

[14] Dylan Foster and Max Simchowitz. Logarithmic regret for adversarial online control. In *International Conference on Machine Learning*, pages 3211–3221. PMLR, 2020.

[15] Ruopeng Gao and Limin Wang. Memotr: Long-term memory-augmented transformer for multi-object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9901–9910, 2023.

[16] 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, 2019.

[17] Google. google/cluster-data. `https://github.com/google/cluster-data`. Accessed: 2025-05-10.

[18] Hengquan Guo, Xin Liu, Honghao Wei, and Lei Ying. Online convex optimization with hard constraints: Towards the best of two worlds and beyond. *Advances in Neural Information Processing Systems*, 35:36426–36439, 2022.

[19] Pragati Gupta. Chatgpt is at capacity. `https://writesonic.com/blog/chatgpt-at-capacity?utm_source=chatgpt.com`, 2025. Accessed: 2025-05-10.

[20] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[22] Chih-Lin Hu and Ming-Syan Chen. Online scheduling sequential objects with periodicity for dynamic information dissemination. *IEEE Transactions on Knowledge and Data Engineering*, 21(2):273–286, 2008.

[23] Rodolphe Jenatton, Jim Huang, and Cédric Archambeau. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*, pages 402–411. PMLR, 2016.

[24] Raunak Kumar, Sarah Dean, and Robert Kleinberg. Online convex optimization with unbounded memory. *Advances in Neural Information Processing Systems*, 36, 2024.

[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. *ACM SIGMETRICS Performance Evaluation Review*, 52(1):45–46, 2024.

[26] Adam Lechowicz, Nicolas Christianson, Jinhang Zuo, Noman Bashir, Mohammad Hajiesmaili, Adam Wierman, and Prashant Shenoy. The online pause and resume problem: Optimal algorithms and an application to carbon-aware load shifting. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 7(3):1–32, 2023.

[27] Jiamin Li, Hong Xu, Yibo Zhu, Zherui Liu, Chuanxiong Guo, and Cong Wang. Lyra: Elastic scheduling for deep learning clusters. EuroSys '23, page 835–850, New York, NY, USA, 2023. Association for Computing Machinery.

[28] Pengfei Li, Jianyi Yang, and Shaolei Ren. Robustified learning for online optimization with memory costs. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2023.

[29] Xu Li, Feilong Tang, Jiacheng Liu, Laurence T Yang, Luoyi Fu, and Long Chen. {AUTO}: Adaptive congestion control based on {Multi-Objective} reinforcement learning for the {Satellite-Ground} integrated network. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 611–624, 2021.

[30] Yingying Li, Xin Chen, and Na Li. Online optimal control with linear dynamics and predictions: Algorithms and regret analysis. *Advances in Neural Information Processing Systems*, 32, 2019.

[31] Yingying Li and Na Li. Leveraging predictions in smoothed online convex optimization via gradient-based algorithms. *Advances in Neural Information Processing Systems*, 33:14520–14531, 2020.

[32] Christos Liaskos, Andreas Xeros, Georgios I Papadimitriou, Marios Lestas, and Andreas Pitsillides. Broadcast scheduling with multiple concurrent costs. *IEEE Transactions on Broadcasting*, 58(2):178–186, 2012.

[33] Minghong Lin, Adam Wierman, Alan Roytman, Adam Meyerson, and Lachlan LH Andrew. Online optimization with switching cost. *ACM SIGMETRICS Performance Evaluation Review*, 40(3):98–100, 2012.

[34] Yiheng Lin, Yang Hu, Guanya Shi, Haoyuan Sun, Guannan Qu, and Adam Wierman. Perturbation-based regret analysis of predictive control in linear time varying systems. *Advances in Neural Information Processing Systems*, 34:5174–5185, 2021.

[35] Yiheng Lin, James A Preiss, Emile Anand, Yingying Li, Yisong Yue, and Adam Wierman. Online adaptive policy selection in time-varying systems: No-regret via contractive perturbations. *Advances in Neural Information Processing Systems*, 36, 2024.

[36] Jiahang Liu, Zhenhua Huang, Xin Xu, Xinglong Zhang, Shiliang Sun, and Dazi Li. Multi-kernel online reinforcement learning for path tracking control of intelligent vehicles. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(11):6962–6975, 2020.

[37] Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *Journal of the ACM (JACM)*, 68(4):1–25, 2021.

[38] Danika MacDonell and Micah Borrero. Thought experiment to explore potential savings from pooled charging infrastructure investment. White paper, Massachusetts Institute of Technology, 2023. Available at `https://dspace.mit.edu/handle/1721.1/153617`.

[39] Mark Manasse, Lyle McGeoch, and Daniel Sleator. Competitive algorithms for on-line problems. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 322–333, 1988.

[40] Merlio. Chatgpt "at capacity right now" error: What it means and how to fix it. `https://merlio.app/blog/fix-chatgpt-at-capacity-error?utm_source=chatgpt.com`, 2025. Accessed: 2025-05-10.

[41] Naram Mhaisen and George Iosifidis. Adaptive online non-stochastic control. In *6th Annual Learning for Dynamics & Control Conference*, pages 248–259. PMLR, 2024.

[42] Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ml predictions. *Advances in Neural Information Processing Systems*, 31, 2018.

[43] Guanya Shi, Yiheng Lin, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Online optimization with memory and competitive control. *Advances in Neural Information Processing Systems*, 33:20636–20647, 2020.

[44] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.

[45] Bo Sun, Jerry Huang, Nicolas Christianson, Mohammad Hajiesmaili, Adam Wierman, and Raouf Boutaba. Online algorithms with uncertainty-quantified predictions. In *Proceedings of the 41st International Conference on Machine Learning*, pages 47056–47077, 2024.

[46] Yong Xu, Meng-Ying Wan, and Zheng-Guang Wu. Adaptive learning-based path-tracking control for unknown vehicle systems under performance optimization. *IEEE Transactions on Intelligent Vehicles*, 2024.

[47] Pengyu Yan, Kaize Yu, Xiuli Chao, and Zhibin Chen. An online reinforcement learning approach to charging and order-dispatching optimization for an e-hailing electric vehicle fleet. *European Journal of Operational Research*, 310(3):1218–1233, 2023.

[48] Yu-Hu Yan, Peng Zhao, and Zhi-Hua Zhou. Online non-stochastic control with partial feedback. *Journal of Machine Learning Research*, 24(273):1–50, 2023.

[49] Hao Yu, Michael Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. *Advances in Neural Information Processing Systems*, 30, 2017.

[50] Ali Zeynali, Bo Sun, Mohammad Hajiesmaili, and Adam Wierman. Data-driven competitive algorithms for online knapsack and set cover. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10833–10841, 2021.

[51] Lijun Zhang, Wei Jiang, Jinfeng Yi, and Tianbao Yang. Smoothed online convex optimization based on discounted-normal-predictor. *Advances in Neural Information Processing Systems*, 35:4928–4942, 2022.

[52] Shiyao Zhang and JQ James. Online joint ride-sharing and dynamic vehicle-to-grid coordination for connected electric vehicle system. *IEEE Transactions on Transportation Electrification*, 10(1):1194–1206, 2023.

[53] Zhiyu Zhang, Ashok Cutkosky, and Ioannis Paschalidis. Adversarial tracking control via strongly adaptive online learning with memory. In *International Conference on Artificial Intelligence and Statistics*, pages 8458–8492. PMLR, 2022.

[54] Zhiyu Zhang, Ashok Cutkosky, and Yannis Paschalidis. Optimal comparator adaptive online learning with switching cost. *Advances in Neural Information Processing Systems*, 35:23936–23950, 2022.

[55] Peng Zhao, Yu-Hu Yan, Yu-Xiang Wang, and Zhi-Hua Zhou. Non-stationary online learning with memory and non-stochastic control. *The Journal of Machine Learning Research*, 24(1):9831–9900, 2023.

[56] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Dynamic regret of convex and smooth functions. *Advances in Neural Information Processing Systems*, 33:12510–12520, 2020.

[57] Peng Zhao, Yu-Jie Zhang, Lijun Zhang, and Zhi-Hua Zhou. Adaptivity and non-stationarity: Problem-dependent dynamic regret for online convex optimization. *Journal of Machine Learning Research*, 25(98):1–52, 2024.

A Additional Literature Review

Online Linear Tracking Control Problem The online linear tracking control problem [34, 35, 53] models a sequential decision-making scenario in which an agent selects actions over a horizon of T time steps. At each time step t, given the current state $s_t \in \mathbb{R}$, the agent selects an action $x_t \in \mathbb{R}^d$. The environment then updates the state s_{t+1} based on a known dynamics model that incorporates the previous state s_t, the current action x_t, and potentially an adversarial perturbation. The agent incurs a cost composed of a state-dependent loss $f_t(s_{t+1})$ and an action-dependent loss $c_t(x_t)$.

This framework introduces significant challenges due to the interaction between the agent's actions, the system dynamics, and adversarial perturbations—making it difficult to match the performance of an optimal offline algorithm that knows all future perturbations in advance. Nonetheless, the problem is highly relevant in several practical domains. For example, in *autonomous systems* [10, 46], self-driving vehicles must adjust their control strategies to track target trajectories despite disturbances such as wind or road condition changes. In *energy grid management* [47, 52], EV charging infrastructures must dynamically adapt to unpredictable demand fluctuations while maintaining load balance. The model is also applicable to *network congestion control* [1, 29], where network traffic must be regulated under fluctuating bandwidth constraints, and to *robotic manipulation* [36, 4], where robots must precisely follow motion plans despite external forces.

A common assumptions in previous works is the convexity of cost functions. If x_t^* denotes the minimizer of the per-step cost, then tracking x_t^* closely over time is key to minimizing cumulative cost. In some variants, such as the *online tracking control with memory*, cost functions additionally depend on a history window of past actions, typically of size w. That is, the action cost at time t may be a function $c_t(x_t, x_{t-1}, \dots, x_{t-w})$. A prominent special case is the *switching cost model*, where the cost penalizes rapid changes between consecutive actions. This is often expressed as and has been widely studied [53, 54, 55, 51, 25] to encourage smoother control policies.

Theoretical guarantees for this problem have been the subject of extensive research. In [53], the authors propose an algorithm for online tracking control with memory using online convex optimization techniques and establish a regret bound of $\mathcal{O}(\log T \cdot \sqrt{T})$. In [34], a predictive control algorithm is introduced that forecasts k steps ahead and selects actions accordingly. They show that the algorithm achieves linear regret in T, with the regret decreasing exponentially as a function of the prediction window size k. However, they also observe that the competitive ratio can increase exponentially with k, revealing a trade-off: longer prediction windows may reduce regret due to foresight, but at the cost of higher sensitivity to prediction errors. More recently, [35] proposed a gradient-based method that achieves a sublinear regret of $\mathcal{O}(\sqrt{T})$.

However, prior theoretically grounded works in this area primarily focus on minimizing short-term state and action costs, often under linear dynamic assumptions and immediate tracking objectives, without explicitly accounting for long-term behavioral constraints. As a result, they do not capture our smoothed tracking objective, which requires the agent to keep the average of its actions over a window close to a dynamically evolving sequence of targets.

Online Convex Optimization The convexity assumption and leveraging an online convex optimization techniques is common in design and analysis of algorithms for online optimization for target tracking [55, 24, 57, 2, 41, 48]. In classic online convex optimization, the agent must selects an action sequentially in order to minimize the aggregate time dependent cost function. Different versions of online convex optimization have been introduced and studied in the literature. This includes time dependent convex cost function, $c_t(x_t)$ [20, 23, 18], convex optimization with switching cost [56, 49, 33], convex optimization with memory [5, 43], enhancement using prediction [11, 30, 31], and considering adversarial perturbation in the cost function [43, 14, 13]. The similarity between online convex optimization and online optimization for target tracking problem and numerous number of previous works in online convex optimization have helped researchers to use their result for solving different aspects of the online target tracking problem. However, most of these works either omit the notion of tracking a time-varying target or focus only on instantaneous objectives, without modeling the long-term smoothed tracking similar to our targeted problem setting.

B Proofs of Theoretical Result

We start by providing proofs of key lemmas that support the theoretical results presented in the main body of the paper.

Proposition B.1 (Lemma 4 from [43])**.** *If $f : \mathbb{R}^d \to \mathbb{R}^+ \cup \{0\}$ is convex and ℓ-smooth, for any input point x, and y, and positive variable δ we have:*

$$f(y) \le (1+\delta) f(x) + (1+\frac{1}{\delta})\,\frac{\ell}{2}||y-x||^2.$$

Proof. The proof of the above proposition is given in lemma 4 of [43]. □

Lemma B.2. *Consider the action selection algorithm defined as:*

$$x(u,h) = \underset{x}{\text{argmin}}\ ||\frac{x+h}{w+1} - \tau||^2 + \lambda_1 f(x-u) + +\lambda_2 ||x-z||^2,$$

where $f(\cdot)$ is an m-strongly convex, and ℓ-smooth function. Then, the following inequality holds:

$$||x(\hat{u},\hat{h}) - x(u,h)||\ \le \frac{1}{\eta}\Big[\lambda_1 \ell ||\hat{u} - u|| + \frac{1}{(w+1)^2}||\hat{h} - h||\Big].$$

where $\eta = \frac{2}{(w+1)^2} + \lambda_1 m + 2\lambda_2$.

Proof. Let define function $\phi\big(x; u, h\big)$ as follows:

$$\phi\big(x; u, h\big) \ = \ ||\tfrac{x+h}{w+1} - \tau||^2 + \lambda_1\, f\big(x-u\big) + \lambda_2\, \|x-z\|^2.$$

We can rewrite it as:

$$\phi\big(x; u, h\big) \ = \ \frac{1}{(w+1)^2}\, \|x + h - (w+1)\tau\|^2 + \lambda_1\, f\big(x-u\big) + \lambda_2\, \|x-z\|^2.$$

The gradient of $\phi\big(x; u, h\big)$ can be derived as follows:

$$\nabla_x\, \phi\big(x; u, h\big) \ = \ \frac{2}{(w+1)^2}\, \big[x + h - (w+1)\tau\big] + \lambda_1\, \nabla f\big(x-u\big) + 2\lambda_2\, \big(x-z\big).$$

By definition, we have:

$$x(u,h) = \arg\min_x\ \phi(x; u, h),$$
$$x(\hat{u},\hat{h}) = \arg\min_x\ \phi(x; \hat{u}, \hat{h}),$$

which implies

$$\nabla_x\, \phi\big(x(u,h); u, h\big) = 0, \tag{5}$$
$$\nabla_x\, \phi\big(x(\hat{u},\hat{h}); \hat{u}, \hat{h}\big) = 0. \tag{6}$$

According to (6) we get:

$$\begin{aligned}\nabla_x\, \phi\big(x(\hat{u},\hat{h}); u, h\big) &= \nabla_x\, \phi\big(x(\hat{u},\hat{h}); u, h\big) - \nabla_x\, \phi\big(x(\hat{u},\hat{h}); \hat{u}, \hat{h}\big) \\ &= \frac{2}{(w+1)^2}\, (\hat{h} - h) + \lambda_1\, \Big[\nabla f\big(x(\hat{u},\hat{h}) - u\big) - \nabla f\big(x(\hat{u},\hat{h}) - \hat{u}\big)\Big].\end{aligned} \tag{7}$$

Since $f(\cdot)$ is ℓ-strongly smooth, we get:

$$||\nabla f\big(x(\hat{u},\hat{h}) - u\big) - \nabla f\big(x(\hat{u},\hat{h}) - \hat{u}\big)|| \le \ell\, ||\big(x(\hat{u},\hat{h}) - u\big) - \big(x(\hat{u},\hat{h}) - \hat{u}\big)|| \le\ \ell\, ||\hat{u} - u||. \tag{8}$$

In addition , since $f(\cdot)$ is η-strongly convex, we get:

$$\begin{aligned}\eta\, ||x(\hat{u},\hat{h}) - x(u,h)|| &\le ||\nabla_x \phi(x(\hat{u},\hat{h}); u, h) - \nabla_x \phi(x(u,h); u, h)|| \\ &\le ||\nabla_x \phi(x(\hat{u},\hat{h}); u, h)||.\end{aligned} \tag{9}$$

where the last inequality holds due to (5). Combining (7), (8), and (9) completes the proof. □

Lemma B.3. *Consider the function $g_t(u)$ defined as:*

$$g_t(u) = \min_x ||\frac{x + h_t}{w+1} - \tau_t||^2 + \lambda_1 \cdot f_t(x-u) + \lambda_2||x - x_{t-1}||^2,$$

where $f_t(.)$ is m-strongly convex function. The function $g_t(u)$ is η_2-strongly convex, with η_2 given by:

$$\eta_2 = \ m\lambda_1(1 - \frac{m\lambda_1}{\eta}),$$

where $\eta = \frac{2}{(w+1)^2} + m \cdot \lambda_1 + 2\lambda_2$.

Proof. To simplify the analysis, we rewrite $g_t(u)$ as:

$$g_t(u) = \min_x ||\frac{x + u + h_t}{w+1} - \tau_t||^2 + \lambda_1 \cdot f_t(x) + \lambda_2||x + u - x_{t-1}||^2,$$

To prove that $g_t(u)$ is η_2-strongly convex, we need to verify the following inequality for any u_1, u_2 and $\gamma \in [0,1]$:

$$g_t(\gamma u_1 + (1-\gamma)u_2) \leq \gamma g_t(u_1) + (1-\gamma)\phi(u_2) - \frac{\eta_2}{2}\gamma(1-\gamma)||u_1 - u_2||^2.$$

Let:

$$x_1 = \operatorname*{argmin}_x \left\|\frac{x + u_1 + h_t}{w+1} - \tau_t\right\|^2 + \lambda_1 f_t(x) + \lambda_2\|x + u_1 - x_{t-1}\|^2,$$
$$x_2 = \operatorname*{argmin}_x \left\|\frac{x + u_2 + h_t}{w+1} - \tau_t\right\|^2 + \lambda_1 f_t(x) + \lambda_2\|x + u_2 - x_{t-1}\|^2.$$

As $g_t(\cdot)$ is strongly convex we get:

$$\begin{aligned}
&\gamma g_t(u_1) + (1-\gamma)g_t(u_2) - \frac{\eta_2}{2}\gamma(1-\gamma)||u_1 - u_2||^2 \\
=&\gamma||\frac{x_1 + u_1 + h_t}{w+1} - \tau_t||^2 + \gamma\lambda_1 \cdot f_t(x_1) + \gamma\lambda_2||x_1 + u_1 - x_{t-1}||^2 \\
+&(1-\gamma)||\frac{x_2 + u_2 + h_t}{w+1} - \tau_t||^2 + (1-\gamma)\lambda_1 \cdot f_t(x_2) + (1-\gamma)\lambda_2||x_2 + u_2 - x_{t-1}||^2 \\
-&\frac{\eta_2}{2}\gamma(1-\gamma)||u_1 - u_2||^2 \\
\geq&\lambda_1 \cdot f_t(\gamma x_1 + (1-\gamma)x_2) + \frac{m \cdot \lambda_1}{2}\gamma(1-\gamma)||x_1 - x_2||^2 + \gamma||\frac{x_1 + u_1 + h_t}{w+1} - \tau_t||^2 \\
+&(1-\gamma)||\frac{x_2 + u_2 + h_t}{w+1} - \tau_t||^2 + \gamma\lambda_2||x_1 + u_1 - x_{t-1}||^2 + (1-\gamma)\lambda_2||x_2 + u_2 - x_{t-1}||^2 \\
-&\frac{\eta_2}{2}\gamma(1-\gamma)||u_1 - u_2||^2,
\end{aligned}$$

where the above inequality holds since $f_t(.)$ is m-strongly convex. By using the definition of $g_t(.)$ we get:

$$\begin{aligned}
&\gamma g_t(u_1) + (1-\gamma)g_t(u_2) - \frac{\eta_2}{2}\gamma(1-\gamma)||u_1 - u_2||^2 \\
\geq&g_t(\gamma u_1 + (1-\gamma)u_2) - ||\frac{\gamma(x_1 + u_1) + (1-\gamma)(x_2 + u_2) + h_t}{w+1} - \tau_t||^2 \\
-&\lambda_2||\gamma(x_1 + u_1) + (1-\gamma)(x_2 + y_2) - x_{t-1}||^2 + \frac{m \cdot \lambda_1}{2}\gamma(1-\gamma)||x_1 - x_2||^2 \\
+&\gamma||\frac{x_1 + u_1 + h_t}{w+1} - \tau_t||^2 + (1-\gamma)||\frac{x_2 + u_2 + h_t}{w+1} - \tau_t||^2
\end{aligned}$$

$$+\gamma\lambda_2||x_1+u_1-x_{t-1}||^2+(1-\gamma)\lambda_2||x_2+u_2-x_{t-1}||^2-\frac{\eta_2}{2}\gamma(1-\gamma)||u_1-u_2||^2,$$

Now using the fact that $\frac{1}{2}||\sqrt{z_1}x-z_2||^2$ is z_1-strongly convex we get:

$$\begin{aligned}
&\gamma g_t(u_1)+(1-\gamma)g_t(u_2)-\frac{\eta_2}{2}\gamma(1-\gamma)||u_1-u_2||^2 \qquad (10)\\
\geq& g_t(\gamma u_1+(1-\gamma)u_2)-||\frac{\gamma(x_1+u_1)+(1-\gamma)(x_2+u_2)+h_t}{w+1}-\tau_t||^2\\
-&\lambda_2||\gamma(x_1+u_1)+(1-\gamma)(x_2+y_2)-x_{t-1}||^2\\
+&||\frac{\gamma(x_1+u_1)+(1-\gamma)(x_2+u_2)+h_t}{w+1}-\tau_t||^2\\
+&\frac{1}{(w+1)^2}\gamma(1-\gamma)||(x_1-x_2)+(u_1-u_2)||^2\\
+&\lambda_2||\gamma(x_1+u_1)+(1-\gamma)(x_2+u_2)-x_{t-1}||^2+\lambda_2\gamma(1-\gamma)||x_1-x_2+u_1-u_2||^2\\
-&\frac{\eta_2}{2}\gamma(1-\gamma)||u_1-u_2||^2+\frac{m\cdot\lambda_1}{2}\gamma(1-\gamma)||x_1-x_2||^2\\
=& g_t(\gamma u_1+(1-\gamma)u_2)+\frac{m\cdot\lambda_1}{2}\gamma(1-\gamma)||x_1-x_2||^2\\
+&\frac{1}{(w+1)^2}\gamma(1-\gamma)||(x_1-x_2)+(u_1-u_2)||^2+\lambda_2\gamma(1-\gamma)||x_1-x_2+u_1-u_2||^2\\
-&\frac{\eta_2}{2}\gamma(1-\gamma)||u_1-u_2||^2. \qquad (11)
\end{aligned}$$

In addition, we have:

$$\begin{aligned}
&m\cdot\lambda_1||x_1-x_2||^2+\frac{2}{(w+1)^2}||(x_1-x_2)+(u_1-u_2)||^2\\
+&2\lambda_2||x_1-x_2+u_1-u_2||^2-\eta_2||u_1-u_2||^2\\
\geq&(m\cdot\lambda_1+\frac{2}{(w+1)^2}+2\lambda_2)||x_1-x_2||^2+(\frac{2}{(w+1)^2}+2\lambda_2-\eta_2)||u_1-u_2||^2\\
+&2(\frac{2}{(w+1)^2}+2\lambda_2)(x_1-x_2)\cdot(u_1-u_2)\\
=&\left(\sqrt{\eta}(x_1-x_2)+\frac{\eta-m\cdot\lambda_1}{\sqrt{\eta}}(u_1-u_2)\right)^2\geq 0. \qquad (12)
\end{aligned}$$

Finally inserting Equation (12) into (11) completes the proof.

□

Lemma B.4 (Adaptation of the Cauchy–Schwarz Bound)**.** *Consider two sequences of actions $x_{1:T}:=[x_1,x_2,...,x_T]$ and $y_{1:T}:=[y_1,y_2,...,y_T]$. The following inequality always holds:*

$$\sum_{t=1}^{T}\left\|\sum_{i=1}^{w}(y_{t-i}-x_{t-i})\right\|^2\leq w^2\sum_{t=1}^{T}\|y_t-x_t\|^2.$$

Proof. Expanding the left-hand side:

$$\sum_{t=1}^{T}\left\|\sum_{i=1}^{w}(y_{t-i}-x_{t-i})\right\|^2=\sum_{t=1}^{T}w^2\left\|\sum_{i=1}^{w}\frac{1}{w}(y_{t-i}-x_{t-i})\right\|^2.$$

Applying Jensen's inequality to the inner sum, we have:

$$\left\|\sum_{i=1}^{w}\frac{1}{w}(y_{t-i}-x_{t-i})\right\|^2\leq\sum_{i=1}^{w}\frac{1}{w}\|y_{t-i}-x_{t-i}\|^2.$$

Substituting this into the original expression:

$$\sum_{t=1}^{T}\left\|\sum_{i=1}^{w}(y_{t-i}-x_{t-i})\right\|^2 \le \sum_{t=1}^{T} w^2 \sum_{i=1}^{w}\frac{1}{w}\|y_{t-i}-x_{t-i}\|^2.$$

Reorganizing the terms:

$$\sum_{t=1}^{T}\left\|\sum_{i=1}^{w}(y_{t-i}-x_{t-i})\right\|^2 \le w^2 \sum_{t=1}^{T}\|y_t-x_t\|^2,$$

This completes the proof. □

B.1 Proof of Theorem 3.1

Proof. Define $\eta = 2/(w+1)^2 + m\lambda_1 + 2\lambda_2$ and the function $\mathcal{F}_1(t)$ as:

$$\mathcal{F}_1(t) = \frac{\eta}{2}||x_t - x_t^*||^2.$$

where x_t^* represents the action of the optimal offline algorithm at time step t. Summing $\mathcal{F}_1(t)$ over all time steps gives:

$$\begin{aligned}\sum_{t=1}^{T}\mathcal{F}_1(t) =& \sum_{t=1}^{T}\frac{\eta}{2}||x_t - x_t^*||^2 \\ =& \sum_{t=1}^{T}\mathcal{F}_1(t-1) + \frac{\eta}{2}\bigg(||x_T - x_T^*||^2 - ||x_0 - x_0^*||^2\bigg) = \mathcal{F}_1(T) + \sum_{t=1}^{T}\mathcal{F}_1(t-1)\end{aligned}$$

which yields:

$$\Rightarrow \sum_{t=1}^{T}\mathcal{F}_1(t) - \mathcal{F}_1(t-1) = \mathcal{F}_1(T) \ge 0. \tag{13}$$

Here, we used the fact that $x_0 = x_0^*$ from Assumption 4. Since $||\frac{x+h_t}{w+1} - \tau_t||^2 + \lambda_1\ f_t(x-u_t) + \lambda_2||x - x_{t-1}||^2$ is η-strongly convex with respect to x, and x_t is the minimizer, for $w > 0$ we obtain:

$$\begin{aligned}&||\frac{x_t+h_t}{w+1} - \tau_t||^2 + \lambda_1\ f_t(x_t-u_t) + \lambda_2||x_t - x_{t-1}||^2 \\ &+ \frac{\eta}{2}||x_t - x_t^*||^2 - \frac{\eta}{2}||x_{t-1} - x_{t-1}^*||^2 \\ &\le ||\frac{x_t^*+h_t}{w+1} - \tau_t||^2 + \lambda_1\ f_t(x_t^*-u_t) + \lambda_2||x_t^* - x_{t-1}||^2 - \frac{\eta}{2}||x_{t-1} - x_{t-1}^*||^2 \\ &= \bigg(\lambda_1 f_t(x_t^*-u_t)\bigg) + \bigg(||\frac{x_t^*+h_t}{w+1} - \tau_t||^2 + \lambda_2||x_t^* - x_{t-1}||^2 - \frac{\eta}{2}||x_{t-1} - x_{t-1}^*||^2\bigg),\end{aligned} \tag{14}$$

For any positive constants α and β, the latter term is bounded as follows:

$$\begin{aligned}&||\frac{x_t^*+h_t}{w+1} - \tau_t||^2 + \lambda_2||x_t^* - x_{t-1}||^2 - \frac{\eta}{2}||x_{t-1} - x_{t-1}^*||^2 \\ &\le ||\frac{x_t^*+h_t^*}{w+1} - \tau_t||^2 + ||\frac{h_t-h_t^*}{w+1}||^2 + 2||\frac{x_t^*+h_t^*}{w+1} - \tau_t||\cdot||\frac{h_t-h_t^*}{w+1}|| \\ &+ \lambda_2||x_t^* - x_{t-1}^*||^2 + 2\lambda_2||x_t^* - x_{t-1}^*||\cdot||x_{t-1} - x_{t-1}^*|| + \lambda_2||x_{t-1} - x_{t-1}^*||^2 \\ &- \frac{\eta}{2}||x_{t-1} - x_{t-1}^*||^2\end{aligned}$$

$$
\begin{aligned}
&\overset{(a)}{\leq} ||\frac{x_t^* + h_t^*}{w+1} - \tau_t||^2 + ||\frac{h_t - h_t^*}{w+1}||^2 + \frac{1}{\beta}||\frac{x_t^* + h_t^*}{w+1} - \tau_t||^2 \\
&+ \beta||\frac{h_t - h_t^*}{w+1}||^2 + \lambda_2||x_t^* - x_{t-1}^*||^2 + \frac{\lambda_2^2}{\alpha}||x_t^* - x_{t-1}^*||^2 \\
&+ \alpha||x_{t-1} - x_{t-1}^*||^2 + (\frac{2\lambda_2 - \eta}{2})||x_{t-1} - x_{t-1}^*||^2 \\
&\leq (1 + \frac{1}{\beta})||\frac{x_t^* + h_t^*}{w+1} - \tau_t||^2 + (1+\beta)||\frac{h_t - h_t^*}{w+1}||^2 \\
&+ \lambda_2(1 + \frac{\lambda_2}{\alpha})||x_t^* - x_{t-1}^*||^2 + (\frac{2\alpha + 2\lambda_2 - \eta}{2})||x_{t-1} - x_{t-1}^*||^2,
\end{aligned}
$$

where (a) follows from the AM-GM inequality. By summing over all time steps, we have:

$$
\begin{aligned}
&\sum_{t=1}^{T}\left(||\frac{x_t^* + h_t}{w+1} - \tau_t||^2 + \lambda_2||x_t^* - x_{t-1}||^2 - \frac{\eta}{2}||x_{t-1} - x_{t-1}^*||^2\right) \\
&\overset{(b)}{\leq}(1 + \frac{1}{\beta})\left(\sum_{t=1}^{T}||\frac{x_t^* + h_t^*}{w+1} - \tau_t||^2\right) + \frac{w^2(1+\beta)}{(w+1)^2}\left(\sum_{t=1}^{T}||x_t - x_t^*||^2\right) \\
&+ \lambda_2(1 + \frac{\lambda_2}{\alpha})\left(\sum_{t=1}^{T}||x_t^* - x_{t-1}^*||^2\right) + (\frac{2\alpha + 2\lambda_2 - \eta}{2})\left(\sum_{t=1}^{T}||x_{t-1} - x_{t-1}^*||^2\right) \\
&\leq (1 + \frac{1}{\beta})\left(\sum_{t=1}^{T}||\frac{x_t^* + h_t^*}{w+1} - \tau_t||^2\right) + \lambda_2(1 + \frac{\lambda_2}{\alpha})\left(\sum_{t=1}^{T}||x_t^* - x_{t-1}^*||^2\right) \\
&+ \left(\frac{2\alpha + 2\lambda_2 + 2(1+\beta)w^2/(w+1)^2 - \eta}{2}\right)\left(\sum_{t=1}^{T}||x_t - x_t^*||^2\right) - (\frac{2\alpha + 2\lambda_2 - \eta}{\eta})\mathcal{F}_1(T),
\end{aligned}
\tag{15}
$$

where (b) uses Lemma B.4. Substituting this into (14), we obtain:

$$
\begin{aligned}
&\sum_{t=1}^{T}||\frac{x_t + h_t}{w+1} - \tau_t||^2 + \lambda_1\ f_t(x_t - u_t) + \lambda_2||x_t - x_{t-1}||^2 \\
&\leq \left(\sum_{t=1}^{T}\lambda_1 f_t(x_t^* - u_t)\right) - \frac{2\alpha + 2\lambda_2 + \eta - \eta}{\eta}\mathcal{F}_1(T) \\
&+ (1 + \frac{1}{\beta})\left(\sum_{t=1}^{T}||\frac{x_t^* + h_t^*}{w+1} - \tau_t||^2\right) + \lambda_2(1 + \frac{\lambda_2}{\alpha})\left(\sum_{t=1}^{T}||x_t^* - x_{t-1}^*||^2\right) \\
&+ \left(\frac{2\alpha + 2\lambda_2 + 2(1+\beta)w^2/(w+1)^2 - \eta}{2}\right)\left(\sum_{t=1}^{T}||x_t - x_t^*||^2\right) \\
&\leq\ \max\{1 + \frac{1}{\beta}, 1 + \frac{\lambda_2}{\alpha}\}\ \text{Cost}(OPT, \mathcal{I}) \\
&- \frac{2\alpha + 2\lambda_2}{\eta}\mathcal{F}_1(T) + \left(\frac{2\alpha + 2\lambda_2 + 2(1+\beta)w^2/(w+1)^2 - \eta}{2}\right)\left(\sum_{t=1}^{T}||x_t - x_t^*||^2\right).
\end{aligned}
\tag{16}
$$

The additive terms would be non-positive if the following inequality holds:

$$
2\alpha + 2\lambda_2 + 2(1+\beta)w^2/(w+1)^2 - \eta \leq 0. \tag{17}
$$

This implies that if condition $2w^2/(w+1)^2 < m\lambda_1 + 2/(w+1)^2$ holds, the competitive ratio of the adversarial aware algorithm is upper bounded by:

$$\texttt{CR(IGA)} \leq 1 + \frac{2(\lambda_2 \, (w+1)^2 + w^2)}{m\lambda_1 (w+1)^2 - 2(w^2-1)}. \tag{18}$$

□

B.2 Proof of Theorem 3.2

Proof. We know the adversarial cost function $f_t(.)$ is m-strongly convex. The cost function at time step t, $\text{Cost}_t(x_t, h_t) = ||\frac{x_t+h_t}{w+1} - \tau_t||^2 + \lambda_1 f_t(x_t - u_t) + \lambda_2 ||x_t - x_{t-1}||^2$ is η-strongly convex where η can be calculated as follows:

$$\eta = \frac{2}{(w+1)^2} + m \cdot \lambda_1 + 2\lambda_2.$$

Consider the following function:

$$g_t(u) = \min_x ||\frac{x_t + \hat{h}_t}{w+1} - \tau_t||^2 + \lambda_1 f_t(x_t - u) + \lambda_2 ||x_t - \hat{x}_{t-1}||^2.$$

By the process of selecting x_t by `BEST` and the fact that function $f_t(.)$ in minimized at the origin, we reach that $u = x_t$ is the minimizer of the $g_t(u)$. From Lemma B.3, $g_t(u)$ is $\eta_2 = m\lambda_1(1 - \frac{m \cdot \lambda_1}{\eta})$-strongly convex. So by the strong convexity of $g_t(.)$ we get:

$$\begin{aligned} &||\frac{x_t + \hat{h}_t}{w+1} - \tau_t||^2 + \lambda_1 f_t(x_t - x_t) + \lambda_2 ||x_t - \hat{x}_{t-1}||^2 + \frac{\eta_2}{2} ||x_t - u_t||^2 \\ \leq\; &||\frac{\hat{x}_t + \hat{h}_t}{w+1} - \tau_t||^2 + \lambda_1 f_t(\hat{x}_t - u_t) + \lambda_2 ||\hat{x}_t - \hat{x}_{t-1}||^2. \end{aligned} \tag{19}$$

Also the function $\mathcal{F}_2(h) = ||\frac{x_t+h}{w+1} - \tau||^2$ is $\frac{2}{(w+1)^2}$-strongly smooth, so for any $0 < \delta_0$ we have:

$$\frac{1}{(1+\delta_0)} ||\frac{x_t + h_t}{w+1} - \tau_t||^2 \leq ||\frac{x_t + \hat{h}_t}{w+1} - \tau_t||^2 + \frac{1}{\delta_0 (w+1)^2} ||\hat{h}_t - h_t||^2, \tag{20}$$

Also, from Proposition B.1, for any $0 < \delta_1$ we have:

$$\frac{1}{1+\delta_1} f_t(x_t - u_t) \leq f_t(x_t - x_t) + \frac{\ell}{2\delta_1} ||u_t - x_t||^2, \tag{21}$$

In addition the function $\mathcal{F}_3(x) = \lambda_2 ||x_t - x||^2$ is $2\lambda_2$-strongly smooth, so for any $0 < \delta_2$ we have:

$$\frac{\lambda_2}{(1+\delta_2)} ||x_t - x_{t-1}||^2 \leq \lambda_2 ||x_t - \hat{x}_{t-1}||^2 + \frac{\lambda_2}{\delta_2} ||\hat{x}_{t-1} - x_{t-1}||^2. \tag{22}$$

By replacing (21), (20), and (22) into (19), we get:

$$\begin{aligned} &\frac{1}{1+\delta_0} ||\frac{x_t + h_t}{q+1} - \tau_t||^2 + \frac{\lambda_1}{1+\delta_1} f_t(x_t - u_t) + \frac{\lambda_2}{1+\delta_2} ||x_t - x_{t-1}||^2 \\ &- \frac{1}{\delta_0 (w+1)^2} ||\hat{h}_t - h_t||^2 - \frac{\lambda_1 \ell}{2\delta_1} ||u_t - x_t||^2 - \frac{\lambda_2}{\delta_2} ||\hat{x}_{t-1} - x_{t-1}||^2 + \frac{\eta_2}{2} ||u_t - x_t||^2 \\ &\leq ||\frac{\hat{x}_t + \hat{h}_t}{w+1} - \tau_t||^2 + \lambda_1 f_t(\hat{x}_t - u_t) + \lambda_2 ||\hat{x}_t - \hat{x}_{t-1}||^2. \end{aligned}$$

Which gives us:

$$\begin{aligned}&\frac{1}{1+\delta_0}||\frac{x_t+h_t}{q+1}-\tau_t||^2+\frac{\lambda_1}{1+\delta_1}f_t(x_t-u_t)+\frac{\lambda_2}{1+\delta_2}||x_t-x_{t-1}||^2\\ \le\ & ||\frac{\hat{x}_t+\hat{h}_t}{w+1}-\tau_t||^2+\lambda_1 f_t(\hat{x}_t-u_t)+\lambda_2||\hat{x}_t-\hat{x}_{t-1}||^2\\ +\ &\frac{1}{\delta_0(w+1)^2}||\hat{h}_t-h_t||^2+(\frac{\lambda_1\ell}{2\delta_1}-\frac{\eta_2}{2})||u_t-x_t||^2+\frac{\lambda_2}{\delta_2}||\hat{x}_{t-1}-x_{t-1}||^2.\end{aligned}$$

By getting sum over different time slots from both sides and using Lemma B.2 we get:

$$\begin{aligned}&\sum_{t=1}^{T}\left(\frac{1}{1+\delta_0}||\frac{x_t+h_t}{q+1}-\tau_t||^2+\frac{\lambda_1}{1+\delta_1}f_t(x_t-u_t)+\frac{\lambda_2}{1+\delta_2}||x_t-x_{t-1}||^2\right)\\ \le\ &\sum_{t=1}^{T}\left(||\frac{\hat{x}_t+\hat{h}_t}{w+1}-\tau_t||^2+\lambda_1 f_t(\hat{x}_t-u_t)+\lambda_2||\hat{x}_t-\hat{x}_{t-1}||^2\right)\\ +\ &\frac{1}{\delta_0(w+1)^2}\left(\sum_{t=1}^{T}||\hat{h}_t-h_t||^2\right)+\left(\frac{\lambda_1\ell}{2\delta_1}+\frac{\lambda_2\lambda_1^2\ell^2}{\delta_2\eta^2}-\frac{\eta_2}{2}\right)\left(\sum_{t=1}^{T}||u_t-x_t||^2\right).\end{aligned} \tag{23}$$

where the inequality is derived by applying $\hat{x}_t = x(u_t, \hat{h}_t)$ and $x_t = x(x_t, \hat{h}_t)$ in Lemma B.2. Combining this with Lemma B.4, we also obtain:

$$\sum_{t=1}^{T}||\hat{h}_t-h_t||^2\le w^2\left(\sum_{t=1}^{T}||\hat{x}_t-x_t||^2\right)\le\frac{w^2\lambda_1^2\ell^2}{\eta^2}\left(\sum_{t=1}^{T}||u_t-x_t||^2\right). \tag{24}$$

By replacing (24) into (23) we get:

$$\begin{aligned}&\min\{\frac{1}{1+\delta_0},\frac{1}{1+\delta_1},\frac{1}{1+\delta_2}\}\ \text{Cost}(\texttt{BEST},\mathcal{I})\\ \le\ &\text{Cost}(\texttt{IGA},\mathcal{I})+(\frac{\lambda_1^2\ell^2}{\delta_0\eta^2}+\frac{\lambda_1\ell}{2\delta_1}+\frac{\lambda_2\lambda_1^2\ell^2}{\delta_2\eta^2}-\frac{\eta_2}{2})\left(\sum_{t=1}^{T}||u_t-x_t||^2\right).\end{aligned} \tag{25}$$

By selecting values for δ_0, δ_1, and δ_2 as

$$\delta_0=\delta_1=\delta_2=\frac{\lambda_1\ell(\eta^2+2\lambda_1\ell(1+\lambda_2))}{\eta_2\cdot\eta^2},$$

the degradation factor of BEST will be upper bounded as follows:

$$\texttt{DF}(\texttt{BEST},\texttt{IGA})\le 1+\frac{\ell(\eta^2+2\lambda_1\ell(1+\lambda_2))}{m\eta(\eta-m\lambda_1)}.$$

□

B.3 Proof of Theorem 4.3

Proof. Let define the error of prediction of adversarial target at time step t as follows:

$$e_t := ||u_t-\hat{u}_t||.$$

We prove this theorem by constructing a specific instance of the problem. Consider the target trajectory u_t and the adversarial target trajectory τ_t defined as follows:

$$u_t = u_0, \tag{26}$$

$$\tau_t = u_0 + e_{\min}\cdot\frac{u_0}{||u_0||}, \tag{27}$$

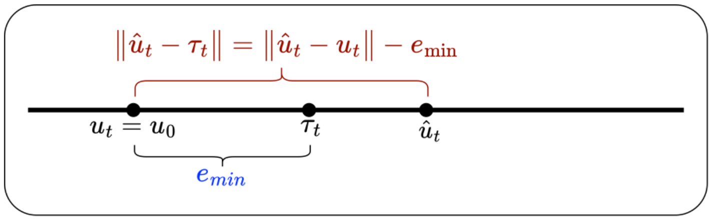


Figure 4: Coordinates of actual and predicted targets used in the proof of Theorem 4.3.

where u_0 is an arbitrary time-independent target, and $e_{\min}$ is constant which satisfies

$$e_{\min} \leq \min_t \; e_t.$$

Now, suppose that the predicted value of u_t satisfies the following condition:

$$\hat{u}_t = u_t + e_t \cdot \frac{u_0}{||u_0||}; \tag{28}$$

see Figure 4 for an illustration.

Under this setup, the cost incurred by IGA is upper-bounded as:

$$\mathrm{Cost}(\texttt{IGA}, \mathcal{I}_0) \leq \lambda_1 \sum_{t=1}^{T} f_t(\tau_t - u_t), \tag{29}$$

where this bound is attained when IGA selects τ_t at every time step.

On the other hand, the cost incurred by PGA satisfies the following lower bound:

$$\mathrm{Cost}(\texttt{PGA}, \mathcal{I}_0) \geq \lambda_1 \sum_{t=1}^{T} f_t(\tilde{x}_t - u_t). \tag{30}$$

Given (27) and (28), there exists a positive constant α_t such that, for every time step t, we can express $\tilde{x}_t$ as:

$$\tilde{x}_t = (1 + \alpha_t \lambda_1)\tau_t. \tag{31}$$

Note that, when λ_1 gets very small values, $\tilde{x}_t$ converges to τ_t. Substituting this into (30) gives:

$$\mathrm{Cost}(\texttt{PGA}, \mathcal{I}_0) \geq \lambda_1 \sum_{t=1}^{T} f_t(\tau_t - u_t) + \frac{\lambda_1 m}{2} \sum_{t=1}^{T} \|e_t - e_{\min}\|^2 . \tag{32}$$

Substituting (29) into (32) gives:

$$\frac{\mathrm{Cost}(\texttt{PGA}, \mathcal{I}_0)}{\mathrm{Cost}(\texttt{IGA}, \mathcal{I}_0)} \geq 1 + \frac{m \sum_{t=1}^{T} ||e_t - e_{\min}||^2}{2 \sum_{t=1}^{T} f_t(\tau_t - u_t)} = 1 + \frac{m\lambda_1 \sum_{t=1}^{T} ||e_t - e_{\min}||^2}{2\lambda_1 \sum_{t=1}^{T} f_t(e_{\min} \cdot u_0/||u_0||)}. \tag{33}$$

and limiting $e_{\min} \to 0$ completes the proof. □

B.4 Proof of Theorem 4.4

Proof. We begin by analyzing the performance of CoRT under fully adversarial predictions, highlighting the robustness of CoRT. Let $\tilde{x}_t$ and x_t denote the actions of CoRT and BEST, respectively, at time step t. By Proposition B.1 and Lemma B.2, for any positive parameter δ, we have

$$\sum_{t=1}^{T} \left\| \frac{\tilde{x}_t + \tilde{h}_t}{w+1} - \tau_t \right\|^2 \leq \sum_{t=1}^{T} (1+\delta) \left\| \frac{x_t + h_t}{w+1} - \tau_t \right\|^2$$

$$+\sum_{t=1}^{T}\left(1+\frac{1}{\delta}\right)\frac{1}{(w+1)^2}\left\|\tilde{x}_t+\tilde{h}_t-x_t-h_t\right\|^2$$

$$\leq \sum_{t=1}^{T}(1+\delta)\left\|\frac{x_t+h_t}{w+1}-\tau_t\right\|^2+\sum_{t=1}^{T}\left(1+\frac{1}{\delta}\right)\|\tilde{x}_t-x_t\|^2, \quad (34)$$

where the last inequality uses Lemma B.4. Similarly, for the regularization term, we have

$$\sum_{t=1}^{T}\lambda_2\|\tilde{x}_t-\tilde{x}_{t-1}\|^2 \leq \sum_{t=1}^{T}(1+\delta)\lambda_2\|x_t-x_{t-1}\|^2+\sum_{t=1}^{T}\left(1+\frac{1}{\delta}\right)\lambda_2\|\tilde{x}_t-\tilde{x}_{t-1}-(x_t-x_{t-1})\|^2$$

$$\leq (1+\delta)\sum_{t=1}^{T}\lambda_2\|x_t-x_{t-1}\|^2+4(1+\frac{1}{\delta})\lambda_2\sum_{t=1}^{T}\|\tilde{x}_t-x_t\|^2. \quad (35)$$

Since $f_t(\cdot)$ is ℓ-strongly smooth, we have

$$\lambda_1 f_t(\tilde{x}_t-u_t) \leq \lambda_1 f_t(x_t-u_t)+\frac{\ell\lambda_1}{2}\|\tilde{x}_t-x_t\|^2+\lambda_1\nabla f_t(x_t-u_t)\cdot(\tilde{x}_t-x_t). \quad (36)$$

By combining (34), (35), and (36), for any instance input $\mathcal{I}$, we obtain

$$\mathrm{Cost}(\texttt{CoRT},\mathcal{I}) \leq (1+\delta)\,\mathrm{Cost}(\texttt{BEST},\mathcal{I})+\left(1+\frac{1}{\delta}\right)\left(1+4\lambda_2+\frac{\ell\lambda_1}{2}\right)\sum_{t=1}^{T}\|\tilde{x}_t-x_t\|^2$$

$$+\lambda_1\sum_{t=1}^{T}\nabla f_t(x_t-u_t)\cdot(\tilde{x}_t-x_t).$$

Moreover, since $f_t(\cdot)$ is ℓ-strongly smooth, it follows that

$$\|\nabla f_t(x_t-u_t)\| \leq \ell\|x_t-u_t\|, \quad (37)$$

which implies

$$\mathrm{Cost}(\texttt{CoRT},\mathcal{I}) \leq (1+\delta)\,\mathrm{Cost}(\texttt{BEST},\mathcal{I})+\left(1+\frac{1}{\delta}\right)\left(1+4\lambda_2+\frac{\ell\lambda_1}{2}\right)\sum_{t=1}^{T}\|\tilde{x}_t-x_t\|^2$$

$$+\lambda_1\ell\sum_{t=1}^{T}\|x_t-u_t\|\cdot\|\tilde{x}_t-x_t\|$$

$$\leq (1+\delta)\,\mathrm{Cost}(\texttt{BEST},\mathcal{I})+\left(1+\frac{1}{\delta}\right)\left(1+4\lambda_2+\frac{\ell\lambda_1}{2}\right)\sum_{t=1}^{T}\|\tilde{x}_t-x_t\|^2$$

$$+\lambda_1\ell\sum_{t=1}^{T}\left[\frac{1}{\alpha}\|x_t-u_t\|^2+\alpha\|\tilde{x}_t-x_t\|^2\right], \quad (38)$$

where α is an arbitrary positive constant. In addition, for BEST we have:

$$\mathrm{Cost}(\texttt{BEST},\mathcal{I}) \geq \lambda_1\sum_{t=1}^{T}f_t(x_t-u_t) \geq \frac{m\lambda_1}{2}\sum_{t=1}^{T}||x_t-u_t||^2. \quad (39)$$

Also, based on Lemma B.2, we have

$$\sum_{t=1}^{T}\|\tilde{x}_t-x_t\|^2 \leq (\frac{\lambda_1\ell}{\eta})^2\sum_{t=1}^{T}\|x_t-\tilde{u}_t\|^2 \leq (\frac{\lambda_1\ell\theta}{\eta})^2\sum_{t=1}^{T}D_t^2$$

$$\leq (\frac{\lambda_1\ell\theta}{\eta})^2\sum_{t=1}^{T}\|x_t-u_t\|^2 \leq \frac{2\lambda_1\ell^2}{m\eta^2}\,\theta^2\,\mathrm{Cost}(\texttt{BEST},\mathcal{I}), \quad (40)$$

By combining (40), (39), and (38), we obtain

$$\text{Cost}(\texttt{CoRT}, \mathcal{I}) \leq \left[1 + \delta + \frac{\lambda_1}{\alpha} + \left((1+\frac{1}{\delta})(1 + 4\lambda_2 + \frac{\ell\lambda_1}{2}) + \alpha\lambda_1\ell(\frac{\lambda_1\ell}{\eta})^2\right)\theta^2\right]\text{Cost}(\texttt{BEST}, \mathcal{I}), \tag{41}$$

Moreover, as $\theta \to 0$, CoRT converges to BEST. This implies

$$\texttt{DF}(\texttt{CoRT}, \texttt{IGA}) \leq \texttt{DF}(\texttt{BEST}, \texttt{IGA})\left(1 + \theta^2\mathcal{O}(1)\right). \tag{42}$$

Next, we proceed to analyze the performance of CoRT under perfect prediction (Consistency analysis).

Let x_t denote the action of IGA at time step t. Under perfect prediction conditions, we have $\hat{u}_t = \tilde{u}_t$. In such a case, if $\|u_t - x_t\| \leq D_t$, the actions of IGA and CoRT coincide. Thus, in order to maximize the gap between the performance of CoRT and IGA, in the worst case scenario, $\mathcal{I}_{worst}$, an adversary must select targets such that the following inequality holds:

$$D_t \leq \|u_t - x_t\|. \tag{43}$$

Based on Assumption 4, u_0 and x_0 are identical initially, implying $D_1 = 0$. Combining this with the above inequality, we conclude that, in the worst-case scenario, the following relation holds:

$$D_t = \|u_{t-1} - x_{t-1}\| \leq \|u_t - x_t\| = D_{t+1}. \tag{44}$$

Furthermore, by the definitions of x_t, $\tilde{x}_t$, $\hat{x}_t$, and u_t, these points lie along a direct line segment. Consequently, there exist constants $\beta_t \in [0,1]$ such that

$$\tilde{x}_t = \beta_t\hat{x}_t + (1-\beta_t)x_t. \tag{45}$$

Based on this, using the convexity of cost terms we get:

$$\begin{aligned}
\text{Cost}_t(\texttt{CoRT}) &= ||\frac{\tilde{x}_t + \tilde{h}_t}{w+1} - \tau_t||^2 + \lambda_1 f_t(\tilde{x}_t - u_t) + \lambda_2||\tilde{x}_t - \tilde{x}_{t-1}||^2 \\
&\leq \beta||\frac{\hat{x}_t + \hat{h}_t}{w+1} - \tau_t||^2 + (1-\beta_t)||\frac{x_t + h_t}{w+1} - \tau_t||^2 \\
&+ \beta_t\lambda_1 f_t(\hat{x}_t - u_t) + (1-\beta_t)\lambda_1 f_t(x_t - u_t) \\
&+ \lambda_2\ \beta_t||\hat{x}_t - \tilde{x}_{t-1}||^2 + \lambda_2(1-\beta_t)||x_t - \tilde{x}_{t-1}||^2 \\
&\leq \beta_t||\frac{\hat{x}_t + \hat{h}_t}{w+1} - \tau_t||^2 + (1-\beta_t)||\frac{x_t + h_t}{w+1} - \tau_t||^2 \\
&+ \beta_t\lambda_1 f_t(\hat{x}_t - u_t) + (1-\beta_t)\lambda_1 f_t(x_t - u_t) \\
&+ \lambda_2\ \beta_t||\hat{x}_t - \hat{x}_{t-1}||^2 + \lambda_2(1-\beta_t)||x_t - x_{t-1}||^2 \\
&+ \lambda_2\beta_t(\frac{\lambda_1\ell}{\eta})^2(D_t - \theta D_{t-1})^2 + \lambda_2(1-\beta_t)(\frac{\lambda_1\ell}{\eta})^2\theta^2 D_t^2,
\end{aligned} \tag{46}$$

where the last inequality holds only for the worst-case instance $\mathcal{I}_{\text{worst}}$, using the fact that, by definition, the following property holds for $\mathcal{I}_{\text{worst}}$:

$$\begin{aligned}
\theta D_t &\leq D_{t+1}, \quad \forall t && (47)\\
||\tilde{u}_t - x_t|| &= \theta D_t, \quad \forall t && (48)\\
||u_t - \tilde{u}_t|| &= D_{t+1} - \theta D_t, \quad \forall t && (49)\\
||u_t - x_t|| &= D_{t+1}. \quad \forall t && (50)
\end{aligned}$$

This yields:

$$\begin{aligned}
\text{Cost}_t(\texttt{CoRT}, \mathcal{I}_{worst}) \leq\ & \beta_t\ \text{Cost}_t(\texttt{IGA}, \mathcal{I}_{worst}) \\
&+ (1-\beta_t)\ \text{Cost}_t(\texttt{BEST}, \mathcal{I}_{worst})
\end{aligned}$$

$$+ \lambda_2(\frac{\lambda_1 \ell}{\eta})^2 \left[D_t^2 \left(\beta_t (1 - \frac{\theta D_{t-1}}{D_t})^2 + (1-\beta_t)\theta^2 \right) \right]. \tag{51}$$

In addition, we can provide upper bounds on the values of β_t and $1 - \beta_t$ as follows:

$$\beta_t = \frac{\|\tilde{x}_t - x_t\|}{\|\hat{x}_t - x_t\|} \leq \frac{\lambda_1 \ell}{\eta} \cdot \frac{\eta}{m\lambda_1} \cdot \frac{\theta D_t}{D_{t+1}} = \frac{\ell}{m} \cdot \frac{\theta D_t}{D_{t+1}}, \tag{52}$$

$$1 - \beta_t = \frac{\|\tilde{x}_t - \hat{x}_t\|}{\|\hat{x}_t - x_t\|} \leq \frac{\ell}{m} \cdot \left(\frac{D_{t+1} - \theta D_t}{D_{t+1}} \right) = \frac{\ell}{m}(1 - \frac{\theta D_t}{D_{t+1}}), \tag{53}$$

where we used the convexity of the cost function and Lemma B.2 to derive these bounds. These expressions reveal that when θ is small (i.e., $\theta \to 0$), β_t also becomes small, indicating that the action of CoRT closely follows that of BEST. Conversely, as θ grows large (i.e., $\theta \to \infty$), β_t approaches 1, and the action of CoRT becomes similar to that of IGA.

Also, since BEST is minimizing the cost value ignoring the adversarial cost at time step t, the cost of IGA in the worst case instance is lower bounded as follows:

$$\begin{aligned}
\text{Cost}(\texttt{IGA}, \mathcal{I}_{worst}) &\geq \sum_{t=1}^{T} \left(||\frac{x_t + \hat{h}_t}{w+1} - \tau_t||^2 + \lambda_2 ||x_t - \hat{x}_{t-1}||^2 \right) \\
&+ (\frac{\eta - \lambda_1 m}{2})(\frac{m\lambda_1}{\eta})^2 \sum_{t=1}^{T} ||x_t - u_t||^2 + \frac{m\lambda_1}{2}(\frac{\eta - m\lambda_1}{\eta})^2 \sum_{t=1}^{T} ||x_t - u_t||^2 \\
&\geq \sum_{t=1}^{T} \left(||\frac{x_t + \hat{h}_t}{w+1} - \tau_t||^2 + \lambda_2 ||x_t - \hat{x}_{t-1}||^2 \right) \\
&+ \frac{m\lambda_1}{2\eta}(\eta - m\lambda_1) \sum_{t=1}^{T} ||x_t - u_t||^2 \\
&\geq \frac{m\lambda_1}{2\eta}(\eta - m\lambda_1) \sum_{t=1}^{T} D_{t+1}^2 \\
&= \frac{m\lambda_1}{2\eta}(\eta - m\lambda_1) \sum_{t=1}^{T+1} D_t^2,
\end{aligned} \tag{54}$$

where in the last inequality we used the fact that $D_1 = 0$. Combining (51) and (54) yields:

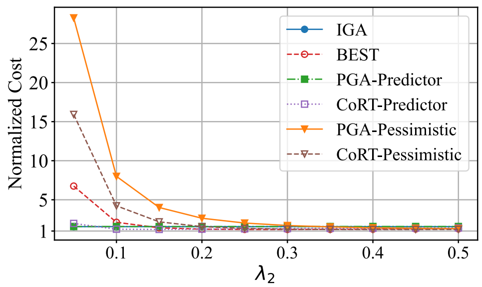
$$\begin{aligned}
\frac{\text{Cost}_t(\texttt{CoRT}, \mathcal{I}_{worst})}{\text{Cost}_t(\texttt{IGA}, \mathcal{I}_{worst})} &\leq \beta_t + (1 - \beta_t) \frac{\text{Cost}_t(\texttt{BEST}, \mathcal{I}_{worst})}{\text{Cost}_t(\texttt{IGA}, \mathcal{I}_{worst})} \\
&+ \frac{2\lambda_2\lambda_1\ell^2 \left[\sum_{t=1}^{T} D_t^2 \left(\beta_t(1 - \frac{\theta D_{t-1}}{D_t})^2 + (1-\beta_t)\theta^2 \right) \right]}{m\eta(\eta - m\lambda_1) \sum_{t=1}^{T+1} D_t^2},
\end{aligned} \tag{55}$$

Note that the latter term increases with θ, and both the numerator and denominator grow at most quadratically with respect to θ. Its maximum value, as $\theta \to \infty$, is bounded by:

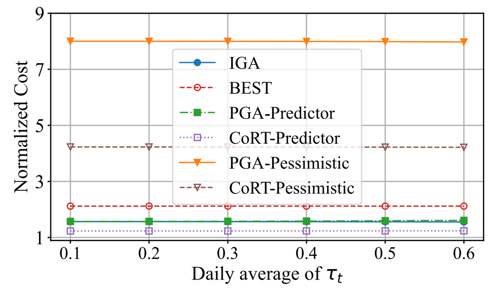
$$\begin{aligned}
&\left[\frac{2\lambda_2\lambda_1\ell^2 \left[\sum_{t=1}^{T} D_t^2 \left(\beta_t(1 - \frac{\theta D_{t-1}}{D_t})^2 + (1-\beta_t)\theta^2 \right) \right]}{m\eta(\eta - m\lambda_1) \sum_{t=1}^{T+1} D_t^2} \Bigg| \theta \to \infty \right] \\
&\leq \frac{2\lambda_2\lambda_1\ell^2 \sum_{t=1}^{T} D_t^2 \theta^2}{m\eta(\eta - m\lambda_1) \sum_{t=1}^{T+1} D_t^2} \leq \frac{2\lambda_2\lambda_1\ell^2}{m\eta(\eta - m\lambda_1)}.
\end{aligned} \tag{56}$$

Finally, the proof follows by noting that β_t increases with θ, converging to 0 as $\theta \to 0$, and approaching 1 as $\theta \to \infty$.

□



(a) Impact of parameter λ_2



(b) Impact of daily average of trajectory targets

Figure 5: Impact of λ_2 and the daily average of trajectory targets, τ_t, on algorithm cost. While λ_2 significantly affects the normalized cost of the algorithms, the daily average of τ_t has a minimal impact.

C Additional Details of Experiments

In this section, we provide additional details of the experimental setup.

C.1 Result of Experiments on Impact of λ_2, and τ_t

Figure 5a illustrates the impact of the switching cost coefficient λ_2 on the total cost incurred by the algorithms. The results show that λ_2 influences the cost functions in a manner similar to the weight parameter w. As λ_2 increases, both online algorithms and the offline optimal algorithm are more heavily penalized for making large changes between consecutive actions. This discourages frequent switching, leading to smoother action sequences. Consequently, the adversarial cost component contributes less to the overall cost, resulting in reduced total cost values.

We also evaluate the impact of the daily average value of τ_t on algorithm performance. Following the structure described in Section 5, we vary τ_t across the day by modifying its value during mid-peak periods and then shifting it by $+0.1$ (i.e., 10%) during off-peak and -0.1 (i.e., 10%) during on-peak hours. This setup ensures that the daily average of τ_t matches its value during mid-peak periods. Results of this analysis, shown in Figure 5b, indicate that the effect of the daily average of τ_t on the normalized cost is modest compared to other parameters like λ_1, λ_2, and w. This limited sensitivity is intuitive, as we preserve the shape of the τ_t variation pattern throughout the day and only apply a uniform shift. Note that this analysis focuses solely on the impact of daily average τ_t on algorithm cost; exploring its influence on other metrics—such as the average allocation to elastic or inelastic workloads—is left for future work.

C.2 More Detail on the LSTM Predictor Used in Section 5

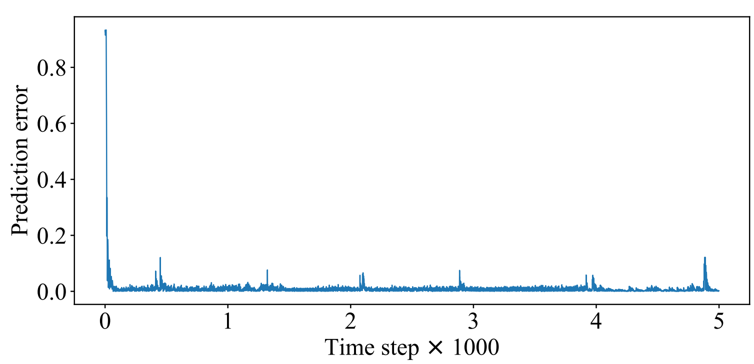


Figure 6: Prediction error $\|u_t - \hat{u}_t\|$ over time steps.

To estimate the adversary's target u_t at each time step in an online fashion, we implement an LSTM-based regression model that learns the temporal dependencies in the observed sequence of u values. Specifically, we train a one-layer Long Short-Term Memory (LSTM) network followed by a fully connected linear layer. The LSTM model receives a sliding window of the previous W observations $\{u_{t-W}, \ldots, u_{t-1}\}$ and predicts the next value $\hat{u}_t$.

Our architecture consists of:

Input layer: A sequence of $W = 10$ scalar values, each representing the observed u_t at previous time steps.

LSTM layer: A single-layer LSTM with hidden size 32, which processes the input sequence and outputs a hidden state vector representing the temporal features of the sequence.

Output layer: A linear layer of size $32 \to 1$ that maps the last hidden state to the final prediction $\hat{u}_t$.

We train the model incrementally in an online manner, using a single gradient update per time step. The model is optimized using the Adam optimizer with a learning rate of 10^{-2}. The training is performed in real-time as new data arrives, making the approach suitable for dynamic and non-stationary environments.

Figure 6 shows the prediction error ($\|u_t - \hat{u}_t\|$) over time for the first 5,000 steps. The results demonstrate that the LSTM network achieves a high level of accuracy in predicting u_t. Specifically, the average prediction error across the entire horizon is 0.01, with a standard deviation of 0.04. Owing to this high accuracy, the performance of PGA-Predictor and CoRT-Predictor closely matches that of PGA-Optimistic and CoRT-Optimistic reported in Section 5.