Learning-Augmented Competitive Algorithms for Spatiotemporal Online Allocation with Deadline Constraints

ADAM LECHOWICZ, University of Massachusetts Amherst, USA
NICOLAS CHRISTIANSON, California Institute of Technology, USA
BO SUN, University of Waterloo, Canada
NOMAN BASHIR, Massachusetts Institute of Technology, USA
MOHAMMAD HAJIESMAILI, University of Massachusetts Amherst, USA
ADAM WIERMAN, California Institute of Technology, USA
PRASHANT SHENOY, University of Massachusetts Amherst, USA

We introduce and study spatiotemporal online allocation with deadline constraints (SOAD), a new online problem motivated by emerging challenges in sustainability and energy. In SOAD, an online player completes a workload by allocating and scheduling it on the points of a metric space (X, d) while subject to a deadline T. At each time step, a service cost function is revealed that represents the cost of servicing the workload at each point, and the player must irrevocably decide the current allocation of work to points. Whenever the player moves this allocation, they incur a movement cost defined by the distance metric $d(\cdot, \cdot)$ that captures, e.g., an overhead cost. SOAD formalizes the open problem of combining general metrics and deadline constraints in the online algorithms literature, unifying problems such as metrical task systems and online search. We propose a competitive algorithm for SOAD along with a matching lower bound establishing its optimality. Our main algorithm, ST-CLIP, is a learning-augmented algorithm that takes advantage of predictions (e.g., forecasts of relevant costs) and achieves an optimal consistency-robustness trade-off. We evaluate our proposed algorithms in a simulated case study of carbon-aware spatiotemporal workload management, an application in sustainable computing that schedules a delay-tolerant batch compute job on a distributed network of data centers. In these experiments, we show that ST-CLIP substantially improves on heuristic baseline methods.

CCS Concepts: • **Theory of computation → Online algorithms**; **Theory and algorithms for application domains**; • **Social and professional topics** → *Sustainability*.

Additional Key Words and Phrases: online allocation, switching costs, deadline constraints, carbon-aware spatiotemporal load shifting, learning-augmented algorithms

---

Authors' Contact Information: Adam Lechowicz, University of Massachusetts Amherst, USA, alechowicz@cs.umass.edu; Nicolas Christianson, California Institute of Technology, USA, nchristianson@caltech.edu; Bo Sun, University of Waterloo, Canada, bo.sun@uwaterloo.ca; Noman Bashir, Massachusetts Institute of Technology, USA, nbashir@mit.edu; Mohammad Hajiesmaili, University of Massachusetts Amherst, USA, hajiesmaili@cs.umass.edu; Adam Wierman, California Institute of Technology, USA, adamw@caltech.edu; Prashant Shenoy, University of Massachusetts Amherst, USA, shenoy@cs.umass.edu.

---

1 Introduction

We introduce and study spatiotemporal online allocation with deadline constraints (SOAD), an online optimization problem motivated by emerging challenges in sustainability. In SOAD, an online player aims to service a workload by allocating and scheduling it on one of n points represented by a metric space (X, d). They pay a service cost at a point if the workload is currently being serviced there, a spatial movement cost defined by the metric whenever they change the allocation between points, and a temporal switching cost when bringing the workload into or out of service at a single point. The workload arrives with a *deadline constraint T* that gives the player some *slack*, i.e., the workload can be *paused* for some time to avoid high cost periods without violating the constraint.

SOAD builds on a long history of related problems in online algorithms. In particular, two lines of work share specific features in common with our setting. One line of work focuses on metrical task systems (MTS) and smoothed online convex optimization (SOCO), where problems consider online optimization with movement costs over general metrics, but do not accommodate long-term constraints, such as deadlines [6, 10, 15–18, 38]. A complementary line of work is that of online search problems with long-term constraints, such as one-way trading (OWT) and online knapsack – these problems enforce that a player's cumulative decisions satisfy a constraint over the entire input, but do not consider general metric (decision) spaces or movement costs [27, 50, 59, 70]. SOAD extends both MTS/SOCO-type problems and OWT-type problems by simultaneously considering general metric movement/switching costs and deadline (i.e., long-term) constraints.

For many applications, the underlying problem to be solved often requires a model with *both* smoothed optimization (i.e., movement costs) *and* deadline constraints. Furthermore, for an application such as carbon-aware workload management in data centers, where the spatial movement cost corresponds to, e.g., network delays (see Section 2.2), it is necessary to consider a general metric space, since pairwise network latencies do not necessarily correspond to simple distances such as Euclidean (geographic) distances. The question of whether it is possible to design competitive online algorithms in this combined setting has remained open for over a decade, with theoretical progress emerging only in the last few years in special cases such as the unidimensional setting, in ℓ_1 vector spaces, or with different performance metrics such as regret [4, 12, 36, 40–42]. This work seeks to close this gap by answering the following question:

Is it possible to design online algorithms for SOAD *that manage the challenges of **general metrics** and provide competitive guarantees without violating the **deadline constraint**?*

It is well known that problems related to SOAD, such as MTS and OWT, are difficult in the sense that their competitive ratios scale in the size of the decision space or the ratio between maximal and minimal prices. However, these pessimistic lower bounds hold in the worst case, while in practice a decision-maker can often leverage data-driven machine learning approaches to obtain algorithms that perform better empirically. Recent work in the online algorithms literature has leveraged the paradigm of learning-augmented algorithms [39, 52] to design and analyze algorithms that can take advantage of patterns in the input via untrusted "advice" (e.g., predictions from a machine learning model) without losing adversarial competitive bounds. Such learning-augmented algorithms have been designed for precursor problems to SOAD, including MTS and OWT [3, 21, 22, 69]. In the SOAD setting, supported by the availability of practical predictions for our motivating applications and a lack of learning-augmented algorithmic strategies that accommodate both general metrics and deadline constraints, we additionally consider the question:

Can we design algorithms for SOAD *that integrate **untrusted advice** (such as machine-learned predictions) to further improve performance without losing worst-case guarantees?*

1.1   Related work

Our results address a long-standing open problem of combining online optimization with general switching costs (MTS/SOCO) and deadline constraints. Although general MTS is a famously well-studied problem in online algorithms [5–7, 9, 10, 23], it has not been studied under the general form of long-term intertemporal constraints that we consider in the SOAD formulation. Amongst the two lines of related work, prior work has focused on either designing MTS-style algorithms using techniques such as mirror descent [16], primal-dual optimization [5], and work functions [1], or OWT-style algorithms using techniques such as threshold-based algorithm design [50, 82], pseudo-reward maximization [70, 78], and protection-level policies [27]. As these distinct techniques have been tailored to their respective problem settings, there has been almost no cross-pollination between MTS-type and OWT-type problems until this work. The combination of smoothed optimization and long-term constraints has drawn recent attention in the paradigm of regret analysis in problems such as bandits with knapsacks and OCO with long-term constraints [4, 36]. However, despite established connections between MTS and online learning [9], the problem of optimal *competitive* algorithm design in general settings of this form has yet to be explored.

A select few works [12, 40–42] consider both *switching costs* and deadline constraints in a competitive regime, although they are restricted to special cases such as unidimensional decisions or ℓ_1 metrics. Due to these assumptions, their results and algorithms fail to capture the general problem that we consider. As just one example, all of these works assume that the switching cost is only temporal, in the sense that the online player pays the same cost whether they are switching into or out of a state that makes progress towards the deadline constraint. This assumption is overly restrictive because it cannot accommodate switching cost situations that may arise in motivating applications, e.g., the case where the player chooses to move between points of the metric while simultaneously switching into an ON state.

Our work also contributes to the field of learning-augmented algorithms, designed to bridge the performance of untrusted advice and worst-case competitive guarantees [39, 52]. Learning-augmented design has been studied in many online problems including ski rental [75], bipartite matching [37], and several related problems including MTS/SOCO and OWT [3, 21, 40, 41, 43, 69]. For MTS/SOCO, a dominant algorithmic paradigm is to adaptively combine the actions of a robust decision-maker and those of e.g., a machine-learning model [3, 21]; optimal trade-offs between robustness and consistency have also been shown in the case of general metrics [22]. For OWT and k-search, several works have given threshold-based algorithms incorporating predictions that are likewise shown to achieve an optimal robustness-consistency trade-off [43, 69]. The advice models for these two tracks of literature are quite different and lead to substantially different algorithms – namely, online search problems typically assume that the algorithm receives a prediction of, e.g., the *best price*, while MTS/SOCO typically consider *black-box advice* predicting the optimal decision at each time step. Amongst the limited prior literature that considers learning-augmentation in problems with switching costs and deadline constraints [12, 40, 41], both advice models have been considered, underscoring the challenge of the SOAD setting, where the optimal choice of advice model (and corresponding design techniques) is not obvious *a priori*.

1.2   Contributions

Our main technical contributions make progress on a longstanding problem in online optimization that models emerging practical problems in areas such as sustainability. Our algorithms and lower bounds for SOAD are the first results to consider competitive analysis for deadline-constrained problems on general metrics. We obtain positive results for both of the questions posed above under assumptions informed by practice. In particular, we provide the first competitive algorithm, PCM

(**P**seudo-**C**ost **M**inimization, see Algorithm 1), for this type of problem in Section 3, and show that it achieves the best possible competitive ratio up to log factors that result from the generality of the metric. Surprisingly, the competitive upper bound we prove for SOAD (Theorem 3.2) compares favorably against known strong lower bounds for precursor problems such as MTS and OWT; which suggests an insight that additional structure imposed by constraints can actually facilitate competitive decision making, despite the added complexity of the general metric and deadline constraint. To achieve this result, we develop theoretical tools from both lines of related work that help us tackle challenging components of SOAD. For instance, we leverage randomized metric embeddings and optimal transport to endow a general metric (X, d) with a structure that facilitates analysis. From the online search literature, we leverage techniques that balance the trade-off between cost and constraint satisfaction, specifically generalizing these ideas to operate in the more complex metric setting necessitated by SOAD.

In Section 4, we introduce our main learning-augmented algorithm, ST-CLIP, which integrates black-box decision advice based on, e.g., machine-learned predictions to significantly improve performance without losing worst-case competitive bounds. In our approach, we first prove an impossibility result on the robustness-consistency trade-off for any algorithm. Using an adaptive optimization-based framework first proposed by [40], we design an algorithm that combines the theoretical tools underpinning PCM in concert with a constraint hedging against worst-case service costs and movement costs that threaten the desired consistency bound. This ensures ST-CLIP attains the optimal trade-off (up to log factors) in general metric spaces.

In Section 5, motivated by real-world applications where the movement cost between points of the metric may not be constant, we present a generalization of SOAD where distances are allowed to be *time-varying* and show that our algorithms extend to this case. Finally, in Section 6, we evaluate our algorithms in a case study of carbon-aware spatiotemporal workload management (see Section 2.2) on a simulated global network of data centers. We show that ST-CLIP is able to leverage *imperfect* advice and significantly improve on heuristic baselines for the problem.

2 Problem Formulation, Motivating Applications, Challenges, and Preliminaries

In this section, we introduce the spatiotemporal online allocation with deadline constraints (SOAD) problem and provide motivating applications as examples. We also discuss some intrinsic challenges in SOAD that prevent the direct application of existing techniques, and introduce relevant preliminaries from related work.

2.1 Spatiotemporal online allocation with deadline constraints (SOAD)

Problem statement. Consider a decision-maker that manages n points defined on a metric space (X, d), where X denotes the set of points and $d(u, v)$ denotes the distance between any two points $u, v \in X$. In a time-slotted system, the player aims to complete a unit-size workload before a deadline T while minimizing the total service cost by allocating the workload across points and time.

Allocation definition.   The decision-maker specifies a spatial allocation to one of the points $u \in X$. At the chosen point, they also make a temporal allocation that *fractionally* divides the allocation between two states, $\mathsf{ON}^{(u)}$ and $\mathsf{OFF}^{(u)}$, where the allocation to $\mathsf{ON}^{(u)}$ represents the amount of resources actively servicing the workload. Let $\mathbf{x}_t := \{x_t^{\mathsf{ON}^{(u)}}, x_t^{\mathsf{OFF}^{(u)}}\}_{u \in X}$ denote the allocation decision at time t across all points and states, where $x_t^{\mathsf{ON}^{(u)}}$ and $x_t^{\mathsf{OFF}^{(u)}}$ denote the allocation to the ON and OFF states at point u, respectively. The feasible set for this vector allocation is given by $\mathcal{X} := \{\mathbf{x} \subseteq [0, 1]^{2n} : x^{\mathsf{ON}^{(u)}} + x^{\mathsf{OFF}^{(u)}} \in \{0, 1\}, \forall u \in X, \|\mathbf{x}\|_1 = 1\}$.

Deadline constraint.   Let $c(\mathbf{x}_t) : \mathcal{X} \to [0, 1]$ denote a constraint function that is known to the decision-maker and models the fraction of the workload completed by an allocation $\mathbf{x}_t$. Specifically,

we let $c(\mathbf{x}_t) = \sum_{u \in X} c^{(u)} \cdot x_t^{\text{ON}^{(u)}}$, where $c^{(u)}$ is a positive *throughput constant* that encodes how much of the workload is completed during one time slot with a full allocation to the state $\text{ON}^{(u)}$. Across the entire time horizon, the decision-maker is subject to a *deadline constraint* stipulating that the cumulative allocations must satisfy $\sum_{t=1}^{T} c(\mathbf{x}_t) \geq 1$. This encodes the requirement that sufficient allocations must be assigned to the ON states to finish a unit-size workload by the deadline T. The unit size is w.l.o.g. by scaling $c(\cdot)$ appropriately – e.g., if a workload doubles in size, $c(\cdot)$ is scaled by a factor of $1/2$ to reflect this.

Service and switching costs. At each time t, the cost of allocation $\mathbf{x}_t$ consists of a *service cost* $f_t(\mathbf{x}_t) = \sum_{u \in X} f_t^{(u)} \cdot x_t^{\text{ON}^{(u)}}$ for allocations to ON states, where $f_t^{(u)}$ represents the service cost of point u at time t; and a *switching cost* $g(\mathbf{x}_t, \mathbf{x}_{t-1})$ that includes a spatial movement cost of moving the allocation between points and a temporal switching cost incurred between ON and OFF states within one point. Specifically, whenever the decision-maker changes the allocation across points, they pay a movement cost $d(u_{t-1}, u_t)$, where $u_{t'} = \{u \in X : x_{t'}^{\text{ON}^{(u)}} + x_{t'}^{\text{OFF}^{(u)}} = 1\}$ is the location of the allocation at time t' (in Section 5, we give a generalization where $d(\cdot, \cdot)$ also varies with time). Within each point, the decision-maker pays a switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\boldsymbol{\beta})} = \sum_{u \in X} \beta^{(u)} |x_t^{\text{ON}^{(u)}} - x_{t-1}^{\text{ON}^{(u)}}|$, where $\beta^{(u)}$ is the switching overhead factor when changing the ON / OFF allocation at point u. The overall switching cost is $g(\mathbf{x}_t, \mathbf{x}_{t-1}) \coloneqq d(u_{t-1}, u_t) + \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\boldsymbol{\beta})}$, and g is known in advance. The decision-maker starts (at $t = 0$) with a full allocation at some OFF state, and they *must* end (at $t = T+1$) with their full allocation at any OFF state.

Spatiotemporal allocation with deadline constraints. The objective of the player is to minimize the total cost while satisfying the workload's deadline constraint. Let $\mathcal{I} := \{f_t(\cdot)\}_{t \in [T]}$ denote an input sequence of SOAD. For a given $\mathcal{I}$, the offline version of the problem can be formulated as:

$$[\text{SOAD}] \quad \min_{\{\mathbf{x}_t\}_{t \in [T]}} \underbrace{\sum_{t=1}^{T} f_t(\mathbf{x}_t)}_{\text{Service cost}} + \underbrace{\sum_{t=1}^{T+1} g(\mathbf{x}_t, \mathbf{x}_{t-1})}_{\text{Switching cost (e.g., overhead)}} \quad \text{s.t.} \quad \underbrace{\sum_{t=1}^{T} c(\mathbf{x}_t) \geq 1}_{\text{Deadline constraint}}, \quad \mathbf{x}_t \in \mathcal{X}. \quad (1)$$

In the offline setting, we note that the above formulation is convex, implying that it can be solved efficiently using, e.g., iterative methods. However, our aim is to design an online algorithm that chooses an allocation $\mathbf{x}_t$ for each time t without knowing future costs $\{f_{t'}(\cdot)\}_{t'>t}$.

2.2 Motivating applications

In this section, we give examples of applications that motivate the SOAD problem. We are particularly motivated by an emerging class of *carbon-aware sustainability problems* that have attracted significant attention in recent years – see the first example. SOAD also generalizes canonical online search problems such as one-way trading [27], making it broadly applicable across domains as we discuss in detail below. We focus on key components of each setting without exhaustively discussing idiosyncrasies, although we mention some extensions of SOAD in each setting. We defer a few more problem examples to Appendix A.

Carbon-aware workload management in data centers. Consider a delay-tolerant compute job scheduled on a distributed network of data centers with the goal of minimizing the total carbon (CO_2) emissions of the job. Each job arrives with a deadline T that represents its required completion time, typically in minutes or hours. Service costs $f_t^{(u)}$ represent the carbon emissions of executing a workload at full speed in data center u at time t. The metric space (X, d) and the spatial movement cost $d(u_t, u_{t-1})$ capture the *carbon emissions overhead* of geographically migrating a compute workload between data centers. The temporal switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\boldsymbol{\beta})}$ captures the carbon emissions overhead due to reallocation of resources (e.g., scaling up/down) within a single data center [32]. Finally, the constraint function $c(\mathbf{x}_t)$ encodes what fraction of the job is

completed by a given scheduling decision $\mathbf{x}_t$. The topic of shifting compute in time and space to decrease its carbon footprint has seen significant attention in recent years [2, 8, 19, 20, 33, 63, 67, 77], particularly for compute needs with long time scales and flexible deadlines (e.g., ML training), which realize the most benefits from temporal shifting. These works build on a long line of work advancing sustainable data centers more broadly (e.g., in terms of energy efficiency), some of which leverage techniques from online optimization [26, 30, 31, 44–49, 61, 65, 76, 80, 81]. We comment that SOAD is the first online formulation that can model the necessary combined dimensions of spatial and temporal switching costs with deadlines. However, we also note that some aspects of this problem may not yet be fully captured by SOAD– for instance, it might be necessary to consider *multiple concurrent* batch workloads rather than a single one, resource contention, data center capacity constraints, or processing delays caused by migration that are not fully captured by the current formulation. In this sense, SOAD serves as a building block that could accommodate extensions to consider these aspects of the problem – we consider one such extension in Section 5.

Supply chain procurement.   Consider a firm that must source a certain amount of a good before a deadline T, where the good is stored in several regional warehouses [53]. Service costs $f_t^{(u)}$ are proportional to the per unit cost of purchasing and transporting goods from warehouse u during time slot t. The metric space (X, d) and the spatial movement cost $d(u_t, u_{t-1})$ capture the overhead of switching warehouses, including, e.g., personnel costs to travel and inspect goods. The temporal switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\boldsymbol{\beta})}$ captures the overhead of stopping or restarting the purchasing and transport of goods from a single warehouse. Finally, the constraint function $c(\mathbf{x}_t)$ dictates how many goods can be shipped during time t according to purchasing decision $\mathbf{x}_t$. We note that in practice, the firm may need to purchase from multiple warehouses concurrently – they are restricted to purchase from only one in the strict SOAD formulation given above, but this can be relaxed without affecting the algorithms or results that we present in the rest of the paper.

Mobile battery storage.   Consider a mobile battery storage unit (e.g., a battery trailer [60]) that must service several discharge locations by, e.g., the end of the day (deadline T), with the goal of choosing when and where to discharge based on the value that storage can provide in that time and place. Service costs $f_t^{(u)}$ can represent the value of discharging at location u during time slot t (lower is better). The metric space (X, d) and the spatial movement cost $d(u_t, u_{t-1})$ capture the *overhead* (e.g., lost time or fuel cost) due to moving between locations, and the temporal switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\boldsymbol{\beta})}$ captures the overhead of connecting or disconnecting from a discharge point at a single location, including e.g., cell degradation due to cycling [79]. The constraint function $c(\mathbf{x}_t)$ captures how much energy has been discharged during time t according to decision $\mathbf{x}_t$. The problem of maximizing the utility that mobile battery storage provides may be useful in e.g., emergency relief situations where the main power grid has gone down. A light extension of SOAD may capture the case where the travel time from point to point significantly affects the feasible discharge time at the destination; such an extension would factor any lost time into the constraint function $c(\mathbf{x}_t)$.

2.3   Background & assumptions

In this section, we provide background on the competitive analysis used throughout the paper and formalize our assumptions on the costs in SOAD, motivated by the structure of applications.

Competitive analysis.   We evaluate the performance of an online algorithm for this problem via the *competitive ratio* [10, 55]: let $\text{OPT}(\mathcal{I})$ denote the cost of an optimal offline solution for instance $\mathcal{I}$, and let $\text{ALG}(\mathcal{I})$ denote the cost incurred by running an online algorithm ALG over the same instance. Then the competitive ratio of ALG is defined as $\text{CR(ALG)} \coloneqq \sup_{\mathcal{I} \in \Omega} \text{ALG}(\mathcal{I})/\text{OPT}(\mathcal{I}) =: \eta$, where Ω is the set of all feasible inputs for the problem, and ALG is said to be η-*competitive*. Note that CR(ALG) is always at least 1, and a *smaller* competitive ratio implies that the online algorithm

is guaranteed to be *closer* to the offline optimal solution. If ALG is randomized, we replace the cost ALG($\mathcal{I}$) with the expected cost (over the randomness of the algorithm).

ASSUMPTION 2.1. *Each service cost function $f_t(\cdot)$ satisfies bounds, i.e., $f_t^{(u)} \in [c^{(u)}L, c^{(u)}U]$ for all* ON$^{(u)}$: $u \in X$ *and for all $t \in [T]$, where L and U are known, positive constants.*

This assumption encodes the physical idea that there exist upper and lower bounds on the service cost faced by the player. L and U are normalized by the throughput coefficient $c^{(a)}$ so that they can be independent of the amount of the deadline constraint satisfied by servicing the workload at a specific point $a \in X$.

ASSUMPTION 2.2. *The temporal switching cost factor is bounded by $\beta^{(a)} \leq \tau c^{(a)}$ for all $a \in X$. The normalized spatial distance between any two* ON *states is upper bounded by D, i.e., $D = \sup_{u,v \in X: u \neq v} \frac{d(u, v)}{\min\{c^{(u)}, c^{(v)}\}}$. Further, we assume that $D + 2\tau \leq U - L$.*

In SOAD, τ represents the worst-case overhead of stopping, starting, or changing the rate of service at a single point, while D represents the worst-case overhead incurred by moving the allocation between the two most distant points. In, e.g., the applications mentioned above, we typically expect τ to be much smaller than D. Note that there are two ON states with a normalized distance greater than $U - L$, one of these states should be pruned from the metric, because moving the allocation between them would negate any benefit to the service cost. Specifically, recall that L and U give bounds on the total service cost of the workload, and consider an example of two points u, v that are normalized distance $D' > U - L$ apart, with $c^{(u)} = c^{(v)} = 1$, and $\tau = 0$. Let the starting point u and other point v have service costs that are the worst-possible and best-possible (i.e., U and L), respectively. Observe that if the player opts to move the allocation to v, they incur a movement cost of D' and a total objective of $L + D'$. In contrast, if they had stayed at point u, their total objective would be U, which is $< L + D'$ by assumption.

2.4 Connections to existing models and challenges

As discussed in the related work, SOAD exhibits similarities to two long-standing tracks of literature in online algorithms; however, SOAD is distinct from and cannot be solved by existing models.

The first of these is the work on the classic *metrical task systems* (MTS) problem introduced by Borodin et al. [10] and related forms, including smoothed online convex optimization (SOCO) [18]. In these works, an online player makes decisions with the objective of minimizing the sum of the service cost and switching cost. However, standard algorithms for MTS/SOCO are not designed to handle the type of long-term constraints (e.g., such as deadlines) that SOAD considers. Moreover, standard MTS and SOCO algorithms are designed to address either movement cost over points (i.e., $d(a_{t-1}, a_t)$) or temporal switching cost (i.e., $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\boldsymbol{\beta})}$). SOAD requires a spatiotemporal allocation that considers both types of switching costs simultaneously.

On the other hand, the *one-way trading* (OWT) problem introduced by El-Yaniv et al. [27] and related online knapsack problems [70, 82] consider online optimization with long-term constraints. To address these constraints, canonical algorithms use techniques such as threat-based or threshold-based designs to "hedge" between the extremes of quickly fulfilling the constraint and waiting for better opportunities that may not materialize. However, these works do not consider switching costs and rarely address multidimensional decision spaces.

The design of algorithms via competitive analysis for MTS/SOCO with long-term constraints has long been an open problem, and has seen only limited progress in a few recent works. These works have primarily leveraged techniques from online search, generalizing unidimensional problems such as k-search [42] and OWT [12, 41] to additionally consider a *temporal switching cost*. A recent study considered a generalization to the multidimensional case, introducing the problem of

convex function chasing with a long-term constraint (CFL) [40]. The authors of this work propose a competitive algorithm for CFL, although their results depend on a very specific metric structure (ℓ_1 vector spaces or weighted star metrics), which cannot be used to model the general spatial and temporal switching costs in SOAD. Furthermore, even in the multidimensional case, these existing works that consider switching costs and long-term constraints assume a *single source* of switching costs, i.e., that the cost to switch into a state making progress towards the long-term constraint is the same as the cost to switch out of that state. This type of structure and analysis fails in the SOAD setting due to the generality of the metric (i.e., moving to a new point while "switching ON" to complete some of the workload and "switching OFF" within the new point have different costs).

Worst-case competitive guarantees provide robustness against non-stationarities in the underlying environment, which may be desired for applications such as carbon-awareness due to the demonstrated non-stationarity associated with such signals [67]. However, algorithms that are purely optimized for worst-case guarantees are often overly pessimistic. To address this, we study learning-augmented design in the SOAD setting, which brings additional challenges. For instance, existing learning-augmented results for MTS/SOCO and OWT each leverage distinct algorithm design strategies based on different advice models that separately address features of their problem setting (i.e., switching costs, deadline constraints). These prior results naturally prompt questions about how to incorporate ML advice in a performant way that can simultaneously handle the generality of the switching costs in SOAD while ensuring that the deadline constraint is satisfied.

2.5 Preliminaries of technical foundations

In this section, we introduce and discuss techniques from different areas of the online algorithms literature that we use in subsequent sections to address the SOAD challenges discussed above.

Unifying arbitrary metrics. The generality of the metric space in SOAD is a key challenge that precludes the application of algorithm design techniques from prior work requiring specific metric structures. In classic MTS, the online player also makes decisions in an arbitrary metric space (X, d), which poses similar challenges for algorithm design. A key result used to address this is that of Fakcharoenphol et al. [29], who show that for any n-point metric space (X, d), there exists a probabilistic embedding into a *hierarchically separated tree* (HST) $\mathcal{T} = (V, E)$ with at most $O(\log n)$ distortion, i.e., $\mathbb{E}_{\mathcal{T}}\left[d^{(\mathcal{T})}(u, v)\right] \leq O(\log n)d(u, v)$ for any $u, v \in X$. For MTS, this result implies that any η-competitive algorithm for MTS *on trees* is immediately $O(\log n)\eta$-competitive in expectation for MTS on general n-point metrics, exactly by leveraging this embedding.

To solve MTS using such a tree, Bubeck et al. [16] consider a randomized algorithm on the leaves of $\mathcal{T}$ denoted by $\mathcal{L}$, where the nodes of $\mathcal{L}$ correspond to points in X. This randomized metric space is given by $(\Delta_{\mathcal{L}}, \mathbb{W}^1)$, where $\Delta_{\mathcal{L}}$ is the probability simplex over the leaves of $\mathcal{T}$, and $\mathbb{W}^1$ denotes the *Wasserstein-1 distance.* For two probability distributions $\mathbf{p}, \mathbf{p}' \in \Delta_{\mathcal{L}}$, the Wasserstein-1 distance is defined as $\mathbb{W}^1(\mathbf{p}, \mathbf{p}') \coloneqq \min_{\pi_{x,x'} \in \Pi(\mathbf{p}, \mathbf{p}')} \mathbb{E}\left[d(x, x')\right]$, where $\Pi(\mathbf{p}, \mathbf{p}')$ is the set of transport distributions over $\mathcal{L}^2$ with marginals $\mathbf{p}$ and $\mathbf{p}'$. A randomized algorithm that produces marginal distributions $\mathbf{p} \in \Delta_{\mathcal{L}}$ then *couples* consecutive decisions according to the optimal transport plan $\pi_{x,x'}$ defined by Wasserstein-1. Bubeck et al. [16] further show that $(\Delta_{\mathcal{L}}, \mathbb{W}^1)$ is bijectively isometric to a convex set K with a weighted ℓ_1 norm $\| \cdot \|_{\ell_1(\mathbf{w})}$ based on edge weights in the tree.

Metric tree embedding for SOAD.   Prior approaches by Fakcharoenphol et al. [29] and Bubeck et al. [16] are able to manage the spatial movement cost from moving allocation between points. To further accommodate the temporal switching cost between ON and OFF states within a single point, we develop a probabilistic tree embedding $\mathcal{T} = (V, E)$ and the corresponding vector space $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ in the following Definition 2.3 and Definition 2.4, respectively.

DEFINITION 2.3 (PROBABILISTIC TREE EMBEDDING $\mathcal{T} = (V, E)$ FOR SOAD). *Let (X, d) denote the underlying metric space over n points, and let $\mathcal{T}'$ denote an HST constructed on the points of X according to the method by Fakcharoenphol et al. [29]. Label the leaves of $\mathcal{T}'$ according to the n ON states, one for each point. Then the final tree $\mathcal{T}$ is constructed by adding n edges and n nodes to just the leaves of $\mathcal{T}'$ – each new node represents the corresponding OFF state at that point, and the new edge is weighted according to the temporal switching cost at that point (i.e., $\beta^{(u)}$). The resultant "state set" $\mathcal{S}$ includes both the leaves of $\mathcal{T}$ (OFF states) and their immediate predecessors (ON states).*
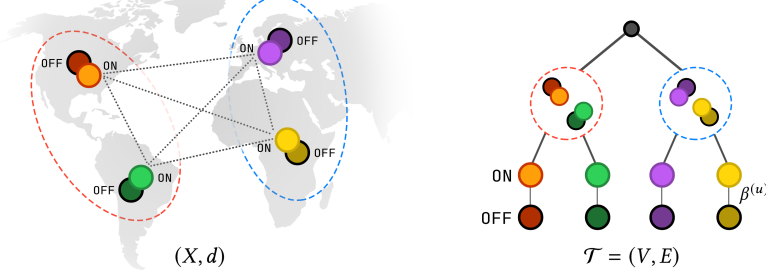

Fig. 1. An illustration of the probabilistic tree embedding (Def. 2.3) for the motivating application. Points in the metric are represented as pairs of circles on the left. On the right, the first three levels of the tree approximate the metric space (X, d) [29], and the last level captures the ON / OFF structure of SOAD.

Note that $\mathcal{T}$ preserves distances between the points of X with expected $O(\log n)$ distortion, while the switching cost between ON and OFF states at a single point is preserved exactly. Our competitive algorithm (see Section 3) operates in a vector space constructed according to this HST embedding.

DEFINITION 2.4 (VECTOR SPACE $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$). *Given a hierarchically separated tree $\mathcal{T} = (V, E)$ constructed according to Definition 2.3, with root $r \in V$, state set $\mathcal{S} \subseteq V$, and leaf set $\mathcal{L} \subset \mathcal{S}$, let $P^{(u)}$ denote the parent of any node $u \in V \setminus r$. Construct the following set:*

$$K \coloneqq \left\{ \mathbf{k} \in \mathbb{R}^{|V|} : \mathbf{k}^{(r)} = 1, \text{ and } \forall u \in V \setminus \mathcal{L}, \mathbf{k}^{(u)} = \sum_{v : P^{(v)} = u} \mathbf{k}^{(v)}, \text{ and } \forall u \in \mathcal{S}, \mathbf{k}^{(u)} \in [0, 1] \right\}.$$

Let $\mathbf{w}$ be a non-negative weight vector on vertices of $\mathcal{T}$, where $\mathbf{w}^{(r)} = 0$ and $\mathbf{w}^{(u)} > 0$ for all $u \in V \setminus r$. Recall that the edges of $\mathcal{T}$ are weighted – we let $\mathbf{w}^{(u)}$ denote the weight of edge $\{P^{(u)}, u\}$, and define a weighted ℓ_1 norm as $\|\mathbf{k}\|_{\ell_1(\mathbf{w})} \coloneqq \sum_{u \in V} \mathbf{w}^{(u)} |\mathbf{k}^{(u)}|$, for any $\mathbf{k} \in K$. Finally, we define a linear map from $\Delta_{\mathcal{S}}$ to K (and its corresponding inverse), given by a matrix map $\Phi : \mathbb{R}^{2n} \to \mathbb{R}^{|V|}$. We let $A^{(u)}$ denote the set of node u's ancestors in $\mathcal{T}$, and (with a slight abuse of notation) let $\mathcal{S}(i) : \{0, \ldots, 2n\} \to \mathcal{S}$ and $K(i) : \{0, \ldots, |V|\} \to V$ denote indexing maps that recover the object in $\mathcal{S}$ or V, respectively. Then $\Phi \in \mathbb{R}^{|V| \times 2n}$ and $\Phi^{-1} \in \mathbb{R}^{2n \times |V|}$ are defined as follows:

$$\Phi_{i,j} \coloneqq \begin{cases} 1 & \text{if } K(i) = r \\ 1 & \text{if } K(i) = \mathcal{S}(j) \\ 1 & \text{if } K(i) \in \mathcal{S} \text{ and } P^{(\mathcal{S}(j))} = K(i) \\ 1 & \text{if } K(i) \in V \setminus \mathcal{S} \text{ and } K(i) \in A^{(\mathcal{S}(j))} \\ 0 & \text{otherwise} \end{cases} \qquad \Phi_{i,j}^{-1} \coloneqq \begin{cases} 1 & \text{if } \mathcal{S}(i) = K(j) \\ -1 & \text{if } K(j) \in \mathcal{L} \text{ and } P^{(K(j))} = \mathcal{S}(i) \\ 0 & \text{otherwise} \end{cases}$$

In words, Φ maps a distribution over $\Delta_{\mathcal{S}}$ to the corresponding vector in K by accumulating probability mass upwards from the leaves of $\mathcal{T}$ towards the root. Φ^{-1} reverses this by selecting the appropriate indices for $u \in \mathcal{S}$ from K, and recovers probabilities by subtracting the mass at the OFF state from the ON state (since the OFF state is a leaf, the ON state accumulates its probability in K).

Randomized algorithm for SOAD. We define some shorthand notation. For a decision $\mathbf{k} \in K$, $\mathbf{p} = \Phi^{-1}\mathbf{k}$ gives a corresponding probability distribution on $\Delta_{\mathcal{S}}$. Note that $\mathcal{X} \subset \Delta_{\mathcal{S}} \subset \mathbb{R}^{2n}$, and by

linearity of expectation, the service and constraint functions $f_t(\cdot), c(\cdot) : \mathcal{X} \to \mathbb{R}$ remain well-defined (in expectation) on $\Delta_{\mathcal{S}}$. Within K, we let $\overline{f}_t(\mathbf{k}) = f_t(\Phi^{-1}\mathbf{k})$ and $\overline{c}(\mathbf{k}) = c(\Phi^{-1}\mathbf{k})$. For a given *starting point* $s \in X$, we slightly abuse notation and let $\delta_s \in \Delta_{\mathcal{S}}$ denote the Dirac measure supported at $\mathsf{OFF}^{(s)}$. Recall that the SOAD formulation specifies an allocation that is *discrete* in terms of choosing a point in the metric, and *fractional* in terms of the resource allocation at a given point. To capture this structure while using the embedding results discussed above, we consider a *mixed setting* that is probabilistic in spatial assignment but deterministic in the ON / OFF allocation. We state the equivalence of this setting and the fully probabilistic one below, deferring the proof to Appendix C.

THEOREM 2.5. *For a randomized SOAD decision $\mathbf{p}_t \in \Delta_{\mathcal{S}}$, the expected cost is equivalent if a point in X is first chosen probabilistically and the ON / OFF probabilities at that point are interpreted as (deterministic) fractional allocations.*

Enforcing a deadline constraint using pseudo-cost. Existing algorithms for MTS-type problems are not designed to handle a deadline constraint while remaining competitive. For SOAD, we draw from the *pseudo-cost minimization* [70] approach for online search problems, where the player is subject to a long-term *buying/selling* constraint that poses similar algorithmic challenges.

Under the pseudo-cost framework, we start by assuming that a *mandatory allocation* condition exists to strictly enforce the deadline constraint. Let $z^{(t)}$ denote the fraction of the deadline constraint satisfied (in expectation) up to time t (we henceforth call this the **utilization**). To avoid violating the constraint, a mandatory allocation begins at time j, as soon as $(T - (j+1))c^{(u)} < (1 - z^{(j)}) \forall u \in X$, i.e., when the remaining time after the current slot would be insufficient to satisfy the constraint. Note that in practice, $z^{(j)}$ would be replaced by the actual constraint satisfaction so far. During the mandatory allocation, a cost-agnostic player takes control and makes maximal allocation decisions to ensure the workload is finished before the deadline.

Intuitively, the mandatory allocation complicates competitive analysis – in the worst-case, an adversary can present the worst service cost (U) during the final steps. The key idea behind pseudo-cost minimization is to rigorously characterize a *trade-off* between completing the constraint "too early" and waiting too long (i.e., risking a mandatory allocation) using a *pseudo-cost function*. Such a function takes the lower and upper bounds on service cost (i.e., L and U) as parameters, and assigns a *pseudo-cost* to each increment of progress towards the constraint. In an algorithm, this function is used by solving a small minimization problem at each step, whose objective considers the true cost of a potential decision and an integral over the pseudo-cost function – generating decisions using this technique creates a connection between the utilization and the best service cost encountered throughout the sequence, ensuring that the algorithm completes "exactly enough" of the constraint before mandatory allocation in order to achieve a certain competitive ratio against the best service cost, which is a lower bound on OPT. We also note that in the learning-augmented setting, the pseudo-cost minimization problem can be combined with a *consistency constraint*, as shown by [40], to integrate certain forms of advice without losing the robust (i.e., competitive) qualities of the pseudo-cost.

3 PCM: A Competitive Online Algorithm

This section presents a randomized competitive algorithm for SOAD that leverages the metric tree embeddings and pseudo-cost minimization design discussed above in Section 2.5. We further show that our algorithm achieves a competitive ratio that is optimal for SOAD up to log factors.

Algorithm 1 Pseudo-cost minimization algorithm for SOAD (PCM)

---

input: constraint function $c(\cdot)$, convex set K with distance metric $\|\cdot\|_{\ell_1(\mathbf{w})}$, pseudo-cost function $\psi(z)$, starting OFF state $s \in \mathcal{S}$.

initialize: $z^{(0)} = 0$; $\mathbf{k}_0 = \Phi\delta_s$; $\mathbf{p}_0 = \delta_s$.

while cost function $f_t(\cdot)$ is revealed and $z^{(t-1)} < 1$ **do**

  Solve **pseudo-cost minimization** problem:

$$\mathbf{k}_t = \underset{\mathbf{k} \in K : \bar{c}(\mathbf{k}) \leq 1 - z^{(t-1)}}{\arg\min} \overline{f}_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\bar{c}(\mathbf{k})} \psi(u)du, \tag{2}$$

$$\mathbf{p}_t = \Phi^{-1}\mathbf{k}_t. \tag{3}$$

  Update utilization $z^{(t)} = z^{(t-1)} + c(\mathbf{p}_t)$.

3.1 Algorithm description

We present a randomized pseudo-cost minimization algorithm (PCM) in Algorithm 1. PCM operates on the metric space $(K, \|\cdot\|_{\ell_1(\mathbf{w})})$ defined in Definition 2.4 and extends the original pseudo-cost minimization framework [70] to address the setting where the decision space is given by an arbitrary convex set K with distances given by $\|\cdot\|_{\ell_1(\mathbf{w})}$.

We define a pseudo-cost function $\psi(z) : [0, 1] \to [L, U]$, where z is the utilization (i.e., the completed fraction of the deadline constraint in expectation). Our construction of ψ takes advantage of additional structure in the SOAD setting – this function depends on the parameters of the SOAD problem, including U, L, D, and τ specified in Assumptions 2.1 and 2.2.

DEFINITION 3.1 (PSEUDO-COST FUNCTION ψ FOR SOAD). *For a given parameter $\eta > 1$, the pseudo-cost function is defined as* $\psi(z) = U - \tau + (U/\eta - U + D + \tau) \exp(z/\eta), z \in [0, 1]$.

Given the pseudo-cost function from Def. 3.1, PCM solves a minimization problem (2) at each step t to generate a decision $\mathbf{k}_t \in K$; the objective of this problem is to minimize a combination of the per-step cost plus a pseudo-cost term that encourages (deadline) constraint satisfaction. At a high level, the ψ term enforces that $\mathbf{k}_t$ satisfies "exactly enough" of the deadline constraint (in expectation) to make adequate progress and maintain an expected competitive ratio of η against the current estimate of OPT, without "overbuying" and preventing better costs from being considered in the future. At a glance, it is not obvious that the pseudo-cost minimization problem is straightforward to actually solve in practice. In the following, we show that (2) is a convex minimization problem.

THEOREM 3.1. *Under the assumptions of* SOAD, *the pseudo-cost minimization* (2) *is a convex minimization problem.*

We defer the proof of Theorem 3.1 to Appendix D.1. At a high-level, the result implies that the solution to (2) can be found efficiently using convex programming techniques [13, 25]. Compared to prior works [40, 70], our design of ψ differentiates between the spatial movement cost and temporal switching cost (in particular, D only appears within the exponential, while τ appears inside and outside of the exponential term). This removes a source of pessimism – when PCM makes a decision to move to a distant point in the metric (i.e., paying a worst-case factor of D), it can safely assume that it will only have to pay a factor of τ to switch OFF if the next cost function is bad. This allows PCM to take advantage of spatially distributed OFF states, where all of the existing works that use a pseudo-cost paradigm for temporal load-shifting cannot.

We note that PCM's decisions in K are marginal probability distributions over $\Delta_\mathcal{S}$ – we briefly detail how feasible deterministic decisions in $\mathcal{X}$ are extracted from these outputs. We assume the player interprets distributions according to a mixed random / fractional setting (see Theorem 2.5), allowing them to make fractional resource allocation decisions within a single point while the

allocation is probabilistically assigned to a single point. At each time step, PCM generates $\mathbf{p}_t \in \Delta_S$. We let $\mathbf{r}_t \coloneqq \{r_t^{(u)} \leftarrow p_t^{\text{ON}(u)} + p_t^{\text{OFF}(u)} : u \in X\}$ aggregate the ON / OFF probabilities at each location of X. Given this *spatial* distribution over X, consecutive decisions should be jointly distributed according to the optimal transport plan between $\mathbf{r}_{t-1}$ and $\mathbf{r}_t$, given by $(\mathbf{r}_t, \mathbf{r}_{t-1}) \sim \pi_t \coloneqq \arg\min_{\pi \in \Pi(\mathbf{r}_t, \mathbf{r}_{t-1})} \mathbb{E}\left[d(u_t, u_{t-1})\right]$, where $(u_t, u_{t-1}) \sim \pi_t$ and $\Pi(\mathbf{r}_t, \mathbf{r}_{t-1})$ is the set of distributions over X^2 with marginals $\mathbf{r}_t$ and $\mathbf{r}_{t-1}$. If the decision-maker couples decisions according to π_t, then the *expected* spatial movement cost of the deterministic decisions is equivalent to $\mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1})$, the spatial Wasserstein-1 distance between $\mathbf{r}_t$ and $\mathbf{r}_{t-1}$. Given a previous deterministic point assignment u_{t-1}, the player can obtain the point assignment u_t by sampling through the conditional distribution $\pi_t(u_t|u_{t-1})$. The fractional ON / OFF allocation in $\mathbf{x}_t$ at the chosen location $u_t \in X$ is then given by $p_t^{\text{ON}(u)}/r_t^{(u)}$ and $p_t^{\text{OFF}(u)}/r_t^{(u)}$, respectively; by Theorem 2.5, this gives that $\mathbb{E}\left[g(\mathbf{x}_t, \mathbf{x}_{t-1})\right] = \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1})$.

3.2 Main results

In Theorem 3.2, we state a bound on the competitive ratio of PCM.

THEOREM 3.2. *Under Assumptions 2.1 and 2.2,* PCM *is $O(\log n)\eta$-competitive for* SOAD*, where η is the solution to* $\ln\left(\frac{U-L-D-2\tau}{U-U/\eta-D}\right) = \frac{1}{\eta}$ *and given by:*

$$\eta \coloneqq \left[W\left(\frac{(D+L-U+2\tau)\exp\left(\frac{D-U}{U}\right)}{U}\right) + \frac{U-D}{U}\right]^{-1}, \tag{4}$$

where W is the Lambert W function [24], and the $O(\log n)$ factor is due to the tree embedding [29].

Compared to previous works such as MTS and OWT, the competitive bound in Theorem 3.2 compares favorably. In particular, the upper bound is *better* than one might expect from e.g., combining the bounds of MTS and OWT. For the minimization variant of OWT, the optimal competitive ratio due to Lorenz et al. [50] is $\left[W\left((1/\theta - 1)e^{-1}\right) + 1\right]^{-1}$, where θ is defined as M/m, and $M \geq m$ are bounds on the prices (i.e., $(m, M) \approx (L, U)$). For MTS, the randomized state-of-the-art due to Bubeck et al. [16] is $O((\log n)^2)$. Asymptotically, compared to both of these bounds, η "loses" a log factor depending on the number of points in the metric, and it is known that $W(x) \sim \ln(x)$ as $x \to \infty$ [35, 66]. Compared to OWT, η adds a dependency on D and τ, parameters describing the cost due to the metric and switching, but we note that Assumption 2.2 (i.e., bounds on D and τ in terms of U and L) prevents the competitive ratio from significantly increasing.

Given the result in Theorem 3.2, a natural question is whether any online algorithm for SOAD can achieve a better competitive bound. We answer this in the negative, showing that PCM's competitive ratio is the best achievable up to log factors that are due to the metric embedding. In particular, we show a class of difficult instances on which no algorithm can achieve a competitive ratio better than η – since the definition of the competitive ratio covers all valid inputs, this gives a corresponding lower bound on the competitive ratio of any algorithm for SOAD.

THEOREM 3.3. *For any U, L, τ, and $D \in [0, (U - L))$, there exists a set of* SOAD *instances on a weighted star on which no algorithm* ALG *can achieve* ALG/OPT *better than η (for η defined in (4)).*

3.3 Proof overviews

We give proof sketches of Theorems 3.2 and 3.3, relegating the full proofs of both to Appendix D.

Proof Sketch of Theorem 3.2. To show this result, we give two lemmas to characterize the cost of OPT and the expected cost of PCM, respectively. First, note that the solution given by PCM is

feasible, by definition of the mandatory allocation (i.e., $\sum_{t=1}^{T} c(\mathbf{p}_t) = 1$). On an arbitrary SOAD instance $\mathcal{I} \in \Omega$, we denote the final utilization (before the mandatory allocation) by $z^{(j)}$.

LEMMA 3.4. *The offline optimum is lower bounded by* $\text{OPT}(\mathcal{I}) \geq \frac{\max\{\psi(z^{(j)}) - D, L\}}{O(\log n)}$.

We can show by contradiction that for any instance, the definition of the pseudo-cost minimization enforces that $\psi(z^{(j)}) - D$ is a lower bound on the *best service cost* seen in the sequence (see (12)). Note that the best choice for OPT is to service the entire workload at the minimum cost (if it is feasible). This yields a corresponding lower bound on OPT – formally, $\text{OPT}(\mathcal{I}) \geq \max\{\psi(z^{(j)}) - D, L\}/O(\log n)$, where the log factor appears due to the distortion in the metric tree embedding.

LEMMA 3.5. *PCM's expected cost is bounded by* $\mathbb{E}[\text{PCM}(\mathcal{I})] \leq \int_0^{z^{(j)}} \psi(u)du + (1 - z^{(j)})U + \tau z^{(j)}$.

The definition of the pseudo-cost provides an automatic bound on the expected cost incurred during any time step where progress is made towards the deadline constraint (i.e., whenever the service cost is non-zero). We show that $\tau z^{(j)}$ is an upper bound on the *excess cost* that can be incurred by PCM in the other time steps (i.e., due to temporal switching costs, see (13)). Summing over all time steps, this gives that the expected cost of PCM is upper bounded by $\int_0^{z^{(j)}} \psi(u)du + (1 - z^{(j)})U + \tau z^{(j)}$, where $(1 - z^{(j)})U$ is due to the mandatory allocation.

Combining the two lemmas and using the definition of the pseudo-cost function to observe that $\int_0^{z^{(j)}} \psi(u)du + (1 - z^{(j)})U + \tau z^{(j)} \leq \eta \left[\psi(z^{(j)}) - D\right]$ (see (24)) completes the proof. □

Proof Sketch of Theorem 3.3. In Definition D.3, we define a class of *y-adversaries* denoted by $\mathcal{G}_y$ and $\mathcal{A}_y$ for $y \in [L, U]$, along with a corresponding weighted star metric X that contains n points, each with 2 states (ON and OFF), where the distance between any two points in the metric is exactly D. These adversaries present cost functions at the ON states of X in an adversarial order that forces an online algorithm to incur a large switching cost. The $\mathcal{G}_y$ adversary presents a cost function at each step that is "bad" (i.e., U) in all ON states except for one which is *not* at the starting point or the current state of online algorithm ALG. The $\mathcal{A}_y$ adversary starts by exactly mimicking $\mathcal{G}_y$ and presenting "good" cost functions at distant points, before eventually presenting "good" cost functions at the starting point. Both adversaries present "good" cost functions in an adversarial non-increasing order, such that the optimal solutions approach y – formally, $\text{OPT}(\mathcal{G}_y) \rightarrow \min\{y + D + \tau, U\}$, and $\text{OPT}(\mathcal{A}_y) \rightarrow y$. By competing against both adversaries simultaneously, this construction captures a trade-off between being too eager/reluctant to move away from the starting point.

Under this special metric and class of adversaries, the cost of any (potentially randomized) online algorithm ALG can be fully described by two arbitrary *constraint satisfaction functions* $s(y), t(y) : [L, U] \rightarrow [0, 1]$ (see (27)), where each function corresponds one of two *stages* of the adversary (i.e., "good" cost functions arriving at spatially distant points, or at the starting point). For ALG to be $\eta^\star$-competitive (where $\eta^\star$ is unknown), we give corresponding conditions on $s(y)$ and $t(y)$ expressed as differential inequalities (see (30)). By applying Grönwall's Inequality [58, Theorem 1, p. 356], this gives a *necessary condition* such that $\eta^\star$ must satisfy: $\eta^\star \ln \left(\frac{U - L - D - 2\tau}{U - U/\eta^\star - D} \right) - \frac{\eta^\star D + \eta^\star 2\tau}{U/\eta^\star - U + D} \leq s(L) \leq 1 - t(L) \leq 1 - \frac{\eta^\star D + \eta^\star 2\tau}{U/\eta^\star - U + D}$. The optimal $\eta^\star$ is obtained by solving for the transcendental equation that arises when the inequalities are binding, yielding the result. □

4 ST-CLIP: A Learning-Augmented Algorithm

In this section, we consider how a learning-augmented algorithm for SOAD can leverage *untrusted advice* to improve on the average-case performance of PCM while retaining worst-case guarantees. For learning-augmented algorithms, competitive ratio is interpreted via the notions of *consistency* and *robustness* [39, 52]. Letting ALG denote a learning-augmented online algorithm provided with advice denoted by ADV, ALG is said to be α-**consistent** if it is α-competitive with respect to ADV, and

γ-**robust** if it is γ-competitive with respect to OPT when given any advice (i.e., regardless of ADV's performance). We present ST-CLIP (see Algorithm 2), which uses an adaptive optimization-based approach combined with the robust design of PCM to achieve an optimal consistency-robustness trade-off. We start by formally defining the advice model we use below.

DEFINITION 4.1 (BLACK-BOX ADVICE MODEL FOR SOAD). *For a given* SOAD *instance* $I \in \Omega$, *we let* ADV(I) *denote* **untrusted black-box decision advice**, *i.e.,* ADV$(I) \coloneqq \{\mathbf{a}_t \in \Delta_S : t \in [T]\}$. *If* ADV *is correct, a player that plays* $\mathbf{a}_t$ *at each step attains the optimal solution (*ADV$(I) = $ OPT(I)*).*

Although $\mathbf{a}_t$ is defined on the probability simplex Δ_S, a deterministic ADV at time t is given by combining the Dirac measure supported at a point and a specific ON / OFF allocation. We henceforth assume that ADV is *feasible*, satisfying the constraint $(\sum_{t=1}^T c(\mathbf{a}_t) \geq 1)$. While it is not obvious a machine learning model could directly provide such feasible predictions, in practice, we leverage the black-box nature of the definition to combine e.g., machine-learned predictions of relevant costs with a post-processing pipeline that solves for a *predicted optimal solution* (see Section 6.1).

4.1 ST-CLIP: an optimal learning-augmented algorithm

We present ST-CLIP (**s**patio**t**emporal **c**onsistency-**li**mited **p**seudo-cost minimization, Algorithm 2), which exactly matches a lower bound on the optimal robustness-consistency trade-off (Theorem 4.4) for SOAD. ST-CLIP takes a hyperparameter $\varepsilon \in (0, \eta - 1]$, which parameterizes a trade-off between following the untrusted advice ($\varepsilon \to 0$) and prioritizing robustness ($\varepsilon \to \eta - 1$). We start by defining a *target robustness factor* $\gamma^{(\varepsilon)}$, which is the unique solution to the following equation:

$$\gamma^{(\varepsilon)} = \varepsilon + \frac{U}{L} - \frac{\gamma^{(\varepsilon)}(U - L + D)}{L} \ln \left(\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau} \right). \tag{5}$$

We note that $\gamma^{(\varepsilon \to 0)} \to U/L$, which is a trivial competitive ratio for any mandatory allocation scheme (i.e., if the entire constraint is satisfied at the deadline for the worst price U). The precise value of $\gamma^{(\varepsilon)}$ originates from a robustness-consistency lower bound (Theorem 4.4), and ST-CLIP uses it to define a pseudo-cost function $\psi^{(\varepsilon)}$ that enforces $\gamma^{(\varepsilon)}$-robustness in its decisions.

DEFINITION 4.2 (PSEUDO-COST FUNCTION $\psi^{(\varepsilon)}$ FOR SOAD). *For* $\rho \in [0, 1]$ *and* $\gamma^{(\varepsilon)}$ *given by* (5), *let* $\psi^{(\varepsilon)}(\rho)$ *be defined as:* $\psi^{(\varepsilon)}(\rho) = U + D - \tau + (U+D/\gamma^{(\varepsilon)} - U + D + \tau) \exp(\rho/\gamma^{(\varepsilon)})$.

Similarly to PCM (see Section 3), $\psi^{(\varepsilon)}$ is used in a minimization problem solved at each time step to obtain a decision. However, since ST-CLIP must also consider the actions of ADV, it follows the consistency-limited pseudo-cost minimization paradigm, which places a *consistency constraint* on the aforementioned minimization. This constraint enforces that ST-CLIP always satisfies $(1 + \varepsilon)$-consistency, which is salient when ADV is close to optimal. Within this feasible set, the pseudo-cost minimization drives ST-CLIP towards decisions that are "as robust as possible."

Additional challenges in algorithm design. In contrast to prior applications of the CLIP technique [40], the SOAD setting introduces a disconnect between the advice and the robust algorithm (e.g., PCM); specifically, ADV furnishes decisions that are supported on the (randomized) metric Δ_S, while PCM makes decisions on the tree metric given by $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$. Since the CLIP technique effectively "combines" ADV with a robust algorithm, this poses a challenge in the SOAD setting, introducing a $O(\log n)$ dependency in the consistency bound.[1] With ST-CLIP (see Algorithm 2), we carefully decouple the "advice side" and the "robust side" of the CLIP technique to achieve a $(1+\varepsilon)$-consistency bound. While a $O(\log n)$ factor is likely unavoidable on arbitrary metrics in the adversarial setting of robustness (e.g., as is the case for metrical task systems [16, 22]), the

---

[1]Directly applying the CLIP technique to the $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ decision space considered in PCM yields an unremarkable consistency upper bound of $O(\log n)(1 + \varepsilon)$, due to the distortion in the tree metric.

non-adversarial setting of consistency (i.e., when advice is correct) implies that such a factor should be avoidable. Furthermore, removing a factor of $O(\log n)$ allows ST-CLIP to achieve consistency arbitrarily close to 1, which is often desirable in practice when the advice is often of high quality.

To accomplish this decoupling, ST-CLIP uses the pseudo-cost minimization defined in (6) to generate intermediate "robust decisions" ($\mathbf{k}_t \in K$) on the tree embedding (see Def. 2.3). These decisions are converted into marginal probability distributions on the underlying simplex (i.e., $\mathbf{p}_t \in \Delta_\mathcal{S}$) *before* evaluating the consistency constraint. Since ADV also specifies decisions on $\Delta_\mathcal{S}$, this decoupling allows the constraint to directly compare the running cost of ST-CLIP and ADV, without losing a $\log(n)$ factor due to the tree embedding. To hedge against worst-case scenarios that might cause ST-CLIP to violate the desired $(1 + \varepsilon)$-consistency, the consistency constraint in (7) extrapolates the cost of such scenarios on the randomized decision space $\Delta_\mathcal{S}$.

Notation.   We introduce some shorthand notation to simplify the algorithm's pseudocode as follows: we let SC_t denote the expected cost of ST-CLIP's decisions up to time t, i.e., $\text{SC}_t := \sum_{j=1}^{t} f_j(\mathbf{p}_j) + \mathbb{W}^1(\mathbf{p}_j, \mathbf{p}_{j-1})$, and similarly let ADV_t denote the (expected) cost of the advice up to time t: $\text{ADV}_t := \sum_{j=1}^{t} f_j(\mathbf{a}_j) + \mathbb{W}^1(\mathbf{a}_j, \mathbf{a}_{j-1})$. As $z^{(t)}$ denotes the utilization of ST-CLIP, we let $A^{(t)}$ denote the utilization of ADV at time t (i.e., the expected fraction of the deadline constraint satisfied by ADV so far). In addition, ST-CLIP also keeps track of a *robust pseudo-utilization* $\rho^{(t)} \in [0, 1]$; this term describes the portion of its decisions thus far that are attributable to the robust pseudo-cost minimization, and we have $\rho^{(t)} \leq z^{(t)}$ for all $t \in [T]$. This quantity is updated according to the $\bar{\mathbf{k}}_t$ that solves *an unconstrained minimization* in (8), ensuring that when ADV has incurred a "bad" service cost that would otherwise not be considered by the robust algorithm, the pseudo-cost $\psi^{(\varepsilon)}$ maintains some headroom to accept better service costs that might arrive in the future.

Consistency constraint intuition.   Within the constraint (7), ST-CLIP encodes several "worst-case" scenarios that threaten the desired consistency bound. The first three terms on the left-hand side and the ADV_t term on the right-hand side consider the actual cost of ST-CLIP and ADV so far, along with the current decision under consideration, where the expected switching cost is captured by the optimal transport plan with respect to the previous decision.

The $\mathbb{W}^1(\mathbf{p}, \mathbf{a}_t)$ term on the left-hand side charges ST-CLIP in advance for the expected movement cost between it and the advice – the reasoning for this term is to hedge against the case where the constraint becomes binding in future steps, thus requiring ST-CLIP to *move* and follow ADV. If the constraint did not charge for this potential movement cost in advance, a binding constraint in future time steps might result in either an infeasible problem or a violation of $(1 + \varepsilon)$-consistency. The $\tau c(\mathbf{a}_t)$ term on both sides charges both ADV and ST-CLIP in advance for the temporal switching cost they must incur before the deadline – ST-CLIP is charged according to $\mathbf{a}_t$ (as opposed to $\mathbf{p}_t$) to continue hedging against the case where it must move to follow the advice in future time steps, finally paying $\tau c(\mathbf{a}_t)$ to switch OFF at the deadline.

On the right-hand side, the $(1 - A^{(t)})L$ term assumes that ADV can satisfy the remaining deadline constraint at the best marginal service cost L. In contrast, the final terms on the left-hand side $(1 - z^{(t-1)} - c(\mathbf{p}))L + \max((A^{(t)} - z^{(t-1)} - c(\mathbf{p})), 0)(U - L)$ balance between two scenarios – namely, they assume that ST-CLIP can satisfy a fraction of the remaining constraint (up to $(1 - A^{(t)})$) at the best cost by following ADV, but any excess beyond this (given by $(A^{(t)} - z^{(t)})$), must be fulfilled at the worst service cost U, possibly during a mandatory allocation.

At a high level, ST-CLIP's constraint on $\Delta_\mathcal{S}$ combined with the pseudo-cost minimization on $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ generates decisions that are *maximally robust* while preserving consistency.

4.2 Main results

In Theorem 4.3, we give upper bounds on the robustness and consistency of ST-CLIP.

Algorithm 2 ST-CLIP (spatiotemporal consistency-limited pseudo-cost minimization) for SOAD

---

input: Consistency parameter ε, constraint function $c(\cdot)$, pseudo-cost $\psi^{(\varepsilon)}(\cdot)$, starting OFF state $s \in \mathcal{S}$.

initialize: $z^{(0)} = 0$; $\rho^{(0)} = 0$; $A^{(0)} = 0$; $\text{SC}_0 = 0$; $\text{ADV}_0 = 0$; $\mathbf{k}_0 = \Phi\delta_s$; $\mathbf{p}_0 = \mathbf{a}_0 = \delta_s$.

while cost function $f_t(\cdot)$ is revealed, untrusted advice $\mathbf{a}_t$ is revealed, and $z^{(t-1)} < 1$ **do**

Update advice cost $\text{ADV}_t \leftarrow \text{ADV}_{t-1} + f_t(\mathbf{a}_t) + \mathbb{W}^1(\mathbf{a}_t, \mathbf{a}_{t-1})$ and advice utilization $A^{(t)} \leftarrow A^{(t-1)} + c(\mathbf{a}_t)$.

Solve **constrained** pseudo-cost minimization problem:

$$\mathbf{k}_t = \underset{\mathbf{k} \in K : \overline{c}(\mathbf{k}) \leq 1 - z^{(t-1)}}{\arg\min} \overline{f}_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{\rho^{(t-1)}}^{\rho^{(t-1)} + \overline{c}(\mathbf{k})} \psi^{(\varepsilon)}(u) \, du, \tag{6}$$

such that $\mathbf{p} \leftarrow \Phi^{-1}\mathbf{k}$ and,

$$\text{SC}_{t-1} + f_t(\mathbf{p}) + \mathbb{W}^1(\mathbf{p}, \mathbf{p}_{t-1}) + \mathbb{W}^1(\mathbf{p}, \mathbf{a}_t) + \tau c(\mathbf{a}_t) + (1 - z^{(t-1)} - c(\mathbf{p}))L + \max\{A^{(t)} - z^{(t-1)} - c(\mathbf{p}), 0\}(U - L) \tag{7}$$

$$\leq (1 + \varepsilon)[\text{ADV}_t + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L].$$

Update running cost $\text{SC}_t \leftarrow \text{SC}_{t-1} + f_t(\mathbf{p}_t) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1})$ and utilization $z^{(t)} \leftarrow z^{(t-1)} + c(\mathbf{p}_t)$.

Solve **unconstrained** pseudo-cost minimization problem:

$$\tilde{\mathbf{k}}_t = \underset{\mathbf{k} \in K : \overline{c}(\mathbf{k}) \leq 1 - z^{(t-1)}}{\arg\min} \overline{f}_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{\rho^{(t-1)}}^{\rho^{(t-1)} + \overline{c}(\mathbf{k})} \psi^{(\varepsilon)}(u) \, du \tag{8}$$

Update *robust pseudo-utilization* $\rho^{(t)} \leftarrow \rho^{(t-1)} + \min\{\overline{c}(\tilde{\mathbf{k}}_t), c(\mathbf{p}_t)\}$.

THEOREM 4.3. *For any $\varepsilon \in (0, \eta - 1]$, ST-CLIP is $(1 + \varepsilon)$-consistent and $O(\log n)\gamma^{(\varepsilon)}$-robust for* SOAD, *where $\gamma^{(\varepsilon)}$ is the solution to* (5).

Furthermore, we give a lower bound on the best achievable robustness ratio for any $(1+\varepsilon)$-consistent algorithm, using a construction of a challenging metric space and service cost sequence – since robustness and consistency are defined over all valid inputs (i.e., based on competitive ratio), this result characterizes the *optimal robustness-consistency trade-off*, and implies that ST-CLIP matches the optimal up to log factors that are due to the metric embedding.

THEOREM 4.4. *Given untrusted advice* ADV *and $\varepsilon \in (0, \eta - 1]$, any $(1 + \varepsilon)$-consistent learning-augmented algorithm for* SOAD *is at least $\gamma^{(\varepsilon)}$-robust, where $\gamma^{(\varepsilon)}$ is defined in* (5).

Learning-augmentation and robustness-consistency trade-offs have been considered in both MTS and OWT – we briefly review how Theorem 4.4 compares. For MTS, Christianson et al. [22] show that for $\varepsilon \in (0, 1]$, any $(1 + \varepsilon)$-consistent algorithm must be $2^{\Omega(1/\varepsilon)}$-robust. While optimal trade-offs for the minimization variant of OWT have not been studied, Sun et al. [69] show that any γ-robust algorithm must be $\theta/[\theta/\gamma + (\theta - 1)(1 - 1/\gamma \ln(\theta - 1/\gamma - 1))]$-consistent in the maximization case, where $\theta = U/L$ is the price bound ratio. While these bounds are not directly comparable, it is notable that the extra structure of SOAD allows it to avoid the unbounded exponential robustness of MTS.

4.3 Proof overviews

We now give proof sketches of Theorems 4.3 and 4.4, relegating the full proofs to Appendix E.

Proof Sketch of Theorem 4.3. We separately consider consistency and robustness in turn.

LEMMA 4.5. ST-CLIP *is $(1 + \varepsilon)$-consistent when the advice is correct, i.e.,* ADV$(I) = $ OPT(I).

For consistency, recall that the constraint enforces that the expected cost of ST-CLIP *thus far* at time j (i.e., before mandatory allocation) satisfies (7). Since this constraint holds for all steps before the mandatory allocation, we must resolve the cost *during the mandatory allocation*. We characterize two worst-case scenarios based on whether ST-CLIP has completed *less* (**Case 1**, see (37)) or *more* (**Case 2**, see (39)) of the deadline constraint compared to ADV. In either of these cases, (37) and (39) show that replacing the "hedging terms" that follow SC_{j-1} and ADV_{j-1} in the constraint with worst-case service and movement costs yields a consistency ratio that is $\leq (1 + \varepsilon)$.

LEMMA 4.6. *ST-CLIP is $O(\log n)\gamma^{(\varepsilon)}$-robust, where $\gamma^{(\varepsilon)}$ is defined in* (5).

For robustness, we define two cases that characterize "bad" advice, namely "inactive" advice that forces mandatory allocation (**Case 1**, see (40)), and "overactive" advice that incurs sub-optimal cost (**Case 2**, see (43)). For each of these, we derive bounds on the portion of ST-CLIP's expected solution that is *allowed* to come from the pseudo-cost minimization without violating consistency.

In Case 1, ST-CLIP assumes that ADV can satisfy the constraint at the best possible service cost L, so we derive an *upper bound* describing the maximum utilization achievable via the pseudo-cost minimization before the mandatory allocation (see Prop. E.1). In Case 2, ST-CLIP must follow ADV to avoid violating consistency, even if ADV incurs sub-optimal cost – we derive a *lower bound* on the amount of utilization that ST-CLIP must "spend" while continually satisfying the $(1+\varepsilon)$-consistency constraint (see Prop. E.2). These characterizations enable pseudo-cost proof techniques (e.g., as in Thm. 3.2) that show $O(\log n)\gamma^{(\varepsilon)}$-robustness in each case.   □

Proof Sketch of Theorem 4.4. In Definition E.3, we define a slight variant of the special metric and y-adversary construction from Thm. 3.3, denoted by $\mathcal{A}'_y$. Informally, $\mathcal{A}'_y$ presents "good" cost functions at distant points, before eventually presenting just the best service cost functions (i.e., y) at the starting point. We consider two types of advice that each capture consistency and robustness, respectively. In this setting, bad advice completes none of the deadline constraint before the mandatory allocation, while good advice makes the exact decisions that recover OPT($\mathcal{A}'_y$).

Using the proof of Theorem 3.3, we characterize the cost of a learning-augmented algorithm ALG according to two arbitrary *constraint satisfaction functions* $s(y), t(y) : [L, U] \to [0, 1]$ (see (50)). Conditioned on the advice that ALG receives, any α-consistent and γ-robust ALG must satisfy two conditions, where the robustness condition follows from the proof of Theorem 3.3 (see (48)), and the consistency condition is given by $\gamma \int_{U/\gamma}^{L} \ln\left(\frac{U-u-D+2\tau}{U-U/\gamma-D-2\tau}\right) du + [2D + 2\tau] \left[\gamma \ln\left(\frac{U-L-D+2\tau}{U-U/\gamma-D-2\tau}\right)\right] \leq \alpha L - L$ (see (52)). Substituting $\alpha \coloneqq (1 + \varepsilon)$ and binding the inequality above yields the result.   □

5 Generalization to Time-Varying Metrics

Before moving to our case study, we present a generalization of the results in Sections 3 and 4 to settings with *time-varying metrics*. This is motivated by the applications of SOAD (see Section 2.2) since in practice, the distance between points in the metric (e.g., network delays, transit costs) may not be constant. The extension to time-varying metrics is straightforward, and we present corollaries for both PCM and ST-CLIP after formalizing the extension of SOAD that we consider.

5.1 SOAD **with time-varying distances (**SOAD-T**)**

In SOAD with time-varying distances, we let $d_t(\cdot, \cdot) : t \in [T]$ denote a *time-varying distance function* between points in X, and we assume that an online algorithm ALG is always able to observe the current distance d_t at time t. Additionally, we redefine D to be an upper bound on the normalized spatial distance between any two points in X over the entire time horizon T, namely $D = \sup_{t \in [T]} \left(\max_{u,v \in X : u \neq v} \frac{d_t(u, v)}{\min\{c^{(u)}, c^{(v)}\}} \right)$. Although distances between the locations of X are time varying, SOAD-T assumes that the temporal switching cost between ON and OFF states at a single point $u \in X$ is constant (i.e., $\| \cdot \|_{\ell_1(\boldsymbol{\beta})}$ is *not* time-varying) for simplicity of presentation.

5.2 **Main results**

In the following results, we show that our robust algorithm PCM (see Algorithm 1), and our learning-augmented algorithm ST-CLIP (see Algorithm 2) are both sufficiently flexible to provide guarantees in SOAD-T with minimal changes. Note that the lower bounds in the time-invariant setting still apply to the time-varying setting (e.g., by setting d_t constant for all $t \in [T]$).

First, as a corollary to Theorem 3.2, we show that PCM retains its $O(\log n)\eta$ competitive bound in the setting of SOAD-T. We state the result here and give the full proof in Appendix F.1.

COROLLARY 5.1. PCM *is* $O(\log n)\eta$*-competitive for* SOAD-T, *where* η *is given by* (4).

Furthermore, as a corollary to Theorem 4.3, we show that ST-CLIP's consistency-robustness bound also holds for the time-varying setting of SOAD-T when just one term is swapped within the consistency constraint. We state the result here and give the full proof in Appendix F.2.

COROLLARY 5.2. *With a minor change to the consistency constraint,* ST-CLIP *is* $(1 + \varepsilon)$*-consistent and* $O(\log n)\gamma^{(\varepsilon)}$*-robust for* SOAD-T, *where* $\gamma^{(\varepsilon)}$ *is the solution to* (5).

6 Case Study: Carbon-aware Workload Management in Data Centers

We end the paper with a case study application of SOAD and our algorithms for the motivating application of carbon-aware workload management on a simulated global network of data centers.

6.1 Experimental setup

We simulate a carbon-aware scheduler that schedules a delay-tolerant batch job on a network of data centers. We simulate a global network of data centers based on measurements between Amazon Web Services (AWS) regions. We construct SOAD instances as follows: we generate a job with length J (in hours), an arrival time (rounded to the nearest hour), and a "data size" G, where G gives the amount of data (in GB) to be transferred while migrating the job. The task is to finish the job before the deadline T while minimizing total CO_2 emissions, which are a function of the scheduling decisions and the carbon intensity at each time step and region.

AWS measurement data. We pick 14 AWS regions [74] based on available carbon data (see Table 2 in the Appendix). Among these regions, we collect 72,900 pairwise measurements of latency and throughput, compute the mean and variance, and sample a latency matrix. To model migration overhead, we scale the data transferred (and corresponding latency) to match G. These values are scaled by carbon data to define a distance metric on the regions in terms of CO_2 *overhead*. To model network heterogeneity, we set a parameter $\kappa \in [0, 1]$ to adjust the simulated energy of the network. κ is a ratio – if $\kappa = 0.5$, a minute of data transfer from machine(s) in one region to machine(s) in another uses half as much energy as executing at the full allocation (i.e., $x_t^{\mathsf{ON}^{(\cdot)}} = 1$) for one minute.

Carbon data traces. We obtain hourly carbon intensity data for each region, expressed as grams of CO_2 equivalent per kilowatt-hour. In the main body, we consider *average carbon intensity* [57], which gives the average emissions of all electricity generated on a grid at a certain time; this data spans 2020-2022 and includes all regions. In Appendix B.1, we also consider *marginal carbon intensity* [72]; this signal is available for 9 regions in 2022, and also includes proprietary forecasts.

We use the latency of moving data between regions to calculate a CO_2 *overhead* for the metric (X, d) (latency × energy × carbon intensity). In most cases, we approximate the network's carbon intensity by the average across regions. When specified, we introduce variation by resampling the carbon intensity of up to $\Upsilon \in [0, n^2]$ links each time step. We henceforth call Υ a *volatility factor*; resampling assigns a new random carbon intensity (within $[L, U]$) to a link between two regions.

Cloud job traces. We use Google cluster traces [64] that provide a real distribution of job lengths. We normalize this distribution such that the maximum length is 12 hours – each job's length J is drawn from the distribution and rounded up to the next integer, so J falls in the range $\{1, ..., 12\}$.

Forecasts. We generate forecasts of the carbon intensity for each location and time. These forecasts are used to solve for a predicted optimal solution that assumes they are correct, which becomes black-box ADV for ST-CLIP. For the average carbon intensity signal, we generate synthetic forecasts
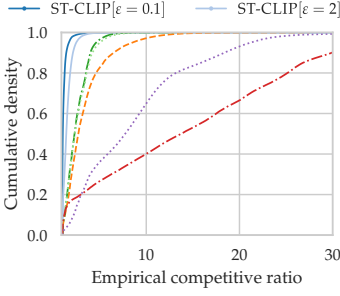
Fig. 2. CDFs of competitive ratios for each algorithm, across all average carbon intensity experiments.
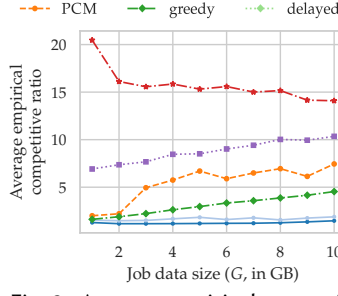
Fig. 3. Average empirical competitive ratios for varying *job data size* G, with $T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.
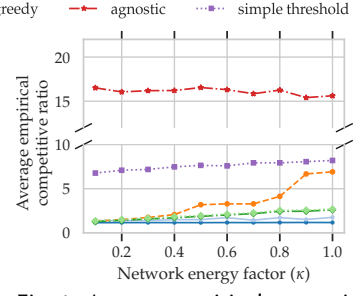
Fig. 4. Average empirical competitive ratios for varying *energy factor* κ, with $G = 4, T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.

by combining true data with random noise.[2] Letting $\text{Carbon}_t^{(u)}$ denote the carbon intensity at data center u and time t, our synthetic forecast is given by $\text{Pred}_t^{(u)} = 0.6 \cdot \text{Carbon}_t^{(u)} + 0.4 \cdot \text{Unif}(L, U)$. To test ST-CLIP's robustness, Experiment V directly manipulates Adv. We set an *adversarial factor* $\xi \in [0, 1]$, where $\xi = 0$ implies Adv is correct. We use a solver on true data to obtain two solutions, where one is given a *flipped objective* (i.e., it maximizes carbon emissions). Letting $\{\mathbf{x}_t^\star\}_{t \in [T]}$ denote the decisions of Opt and $\{\check{\mathbf{x}}_t\}_{t \in [T]}$ denote the decisions of the maximization solution, we have $\text{Adv} \coloneqq \{(1 - \xi)\mathbf{x}_t^\star + \xi\check{\mathbf{x}}_t\}_{t \in [T]}$. We note that although this is unrealistic in practice, manipulating Adv directly allows us to to quantify the sensitivity of ST-CLIP against all sources of error.

Setup details. We simulate 1,500 jobs for each configuration. Each job's arrival region and arrival time is uniformly random across all active regions and times. Each job's deadline T and data size G are either fixed or drawn from a distribution, and this is specified. To set the parameters L and U, we examine the preceding month of carbon intensities (in all regions) leading up to the arrival time and set L and U according to the minimum and maximum, respectively. We set the following defaults (i.e., unless otherwise specified): The metric covers all 14 regions. Each job's length is drawn from the Google traces as above. The temporal switching coefficient τ is set to 1, the network energy factor κ is set to 0.5, and the volatility factor Υ is set to 0 (i.e., the network is stable).

Benchmark algorithms. We compute the offline optimal solution for each instance using CVXPY [25]. We compare ST-CLIP and PCM against four baselines adapted from literature. The first is a **carbon-agnostic** approach that runs the job whenever it is submitted without migration, simulating the behavior of a non-carbon-aware scheduler. We also consider two *greedy baselines* that use simple decision rules. The first of these is a **greedy** policy that examines the current carbon intensity across all regions at the arrival time, migrates to the "greenest" region (i.e., with lowest carbon intensity), and runs the full job. This captures an observation [68] that one migration to a consistently low-carbon region yields most of the benefits of spatiotemporal shifting. We also consider a policy that we term **delayed greedy**, which examines the *full forecast* across all regions, migrating to start the job at the "best region and time" (i.e., slot with lowest predicted carbon anywhere). If there is not enough time to finish the job after the identified slot, it is scheduled to start as close to it as possible. The final baseline is a **simple threshold**-based approach from temporal shifting literature [11, 42]; it sets a threshold $\sqrt{UL}$, based on prior work in online search [27]. At each time step, it runs the job in the best region whose carbon intensity is $\leq \sqrt{UL}$, without considering migration overheads. If no regions are $\leq \sqrt{UL}$ at a particular time, the job is checkpointed in place, and a *mandatory allocation* happens when approaching the deadline if the job is not finished.

---

[2]We use an open-source ML model that provides carbon intensity forecasts for U.S. regions [54] to tune the magnitude of random noise such that Adv's empirical competitive ratio is slightly worse than an Adv that uses the ML forecasts.
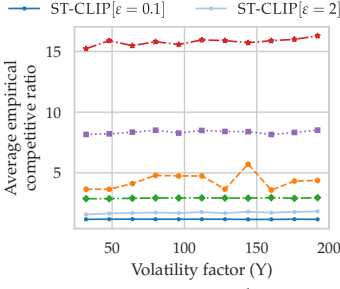
Fig. 5. Average empirical competitive ratios for varying *volatility factor* Υ. $G = 4, T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.
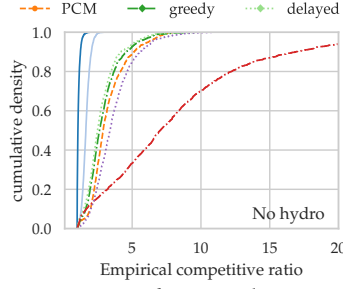
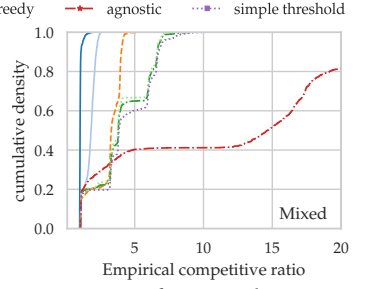Fig. 6. CDFs of empirical competitive ratios for each tested algorithm on the "no hydro" region subset.

Fig. 7. CDFs of empirical competitive ratios for each tested algorithm on the "mixed" region subset.

6.2 Experimental results

We highlight several experiments here, referring to Appendix B for the extended set. A summary is given in Fig. 2, where we plot a cumulative distribution function (CDF) of the empirical competitive ratio for all tested algorithms in Expts. I-IV and VI-VIII. Given imperfect advice, ST-CLIP[$\varepsilon = 2$] significantly outperforms the baselines, improving on greedy, delayed greedy, simple threshold, and carbon-agnostic by averages of 32.1%, 33.5%, 79.4%, and 88.7%, respectively. In Expts. I-III, each job's deadline is a random integer between 12 and 48 (denoted by $T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$). In these experiments, both greedy policies outperform our robust algorithm, PCM. This result aligns with prior findings [68]; since these experiments consider all 14 regions, there are consistent low-carbon grids in the mix that give an advantage to the greedy policies. In Expt. IV, we examine this further, showing that the performance of greedy policies can degrade in realistic situations.

Experiment I: *Effect of job data size G.* In Fig. 3, we plot the average empirical competitive ratio for job data sizes $G \in \{1, \ldots, 10\}$. Recall that parameter D depends on the diameter of the metric space (i.e., the worst migration overhead between regions); as G increases, this maximum overhead grows. As predicted by the theoretical bounds, PCM's performance degrades as G grows; we observe the same effect for the greedy policies and simple threshold. Since it can leverage advice, ST-CLIP maintains consistent performance for many settings of G.

Experiment II: *Effect of network energy scale κ.* Fig. 4 plots the average empirical competitive ratio for $\kappa \in [0.1, 1]$, fixing $G = 4$. As in Expt. I, κ affects the parameter D – thus, the performance of PCM degrades slightly as κ grows. When κ is small, greedy policies perform nearly as well as ST-CLIP, though they degrade as κ increases; ST-CLIP's usage of advice yields consistent performance.

Experiment III: *Effect of volatility factor Υ.* In Fig. 5, we plot the average empirical competitive ratio for $\Upsilon \in [28, 196]$, fixing $G = 4$. Aligning with the theoretical results (Corollary 5.1 & 5.2), we find that PCM and ST-CLIP's performance is robust to this volatility. Both of the greedy policies do not consider the migration overhead and only migrate once, so their performance is consistent.

Experiment IV: *Effect of electric grids and data center availability.* Greedy policies do well in Expts. I-III, where some regions have consistently low-carbon grids.[3] In practice, a greedy policy may face obstacles if it is unable to migrate to low-carbon regions. For instance, such regions might reach capacity, *removing them* as migration options. Jobs may also be *restricted* from leaving a region due to regulations [28]. Further, consistently low-carbon grids often leverage hydroelectric or nuclear sources that are difficult to build at scale compared to cheaper renewables [62]. This is important because it suggests future grids will moreso resemble those marked by renewable intermittency.

---

[3]Fig. 9 in the Appendix plots a sample of carbon intensity data for all 14 regions to motivate this visually.

In Fig. 6 and 7, we present results where the metric is a *subset* of the 14 regions, giving results on more subsets in Appendix B. By considering these subsets, we approximate issues discussed above (e.g., data center congestion, grid characteristics). Fig. 6 considers a "no hydroelectric" subset that omits Sweden and Quebec. Under this subset, PCM closes the gap with the greedy policies, with an average competitive ratio that is within 4.24% of both. Fig. 7 considers a smaller subset of 5 regions: South Korea, Virginia, Sydney, Quebec, and France. On this mixed set of grids, PCM outperforms greedy and delayed greedy by 30.91% and 28.79%, respectively. The results above highlight that situations do arise where greedy policies perform worse than both ST-CLIP and PCM. However, such situations are not a majority – out of the 14 random subsets that we tested, PCM outperformed the greedy policy in *four* subsets, including the "mixed" (Fig. 7) and "mixed 2" (Fig. 15) subsets. PCM is relatively conservative in its decisions,

Fig. 8. Average empirical competitive ratio for varying *adversarial factor* ξ, with $G = 4, T = 12$. $\xi \to 0$ implies that ADV $\to$ OPT, and as ξ grows, ADV degrades.

being optimized for worst-case (adversarial) inputs – this advantages greedy policies on the majority of instances that do not benefit from the "worst-case hedging" behavior that PCM exhibits.

Experiment V: *Effect of bad black-box advice ξ.* Fig. 8 plots the effect of bad black-box advice on ST-CLIP's performance. We test values of $\xi \in [0, 0.6]$, generating ADV according to the technique discussed in Section 6.1. ST-CLIP is initialized with $\varepsilon \in [0.1, 2, 5, 10]$, where $\varepsilon \to 0$ implies that it follows ADV more closely. We also plot the empirical competitive ratio of PCM and ADV as opposing baselines. We find that ST-CLIP nearly matches ADV when it is correct, while degrading more gracefully as ADV's cost increases. This result shows that ST-CLIP is empirically robust to even adversarial black-box advice.

7 Conclusion

Motivated by sustainability applications, we introduce and study spatiotemporal online allocation with deadlines (SOAD), the first online problem that combines general metrics with deadline constraints, bridging the gap between existing literature on metrical task systems and online search. Our main results present PCM as a competitive algorithm for SOAD and ST-CLIP, a learning-augmented algorithm that achieves the optimal robustness-consistency trade-off. We evaluate our proposed algorithms in a case study of carbon-aware workload management in data centers. A number of questions remain for future work, including natural extensions motivated by applications. For computing applications, SOAD may be extended to model resource contention and/or delayed access to resources, particularly after moving the allocation to a new point. Similarly, an extension to model *multiple* workloads with different deadlines would be natural (e.g., scheduling multiple batch jobs, dynamic job arrivals/departures). Our theoretical results contend with substantial generality (i.e., in the metric); it would be interesting to explore whether improved results can be obtained under a more structured setting.

Acknowledgments

Disclaimers

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

References

[1] Jacob Abernethy, Peter L. Bartlett, Niv Buchbinder, and Isabelle Stanton. 2010. *A Regularization Approach to Metrical Task Systems*. Springer Berlin Heidelberg, 270–284. https://doi.org/10.1007/978-3-642-16108-7_23

[2] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Kiwan Maeng, Udit Gupta, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. 2023. Carbon Explorer: A Holistic Framework for Designing Carbon Aware Datacenters. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2* (Vancouver, BC, Canada) *(ASPLOS 2023)*. 118–132. https://doi.org/10.1145/3575693.3575754

[3] Antonios Antoniadis, Christian Coester, Marek Elias, Adam Polak, and Bertrand Simon. 2020. Online Metric Algorithms with Untrusted Predictions. In *Proceedings of the 37th International Conference on Machine Learning*. 345–355.

[4] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. 2018. Bandits with Knapsacks. *J. ACM* 65, 3, Article 13 (Mar 2018), 55 pages. https://doi.org/10.1145/3164539

[5] Nikhil Bansal, Niv Buchbinder, and Joseph Naor. 2010. Metrical task systems and the k-server problem on HSTs. In *International Colloquium on Automata, Languages, and Programming*. 287–298.

[6] Nikhil Bansal and Christian Coester. 2022. Online Metric Allocation and Time-Varying Regularization. In *30th Annual European Symposium on Algorithms (ESA 2022) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 244)*, Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman (Eds.). 13:1–13:13. https://doi.org/10.4230/LIPIcs.ESA.2022.13

[7] Yair Bartal, Avrim Blum, Carl Burch, and Andrew Tomkins. 1997. A polylog(n)-competitive algorithm for metrical task systems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing* (El Paso, TX, USA). 711–719.

[8] Noman Bashir, Tian Guo, Mohammad Hajiesmaili, David Irwin, Prashant Shenoy, Ramesh Sitaraman, Abel Souza, and Adam Wierman. 2021. Enabling Sustainable Clouds: The Case for Virtualizing the Energy System. In *Proceedings of the ACM Symposium on Cloud Computing* (Seattle, WA, USA) *(SoCC '21)*. 350–358. https://doi.org/10.1145/3472883.3487009

[9] Avrim Blum and Carl Burch. 1997. On-line learning and the metrical task system problem. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory* (Nashville, Tennessee, USA) *(COLT '97)*. 45–53. https://doi.org/10.1145/267460.267475

[10] Allan Borodin, Nathan Linial, and Michael E. Saks. 1992. An Optimal On-Line Algorithm for Metrical Task System. *J. ACM* 39, 4 (Oct 1992), 745–763. https://doi.org/10.1145/146585.146588

[11] Roozbeh Bostandoost, Walid A. Hanafy, Adam Lechowicz, Noman Bashir, Prashant Shenoy, and Mohammad Hajiesmaili. 2024. Data-driven Algorithm Selection for Carbon-Aware Scheduling. In *Proceedings of the 3rd Workshop on Sustainable Computer Systems* (Santa Cruz, CA, USA) *(HotCarbon '24)*.

[12] Roozbeh Bostandoost, Adam Lechowicz, Walid A. Hanafy, Noman Bashir, Prashant Shenoy, and Mohammad Hajiesmaili. 2024. LACS: Learning-Augmented Algorithms for Carbon-Aware Resource Scaling with Uncertain Demand. In *Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems* (Singapore, Singapore) *(e-Energy '24)*. 27–45. https://doi.org/10.1145/3632775.3661942

[13] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[14] Patrick R. Brown and Francis M. O'Sullivan. 2020. Spatial and temporal variation in the value of solar power across United States electricity markets. *Renewable and Sustainable Energy Reviews* 121 (2020), 109594. https://doi.org/10.1016/j.rser.2019.109594

[15] Sébastien Bubeck, Christian Coester, and Yuval Rabani. 2023. The Randomized k-Server Conjecture Is False!. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing* (Orlando, FL, USA) *(STOC 2023)*. 581–594.

https://doi.org/10.1145/3564246.3585132

[16] Sébastien Bubeck, Michael B. Cohen, James R. Lee, and Yin Tat Lee. 2021. Metrical Task Systems on Trees via Mirror Descent and Unfair Gluing. *SIAM J. Comput.* 50, 3 (Jan. 2021), 909–923. https://doi.org/10.1137/19M1237879

[17] Sébastien Bubeck, Bo'az Klartag, Yin Tat Lee, Yuanzhi Li, and Mark Sellke. 2020. Chasing Nested Convex Bodies Nearly Optimally. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '20)*. 1496–1508. https://doi.org/10.1137/1.9781611975994.91

[18] NiangJun Chen, Gautam Goel, and Adam Wierman. 2018. Smoothed Online Convex Optimization in High Dimensions via Online Balanced Descent. In *Proceedings of the 31st Conference On Learning Theory*. 1574–1594.

[19] Andrew A. Chien. 2021. Driving the cloud to true zero carbon. *Commun. ACM* 64, 2 (Jan 2021), 5. https://doi.org/10.1145/3445037

[20] Andrew A Chien, Liuzixuan Lin, Hai Nguyen, Varsha Rao, Tristan Sharma, and Rajini Wijayawardana. 2023. Reducing the Carbon Impact of Generative AI Inference (today and in 2035). In *Proceedings of the 2nd Workshop on Sustainable Computer Systems* (Boston, MA, USA) *(HotCarbon '23)*. Article 11, 7 pages. https://doi.org/10.1145/3604930.3605705

[21] Nicolas Christianson, Tinashe Handina, and Adam Wierman. 2022. Chasing Convex Bodies and Functions with Black-Box Advice. In *Proceedings of the 35th Conference on Learning Theory*, Vol. 178. 867–908.

[22] Nicolas Christianson, Junxuan Shen, and Adam Wierman. 2023. Optimal robustness-consistency tradeoffs for learning-augmented metrical task systems. In *International Conference on Artificial Intelligence and Statistics*. 9377–9399.

[23] Christian Coester and James R Lee. 2019. Pure entropic regularization for metrical task systems. In *Conference on Learning Theory*. PMLR, 835–848.

[24] Robert M Corless, Gaston H Gonnet, David EG Hare, David J Jeffrey, and Donald E Knuth. 1996. On the Lambert W function. *Advances in Computational Mathematics* 5 (1996), 329–359.

[25] Steven Diamond and Stephen Boyd. 2016. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research* 17, 83 (2016), 1–5.

[26] Hui Dou, Yong Qi, Wei Wei, and Houbing Song. 2017. Carbon-Aware Electricity Cost Minimization for Sustainable Data Centers. *IEEE Transactions on Sustainable Computing* 2, 2 (2017), 211–223. https://doi.org/10.1109/TSUSC.2017.2711925

[27] Ran El-Yaniv, Amos Fiat, Richard M. Karp, and Gordon Turpin. 2001. Optimal Search and One-Way Trading Online Algorithms. *Algorithmica* 30, 1 (May 2001), 101–139. https://doi.org/10.1007/s00453-001-0003-0

[28] European Parliament and Council of the European Union. 2016. *Regulation (EU) 2016/679 of the European Parliament and of the Council.* https://data.europa.eu/eli/reg/2016/679/oj

[29] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. 2007. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. System Sci.* 69, 3 (October 2007), 485–497. https://www.microsoft.com/en-us/research/publication/a-tight-bound-on-approximating-arbitrary-metrics-by-tree-metrics/

[30] Lingwen Gan, Adam Wierman, Ufuk Topcu, Niangjun Chen, and Steven H. Low. 2014. Real-time deferrable load control: handling the uncertainties of renewable generation. *SIGMETRICS Perform. Eval. Rev.* 41, 3 (Jan 2014), 77–79. https://doi.org/10.1145/2567529.2567553

[31] Vani Gupta, Prashant Shenoy, and Ramesh K Sitaraman. 2019. Combining renewable solar and open air cooling for greening internet-scale distributed networks. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems*. 303–314.

[32] Walid A. Hanafy, Roozbeh Bostandoost, Noman Bashir, David Irwin, Mohammad Hajiesmaili, and Prashant Shenoy. 2023. The War of the Efficiencies: Understanding the Tension between Carbon and Energy Optimization. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems* (Boston, MA, USA) *(HotCarbon '23)*. Article 19, 7 pages. https://doi.org/10.1145/3604930.3605709

[33] Walid A. Hanafy, Roozbeh Bostandoost, Noman Bashir, David Irwin, Mohammad Hajiesmaili, and Prashant Shenoy. 2023. The War of the Efficiencies: Understanding the Tension between Carbon and Energy Optimization. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems*. https://doi.org/10.1145/3604930.3605709

[34] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, 7825 (Sept. 2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[35] Abdolhossein Hoorfar and Mehdi Hassani. 2008. Inequalities on the Lambert W function and hyperpower function. *Journal of Inequalities in Pure and Applied Mathematics* 9, 51 (Jan. 2008). Issue 2.

[36] Rodolphe Jenatton, Jim Huang, and Cedric Archambeau. 2016. Adaptive Algorithms for Online Convex Optimization with Long-term Constraints. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). 402–411. https://proceedings.mlr.press/v48/jenatton16.html

[37] Billy Jin and Will Ma. 2022. Online Bipartite Matching with Advice: Tight Robustness-Consistency Tradeoffs for the Two-Stage Model. *Advances in Neural Information Processing Systems* 35 (2022), 14555–14567.

[38] Elias Koutsoupias. 2009. The *k*-server problem. *Computer Science Review* 3, 2 (May 2009), 105–118. https://doi.org/10.1016/j.cosrev.2009.04.002

[39] Ravi Kumar, Manish Purohit, and Zoya Svitkina. 2018. Improving Online Algorithms via ML Predictions. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31.

[40] Adam Lechowicz, Nicolas Christianson, Bo Sun, Noman Bashir, Mohammad Hajiesmaili, Adam Wierman, and Prashant Shenoy. 2024. Chasing Convex Functions with Long-term Constraints. In *Proceedings of the 41st International Conference on Machine Learning (ICML '24)*. arXiv:2402.14012 [cs.DS]

[41] Adam Lechowicz, Nicolas Christianson, Bo Sun, Noman Bashir, Mohammad Hajiesmaili, Adam Wierman, and Prashant Shenoy. 2024. Online Conversion with Switching Costs: Robust and Learning-augmented Algorithms. In *Proceedings of the 2024 SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems* (Venice, Italy) *(SIGMETRICS / Performance '24)*.

[42] Adam Lechowicz, Nicolas Christianson, Jinhang Zuo, Noman Bashir, Mohammad Hajiesmaili, Adam Wierman, and Prashant Shenoy. 2023. 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, Article 53 (Dec 2023), 36 pages. arXiv:2303.17551 [cs.DS]

[43] Russell Lee, Bo Sun, Mohammad Hajiesmaili, and John C. S. Lui. 2024. Online Search with Predictions: Pareto-optimal Algorithm and its Applications in Energy Markets. In *Proceedings of the 15th ACM International Conference on Future Energy Systems* (Singapore, Singapore) *(e-Energy '24)*.

[44] Minghong Lin, Zhenhua Liu, Adam Wierman, and Lachlan L. H. Andrew. 2012. Online algorithms for geographical load balancing. In *2012 International Green Computing Conference (IGCC)*. https://doi.org/10.1109/igcc.2012.6322266

[45] Minghong Lin, Adam Wierman, Lachlan LH Andrew, and Eno Thereska. 2012. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Transactions on Networking* 21, 5 (2012), 1378–1391.

[46] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. 2012. Renewable and cooling aware workload management for sustainable data centers. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*. 175–186.

[47] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H. Low, and Lachlan L.H. Andrew. 2011. Geographical load balancing with renewables. *SIGMETRICS Perform. Eval. Rev.* 39, 3 (Dec 2011), 62–66. https://doi.org/10.1145/2160803.2160862

[48] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H Low, and Lachlan LH Andrew. 2011. Greening geographical load balancing. *ACM SIGMETRICS Performance Evaluation Review* 39, 1 (2011), 193–204.

[49] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, and Niangjun Chen. 2013. Data center demand response: Avoiding the coincident peak via workload shifting and local generation. *Performance Evaluation* 70, 10 (Oct. 2013), 770–791. https://doi.org/10.1016/j.peva.2013.08.014

[50] Julian Lorenz, Konstantinos Panagiotou, and Angelika Steger. 2008. Optimal Algorithms for *k*-Search with Application in Option Pricing. *Algorithmica* 55, 2 (Aug. 2008), 311–328. https://doi.org/10.1007/s00453-008-9217-8

[51] Los Angeles Department of Water and Power. 2024. Residential Electric Rates - Los Angeles. https://www.ladwp.com/account/understanding-your-rates/residential-electric-rates [Accessed 09-02-2024].

[52] Thodoris Lykouris and Sergei Vassilvtiskii. 2018. Competitive Caching with Machine Learned Advice. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). 3296–3305. https://proceedings.mlr.press/v80/lykouris18a.html

[53] Will Ma, David Simchi-Levi, and Jinglong Zhao. 2019. A Competitive Analysis of Online Knapsack Problems with Unit Density. *SSRN Electronic Journal* (2019). https://doi.org/10.2139/ssrn.3423199

[54] Diptyaroop Maji, Prashant Shenoy, and Ramesh K. Sitaraman. 2022. CarbonCast: Multi-Day Forecasting of Grid Carbon Intensity. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation* (Boston, Massachusetts) *(BuildSys '22)*. 198–207. https://doi.org/10.1145/3563357.3564079

[55] Mark Manasse, Lyle McGeoch, and Daniel Sleator. 1988. Competitive Algorithms for On-Line Problems. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing* (Chicago, Illinois, USA) *(STOC '88)*. 322–333. https://doi.org/10.1145/62212.62243

[56] Vahideh Manshadi, Scott Rodilitz, Daniela Saban, and Akshaya Suresh. 2023. Redesigning VolunteerMatch's Ranking Algorithm: Toward More Equitable Access to Volunteers. *SSRN Electronic Journal* (2023). https://doi.org/10.2139/ssrn.4497747

[57] Electricity Maps. 2024. Electricity Map. https://www.electricitymap.org/map.

[58] Dragoslav S. Mitrinovic, Josip E. Pečarić, and A. M. Fink. 1991. *Inequalities Involving Functions and Their Integrals and Derivatives*. Vol. 53. Springer Science & Business Media.

[59] Esther Mohr, Iftikhar Ahmad, and Günter Schmidt. 2014. Online algorithms for conversion problems: A survey. *Surveys in Operations Research and Management Science* 19, 2 (July 2014), 87–104. https://doi.org/10.1016/j.sorms.2014.08.001

[60] Moxion Power. 2024. Technology. https://www.moxionpower.com/technology/ [Accessed 09-02-2024].

[61] Adel Nadjaran Toosi, Chenhao Qu, Marcos Dias de Assunção, and Rajkumar Buyya. 2017. Renewable-aware geographical load balancing of web applications for sustainable data centers. *Journal of Network and Computer Applications* 83 (2017), 155–168. https://doi.org/10.1016/j.jnca.2017.01.036

[62] NREL (National Renewable Energy Laboratory). 2024. *2024 Annual Technology Baseline*. National Renewable Energy Laboratory. https://atb.nrel.gov/

[63] Ana Radovanovic, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, et al. 2022. Carbon-Aware Computing for Datacenters. *IEEE Transactions on Power Systems* (2022).

[64] Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Randy H. Katz, and Michael A. Kozuch. 2012. Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis. In *Proceedings of the Third ACM Symposium on Cloud Computing* (San Jose, California) *(SoCC '12)*. Article 7, 13 pages. https://doi.org/10.1145/2391229.2391236

[65] Junaid Shuja, Abdullah Gani, Shahaboddin Shamshirband, Raja Wasim Ahmad, and Kashif Bilal. 2016. Sustainable Cloud Data Centers: A survey of enabling techniques and technologies. *Renewable and Sustainable Energy Reviews* 62 (Sept. 2016), 195–214. https://doi.org/10.1016/j.rser.2016.04.034

[66] Seán M. Stewart. 2009. On Certain Inequalities Involving the Lambert W function. *Journal of Inequalities in Pure and Applied Mathematics* 10, 96 (Nov. 2009). Issue 4.

[67] Thanathorn Sukprasert, Abel Souza, Noman Bashir, David Irwin, and Prashant Shenoy. 2023. Spatiotemporal Carbon-Aware Scheduling in the Cloud: Limits and Benefits. In *Companion Proceedings of the 14th ACM International Conference on Future Energy Systems* (Orlando, FL, USA) *(e-Energy '23 Companion)*. https://doi.org/10.1145/3599733.3606301

[68] Thanathorn Sukprasert, Abel Souza, Noman Bashir, David Irwin, and Prashant Shenoy. 2024. On the Limitations of Carbon-Aware Temporal and Spatial Workload Shifting in the Cloud. In *Proceedings of the Nineteenth European Conference on Computer Systems* (Athens, Greece) *(EuroSys '24)*. 924–941. https://doi.org/10.1145/3627703.3650079

[69] Bo Sun, Russell Lee, Mohammad Hajiesmaili, Adam Wierman, and Danny Tsang. 2021. Pareto-Optimal Learning-Augmented Algorithms for Online Conversion Problems. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34.

[70] Bo Sun, Ali Zeynali, Tongxin Li, Mohammad Hajiesmaili, Adam Wierman, and Danny H.K. Tsang. 2021. Competitive Algorithms for the Online Multiple Knapsack Problem with Application to Electric Vehicle Charging. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4, 3, Article 51 (June 2021), 32 pages.

[71] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[72] WattTime. 2024. WattTime. https://www.watttime.org.

[73] Waymo LLC. 2024. Waymo - Self-Driving Cars - Autonomous Vehicles - Ride-Hail. https://waymo.com [Accessed 09-02-2024].

[74] Amazon Web Services. 2024. Global Infrastructure Regions. https://aws.amazon.com/about-aws/global-infrastructure/regions_az/.

[75] Alexander Wei and Fred Zhang. 2020. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, Canada) *(NeurIPS '20)*. Article 674, 12 pages.

[76] Adam Wierman, Zhenhua Liu, Iris Liu, and Hamed Mohsenian-Rad. 2014. Opportunities and challenges for data center demand response. In *International Green Computing Conference*. https://doi.org/10.1109/igcc.2014.7039172

[77] Philipp Wiesner, Ilja Behnke, Dominik Scheinert, Kordian Gontarska, and Lauritz Thamsen. 2021. Let's Wait AWhile: How Temporal Workload Shifting Can Reduce Carbon Emissions in the Cloud. In *Proceedings of the 22nd International Middleware Conference*. 260–272.

[78] Lin Yang, Ali Zeynali, Mohammad H. Hajiesmaili, Ramesh K. Sitaraman, and Don Towsley. 2021. Competitive Algorithms for Online Multidimensional Knapsack Problems. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 5, 3, Article 30 (Dec 2021), 30 pages.

[79] Sheng Shui Zhang. 2006. The effect of the charging protocol on the cycle life of a Li-ion battery. *Journal of Power Sources* 161, 2 (2006), 1385–1391. https://doi.org/10.1016/j.jpowsour.2006.06.040

[80] ZiJun Zhang, Zongpeng Li, and Chuan Wu. 2017. Optimal posted prices for online cloud resource allocation. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 1, 1 (2017), 1–26.

[81] Jiajia Zheng, Andrew A. Chien, and Sangwon Suh. 2020. Mitigating Curtailment and Carbon Emissions through Load Migration between Data Centers. *Joule* 4, 10 (Oct. 2020), 2208–2222. https://doi.org/10.1016/j.joule.2020.08.001

[82] Yunhong Zhou, Deeparnab Chakrabarty, and Rajan Lukose. 2008. Budget Constrained Bidding in Keyword Auctions and Online Knapsack Problems. In *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 566–576.

Appendix

Table 1. A summary of key notations.

Notation	Description
$t \in [T]$	Time step index
$\mathcal{X}$	Feasible set for vector allocation decisions
$\mathbf{x}_t \in \mathcal{X}$	Allocation decision at time t
$f_t(\cdot) : \mathcal{X} \to \mathbb{R}$	(*Online input*) Service cost function revealed to the player at time t
$c(\cdot) : \mathcal{X} \to [0,1]$	Constraint function; describes the fraction satisfied by an allocation
$d(u,v) : u, v \in X \to \mathbb{R}$	Spatial distance in the metric (X, d)
$\|\mathbf{x} - \mathbf{x}'\|_{\ell_1(\beta)} : \mathbf{x}, \mathbf{x}' \in \mathcal{X} \to \mathbb{R}$	Switching costs between ON and OFF allocations
$g(\cdot, \cdot) \coloneqq d(\cdot, \cdot) + \|\cdot\|_{\ell_1(\beta)}$	Combined movement & switching cost between points and allocations
$u \in X$	Point u in an n-point metric space (X, d)
$\text{ON}^{(u)}, \text{OFF}^{(u)}$	ON state and OFF state at point u, respectively
$x_t^{\text{ON}^{(u)}}, x_t^{\text{OFF}^{(u)}} \in [0,1]$	Fractional allocations to ON / OFF states at point u at time t
$c^{(u)} \in (0,1]$	Throughput coefficient; describes constraint satisfied by $x^{\text{ON}^{(u)}} = 1$
$f_t^{(u)} \in [c^{(u)}L, c^{(u)}U]$	Service cost coefficient at $\text{ON}^{(u)}$ & time t; proportional to $L > 0$ and U.
$\beta^{(u)} > 0$	Switching coefficient; describes switching cost between $\text{ON}^{(u)} \leftrightarrow \text{OFF}^{(u)}$
$\tau : \beta^{(u)} \leq \tau c^{(u)} \ \forall u \in X$	Upper bound on normalized switching coefficient
$D = \sup_{u,v \in X} \frac{d(u,v)}{\min\{c^{(u)}, c^{(v)}\}}$	Upper bound on normalized spatial distance between any two points
$\mathcal{S}$	Discrete set of all ON and OFF states
$\Delta_{\mathcal{S}}$	Probability measure over $\mathcal{S}$
$\mathbf{p}_t \in \Delta_{\mathcal{S}}$	Probability distribution (& corresponding random allocation) at time t
$\mathbb{W}_1(\mathbf{p}, \mathbf{p}') : \mathbf{p}, \mathbf{p}' \in \Delta_{\mathcal{S}} \to \mathbb{R}$	Optimal transport distance between distributions (in terms of $g(\cdot, \cdot)$)
$\delta_s \in \Delta_{\mathcal{S}}$	Dirac measure supported at $\text{OFF}^{(s)}$
$\mathcal{T} = (V, E)$	Hierarchically separated tree (HST) constructed by Def. 2.3
$K \subset \mathbb{R}^{|V|}$	Vector space corresponding to $\mathcal{T}$ (see Def. 2.4)
$\mathbf{k}_t \in K$	Vector decision (& corresponding prob. distribution) at time t
$\Phi \in \mathbb{R}^{|V| \times 2n}$	Linear map such that $\Phi\mathbf{p} \in K$ and $\Phi^{-1}\mathbf{k} \in \Delta_{\mathcal{S}}$
$\|\cdot\|_{\ell_1(\mathbf{w})} : K \to \mathbb{R}$	Weighted ℓ_1 norm that recovers optimal transport distances in $\mathcal{T}$
$\overline{f}_t(\mathbf{k}) = f_t(\Phi^{-1}\mathbf{k})$ $\overline{c}(\mathbf{k}) = c(\Phi^{-1}\mathbf{k})$	Notation shorthand for functions defined on vector space K
$z^{(t)} \in [0,1]$	Utilization; fraction of constraint satisfied in expectation up to time t
$\text{Adv}(\mathcal{I}) \coloneqq \{\mathbf{a}_t \in \Delta_{\mathcal{S}}\}_{t \in [T]}$	Black-box advice provided to ST-CLIP (see Def. 4.1)
$A^{(t)} \in [0,1]$	Adv utilization; fraction of constraint satisfied by $c(\mathbf{a}_1) + \cdots + c(\mathbf{a}_t)$

A Deferred Examples

In this section, we detail two more examples of applications that motivate the SOAD problem introduced in the main body, picking up from Section 2.2.

Carbon-aware or cost-aware autonomous electric vehicle charging. Consider an autonomous electric vehicle taxi (AEV) servicing a city [73] with multiple charging stations. Suppose that by the end of a day (i.e., deadline T), the AEV must replenish the charge that it will have used throughout the day. Service costs $f_t^{(u)}$ can represent either the carbon emissions of charging at location u during time slot t, or the charging cost plus opportunity cost of charging at location u during time slot t. We note that even within a single city, the *locational marginal emissions* (i.e., the carbon intensity of electricity at a specific location) may vary significantly [14], and charging prices can be similarly variable based on e.g., time-of-use and/or zonal energy pricing [51] The metric space (X, d) and the spatial movement cost $d(u_{t-1}, u_t)$ capture either the carbon overhead (in terms of "wasted" electricity) or the opportunity/time cost of moving to a different location for charging. The temporal switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\beta)}$ captures the small overhead of stopping or restarting charging at a single location, due to extra energy or time spent connecting or disconnecting from the charger at location u. Finally, the constraint function $c(\mathbf{x}_t)$ captures how much charge is delivered during time t according to decision $\mathbf{x}_t$, where an $\mathbf{x}_t$ that places a full allocation in the $\mathtt{OFF}^{(u)}$ state indicates that the AEV is serving customers (i.e., not charging). Since the AEV may move around in the city while serving customers, $\mathbf{x}_t$ should be updated exogenously to reflect the true state. We note that SOAD may not capture the case where the distance between charging stations is large (i.e., moving to a different location incurs substantial discharge) or the case where the AEV is at risk of fully discharging the battery before time T, which would require immediate charging.

Allocating tasks to volunteers. Consider a non-profit that has a task to complete before some short-term deadline T, with several locations and scheduled time slots for volunteer efforts (e.g., stores, community centers) throughout a region. Using platforms such as VolunteerMatch [56], volunteers can signal their interests in tasks (e.g., via a ranking) and availability – in assigning this task, the non-profit may want to maximize the engagement of their assigned volunteer(s). Service costs $f_t^{(u)}$ can represent the aggregate rankings of the volunteers present at location u during time slot t, where a lower number means that they are more interested in a given task. The metric space (X, d) and the spatial movement cost $d(u_{t-1}, u_t)$ can capture the cost of e.g., moving supplies that must be present at the location to work on the task. The temporal switching cost $\|\mathbf{x}_t - \mathbf{x}_{t-1}\|_{\ell_1(\beta)}$ may capture the cost of e.g., setting up or breaking down the setup required to work on the task at a given location u. Finally, the constraint function $c(\mathbf{x}_t)$ captures how much of the task can be completed during time t according to assignment decision $\mathbf{x}_t$. We note that the fractional allocation to $\mathtt{ON}$ / $\mathtt{OFF}$ states specified by SOAD may not be useful in this setting because e.g., groups of volunteers may not be fractionally divisible.

B   Supplemental Experiments

In this section, we present additional results and figures to complement those in the main body. In the first few results, we present additional experiments manipulating parameters using the *average carbon intensity signal*. Then, in Appendix B.1, we present a supplemental slate of experiments using the *marginal carbon intensity signal* obtained from WattTime [72].

Experiment VI: *Effect of job length J.* Fig. 10 plots the average empirical competitive ratio for fixed job lengths $J \in \{1, \ldots, 10\}$, where J is the length of the job in hours. In this experiment, we fix each job's data size to $G = 4$. As J increases, the empirical competitive ratio of PCM improves, and ST-CLIP remains consistent. We note that the simple threshold technique is able to achieve good performance in the case when $J = 1$ – intuitively, since simple threshold is agnostic to the switching overhead, its performance degrades when it uses more than one opportunity to migrate between regions (i.e., when the job takes more than one time slot).

Location	AWS Region	Average Carbon Intensity (in gCO_2eq/kWh) [57]			Marginal Carbon Intensity (in gCO_2eq/kWh) [72]		
		Duration	Min.	Max.	Duration	Min.	Max.
🇺🇸 Virginia, U.S.	`us-east-1`		293	567		48	1436
🇺🇸 California, U.S.	`us-west-1`		83	451	01/01/2022 - 12/31/2022	67	1100
🇺🇸 Oregon, U.S.	`us-west-2`		42	682		427	2000
🇨🇦 Quebec, Canada	`ca-central-1`	01/01/2020 - 12/31/2022	26	109		887	1123
🇬🇧 London, U.K.	`eu-west-2`		56	403	Hourly granularity	706	1082
🇫🇷 France	`eu-west-3`		18	199		549	1099
🇸🇪 Sweden	`eu-north-1`	Hourly granularity	12	59	8,760 data points	438	2556
🇩🇪 Germany	`eu-central-1`		130	765		11	1877
🇦🇺 Sydney, Australia	`ap-southeast-2`		267	817		12	1950
🇧🇷 Brazil	`sa-east-1`	26,304 data points	46	292			
🇿🇦 South Africa	`af-south-1`		586	785			
🇮🇱 Israel	`il-central-1`		514	589	Data not available		
🇮🇳 Hyderabad, India	`ap-south-2`		552	758			
🇰🇷 South Korea	`ap-northeast-2`		453	503			

Table 2. Summary of CO_2 data sets for each tested AWS region in our case study experiments, including the minimum and maximum carbon intensities, duration, granularity, and data availability.
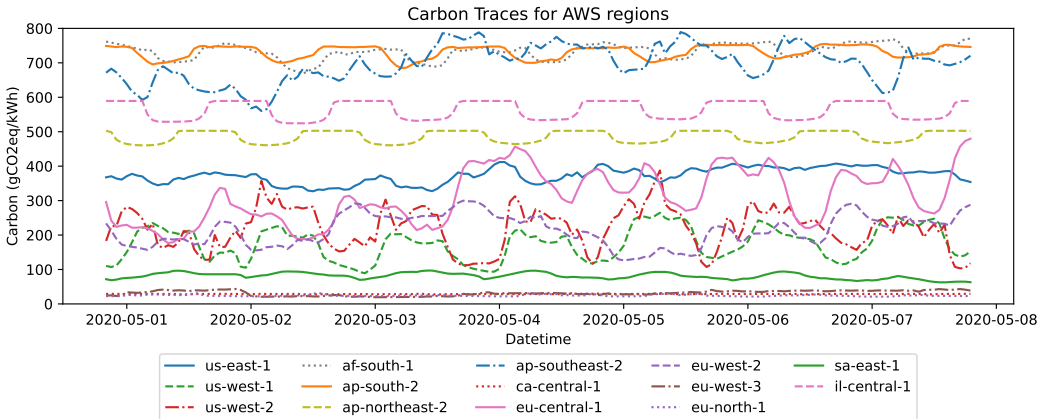

Fig. 9. Average carbon intensity traces [57] for all 14 AWS regions, over a week-long period in 2020.

As is characteristic of realistic cloud traces, the Google cluster traces we use in Experiments I-V are mostly composed of shorter jobs between 1 and 2 hours long. These results for fixed job lengths highlight that ST-CLIP and PCM do even better when given lengthy jobs – such jobs are less frequent but take up a disproportionate amount of compute cycles (and thus contribute disproportionately to the carbon footprint of) a typical data center.

Experiment VII: *Effect of temporal switching coefficient τ.*

Recall that increasing or decreasing τ simulates jobs that have more or less time-consuming checkpoint and resume overheads, respectively. In Fig. 11, we plot the average empirical competitive ratio for varying $\tau \in \{0, \ldots, 100\}$. In this experiment, we fix $J = 4$ and $G = 4$. Compared to varying
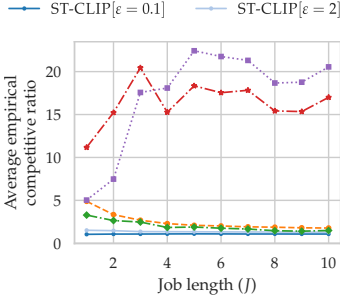
Fig. 10. Average empirical competitive ratios for varying *job length J*, with $G = 4, T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.
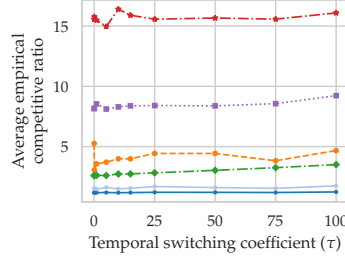
Fig. 11. Average empirical competitive ratios for varying *temporal switching coefficient* τ, with $G = 4, T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.
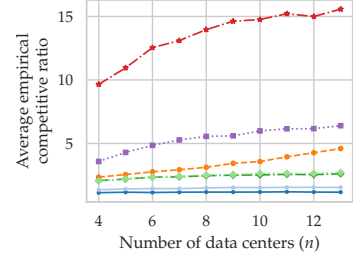
Fig. 12. Average empirical competitive ratios for varying *number of data centers n*, with $G = 4, T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.
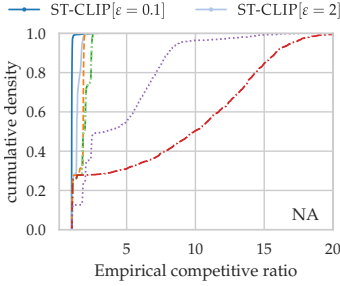

Fig. 13. CDFs of empirical competitive ratios for each tested algorithm on the North America subset.
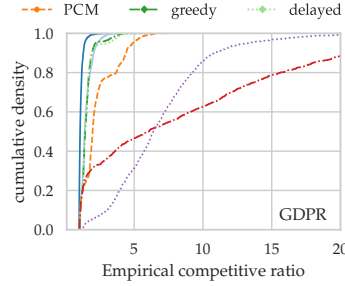
Fig. 14. CDFs of empirical competitive ratios for each tested algorithm on the EU (GDPR) region subset.
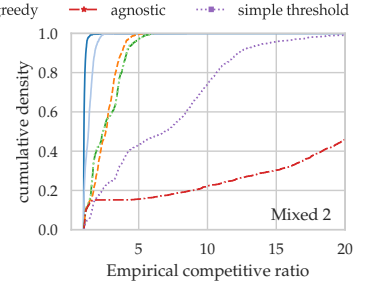
Fig. 15. CDFs of empirical competitive ratios for each tested algorithm on the second "mixed" subset.

G, τ has a smaller impact across the board, although as predicted by the theoretical bounds, the performance of PCM degrades slightly as τ grows, and the greedy policies are similarly affected.

Experiment VIII: *Effect of number of data centers n.*

Building off of the idea in Experiment IV, in Fig. 12 we plot the average empirical competitive ratio for varying $n \in [4, 13]$, where a random subset (of size n) is sampled from the base 14 regions for each batch job instance. For each job, we fix $G = 4$, a deadline T that is a random integer between 12 and 48, and use the average carbon intensity signal. We find that most algorithms' performance degrades as the size of the subset increases. This is likely because the expected range of carbon intensities expands as more diverse electric grids are included in the subset. As in previous experiments, ST-CLIP's performance with black-box advice is consistent as n increases.

Experiment IV (continued): *Effect of electric grids and data center availability.*

Continuing from Experiment IV in the main body, we present results where the metric space is constructed on four additional *subsets* of the 14 regions. By considering these region subsets, we approximate issues of data center availability, and electric grid characteristics that might face a deployment in practice.

Fig. 13 considers a North American subset of 4 regions: California, Oregon, Virginia, and Quebec. Under this subset, PCM slightly outperforms greedy and delayed greedy by 8.31% and 8.24%, respectively. Fig. 14 considers an "EU / GDPR" subset that includes 4 regions: France, Germany, London, and Sweden. On this subset, with a large proportion of consistent low-carbon grids (i.e., both Sweden and France), the greedy and delayed greedy policies outperform PCM by 21.54% and

20.25%, respectively. These results highlight a "best case" situation where the greedy policies are able to outperform PCM and nearly match the performance of ST-CLIP. Fig. 15 considers a second "mixed" subset of 7 regions: California, South Korea, Germany, Hyderabad, Israel, Sweden, and South Africa. On this subset, with a geographically distributed mix of high and low-carbon grids, PCM outperforms greedy and delayed greedy by 3.74% and 4.73%, respectively.

We briefly note that since the greedy policies exhibit fairly good performance across many of these experiments, for the intended application of carbon-aware workload management in data centers it may be worthwhile to evaluate the performance of ST-CLIP when given black-box advice ADV that simply encodes the decisions of the greedy policy. Since the black-box advice model can accommodate any arbitrary sequence of decisions, including heuristics, such a composition may achieve a favorable trade-off between average-case performance and worst-case guarantees if e.g., machine-learned forecasts are not available.

Experiment IX: *Runtime (wall clock) overhead measurements.*

In this experiment, we measure the wall clock runtime of each tested algorithm. Our experiment implementations are in Python, using NumPy [34], SciPy [71], and CVXPY [25] – we use the `time.perf_counter()` module in Python to calculate the total runtime (in milliseconds) for each algorithm on each instance, and report this value *normalized by the deadline* to give the *per-slot* (i.e., hourly) overhead of each algorithm.

Fig. 16 reports these measurements for batch jobs with deadlines $T \in \{6, ..., 48\}$, fixed $G = 4$, and job lengths from the Google trace are truncated to $T/2$ if necessary. For this experiment, we run each algorithm and each instance in a single thread on a MacBook Pro with M1 Pro processor and 32 GB of RAM.

Fig. 16. Average per-slot wall clock runtime for *instance sizes (i.e., deadlines)* $T \in \{6, ..., 48\}$, with $G = 4$.

We find that the average per-slot (i.e., once per hour) runtimes of PCM and ST-CLIP are 2.3 milliseconds and 24.3 milliseconds, respectively – this reflects the relative complexity of the optimization problems being solved in each, and the runtime is steady in the size of the instance, as is expected from an online algorithm. The optimal solution takes, on average, 173.8 milliseconds per slot to solve, although note that it finds the solution for all time slots at once. As the size of the instance (T) grows, this per-slot time slightly increases. Intuitively, the decision rule-based algorithms (carbon agnostic, simple threshold, and the greedy policies (not included in the plot)) have the lowest, and functionally negligible, runtime. This last result suggests that one effective way to reduce the impact of PCM and ST-CLIP's runtime overhead would be to develop approximations that avoid computing the exact solution to the minimization problem. However, for carbon intensity signals that are updated every 5 minutes to one hour, the overhead of PCM and ST-CLIP is likely reasonable in practice.

B.1 Marginal Carbon Intensity

In contrast to average carbon intensity, the marginal carbon intensity signal calculates the emissions of the generator(s) that are responding to changes in load on a grid at a certain time. From WattTime [72], we obtain data for 9 of the 14 regions we consider, spanning all of 2022. This data also includes carbon intensity forecasts published by WattTime, and we use these forecasts directly instead of generating synthetic forecasts as in Section 6.1. In Fig. 17, we plot a one-week sample of carbon intensity data to motivate this visually.

Fig. 17. Marginal carbon intensity traces [57] for 9 AWS regions, over a week-long period in 2022.

Fig. 18. CDFs of competitive ratios for each algorithm, across all marginal carbon intensity experiments.

Fig. 19. Average competitive ratios using marginal carbon data, with varying *job length J* and $G = 4$, $T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.

Fig. 20. Average competitive ratios using marginal carbon data, with varying *job data size G* and $T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$.

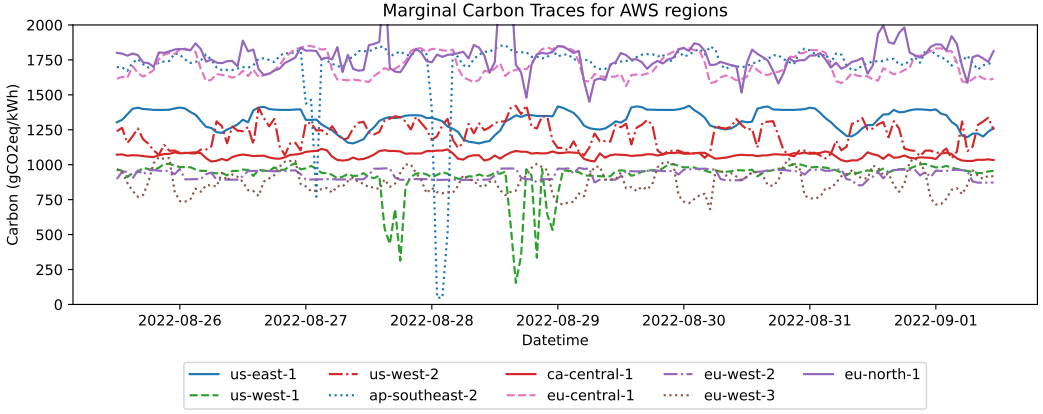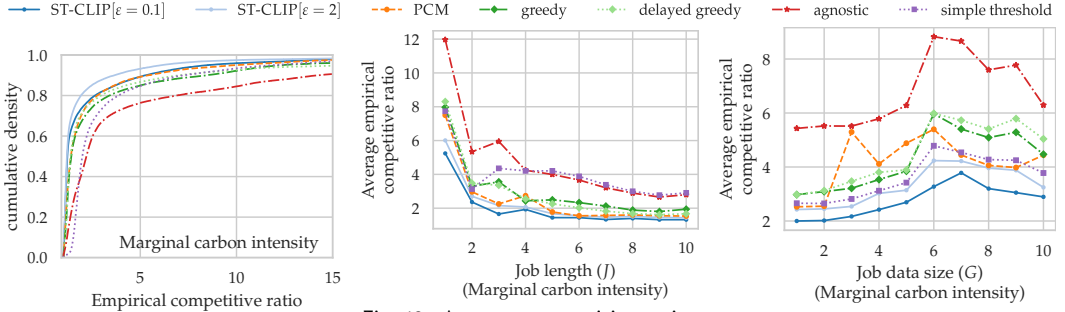A top-level summary of these experiments is given in Fig. 18 – this plot gives a cumulative distribution function (CDF) of the empirical competitive ratios for all tested algorithms, aggregating over all experiments that use the marginal carbon intensity signal. In these experiments, we consider all of the 9 regions for which the marginal data is available, and each job's deadline T is a random integer between 12 and 48 (henceforth denoted by $T \sim \text{Unif}_{\mathbb{Z}}(12, 48)$).

In these experiments, we observe differences that are likely attributable to the characteristic behavior of the marginal carbon intensity signal, which generally represents the high emissions rate of a quick-to-respond generator (e.g., a gas turbine) unless the supply of renewables on the grid exceeds the current demand. However, the relative ordering of performance has been largely preserved.

Interestingly, we note that the real forecasts are worse in the marginal setting compared to the average carbon intensity signal, again likely because of the characteristics of marginal carbon. Rather than predicting e.g., the diurnal patterns of an average signal, predicting marginal carbon requires a model to pick out specific time slots where curtailment is expected to occur.

Despite these challenges of forecasting the marginal signal, ST-CLIP with $\varepsilon = 2$ outperforms the baselines in both average and worst-case performance, improving on the closest greedy policy by an average of 12.04%, and outperforming delayed greedy, simple threshold, and carbon-agnostic by averages of 11.02%, 37.12%, and 85.67%, respectively.

Marginal Experiment I: *Effect of job length J.*

In Fig. 19, we plot the average competitive ratio for different job lengths $J \in \{1, \ldots, 10\}$. Each job has $G = 4$ and an integer deadline T randomly sampled between 12 and 48. Compared to the greedy policies, our robust baseline PCM performs favorably in these experiments, which is likely due to a combination of differences in the marginal setting and the less performant forecasts. Similar to the average carbon setting, the simple threshold technique performs well when the job length is short, but performance suffers when it has more opportunities for migration.

Marginal Experiment II: *Effect of job data size G.*

In Fig. 20, we plot the average empirical competitive ratio for different job data sizes $G \in \{1, \ldots, 10\}$ using the marginal carbon signal. As predicted by the theoretical bounds, and observed in the main body for the average carbon experiments, the performance of PCM degrades as G grows – we observe the same effect for the greedy policies and the simple threshold algorithm. The performance of ST-CLIP also grows, which suggests that larger migration overhead does have an impact when the advice suffers from a lack of precision due to the more challenging setting for forecasts posed by marginal carbon intensity.

C Proofs for Section 2 (Problem Formulation, Challenges and Technical Preliminaries)

In the following, we prove Theorem 2.5, which shows that the expected cost of any randomized SOAD decision $\mathbf{p}_t \in \Delta_{\mathcal{S}}$ is equivalent to that of a decision which chooses a point in X probabilistically according to the distribution of $\mathbf{p}_t$ and then interprets the ON / OFF probabilities at that point as deterministic allocations in $\mathcal{X}$.

PROOF OF THEOREM 2.5. Suppose that $\mathbf{p}_t, \mathbf{p}_{t-1}$ are probability distributions over the randomized state space $\Delta_{\mathcal{S}}$, and let $\mathbb{E}\left[\text{Cost}(\mathbf{p}_t, \mathbf{p}_{t-1})\right]$ denote the expected cost of decision $\mathbf{p}_t$. This cost is defined as follows:

$$\mathbb{E}\left[\text{Cost}(\mathbf{p}_t, \mathbf{p}_{t-1})\right] = \mathbb{E}\left[f_t(\mathbf{p}_t) + g(\mathbf{p}_t, \mathbf{p}_{t-1})\right].$$

Recall that because the cost function f_t is linear and separable, the expectation can be written as:

$$\mathbb{E}\left[f_t(\mathbf{p}_t)\right] = \sum_{u \in X} f_t^{(u)} p_t^{\mathrm{ON}^{(u)}}.$$

For any $\mathbf{p}_t \in \Delta_{\mathcal{S}}$, let $\mathbf{r}_t := \{r_t^{(u)} \leftarrow p_t^{\mathrm{ON}^{(u)}} + p_t^{\mathrm{OFF}^{(u)}} : u \in X\} \in \Delta_X$, i.e., a vector that aggregates the total probabilities across the states space $\mathcal{S}$ at each point of X.

We note that by disaggregating the spatial and temporal switching costs, we have that the expectation $\mathbb{E}\left[g(\mathbf{p}_t, \mathbf{p}_{t-1})\right]$ can be written in terms of the Wasserstein-1 distance with respect to the underlying metric X and a linear temporal term that depends on the probability assigned to the OFF state. This is the case because the optimal transport plan $\mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1})$ must always involve first moving probability mass to/from the OFF and ON states at each point of X, and then within the spatial metric – this follows since SOAD defines that movement within the spatial metric X can only be made between ON states – i.e., a player moving from $\mathrm{OFF}^{(u)}$ to $\mathrm{OFF}^{(v)}$ must first traverse to $\mathrm{ON}^{(u)}$, then through the metric X, and finally through $\mathrm{ON}^{(v)}$.

$$\mathbb{E}\left[g(\mathbf{p}_t, \mathbf{p}_{t-1})\right] = \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \sum_{u \in X} \beta^{(u)} |p_t^{\mathrm{OFF}^{(u)}} - p_{t-1}^{\mathrm{OFF}^{(u)}}|.$$

Thus, the expected cost of $\mathbf{p}_t$ can be written as:

$$\mathbb{E}\left[\text{Cost}(\mathbf{p}_t, \mathbf{p}_{t-1})\right] = \sum_{u \in X} f_t^{(u)} p_t^{\mathrm{ON}^{(u)}} + \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \sum_{u \in X} \beta^{(u)} |p_t^{\mathrm{OFF}^{(u)}} - p_{t-1}^{\mathrm{OFF}^{(u)}}|.$$

In the mixed probabilistic/deterministic setting, the true allocation to $\text{ON}^{(u)}$ (denoted by $\tilde{p}_t^{\text{ON}^{(u)}}$) for any point $u \in X$ is defined as $\tilde{p}_t^{\text{ON}^{(u)}} = p_t^{\text{ON}^{(u)}}/r_t^{(u)}$ (conversely, we have $\tilde{p}_t^{\text{OFF}^{(u)}} = p_t^{\text{OFF}^{(u)}}/r_t^{(u)}$). Letting $\mathbb{L}_t^{(u)} \in \{0, 1\}$ denote an indicator variable that encodes the player's location (i.e., point) at time t (0 if player is not at u, 1 if player is at u), the expected cost of $\tilde{\mathbf{p}}_t$ can be written as follows:

$$\mathbb{E}\left[\text{Cost}(\tilde{\mathbf{p}}_t, \tilde{\mathbf{p}}_{t-1})\right] = \mathbb{E}\left[\sum_{u \in X} \mathbb{L}_t^{(u)} f_t^{(u)} \tilde{p}_t^{\text{ON}^{(u)}}\right] + \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \mathbb{E}\left[\sum_{u \in X} \left|\mathbb{L}_t^{(u)} \beta^{(u)} \tilde{p}_t^{\text{OFF}^{(u)}} - \mathbb{L}_{t-1}^{(u)} \beta^{(u)} \tilde{p}_{t-1}^{\text{OFF}^{(u)}}\right|\right].$$

Noting that $\mathbb{E}\left[\mathbb{L}_t^{(u)}\right] = r_t^{(u)}$ by linearity of expectation we have:

$$\mathbb{E}\left[\text{Cost}(\tilde{\mathbf{p}}_t, \tilde{\mathbf{p}}_{t-1})\right] = \sum_{u \in X} r_t^{(u)} f_t^{(u)} \tilde{p}_t^{\text{ON}^{(u)}} + \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \sum_{u \in X} |r_t^{(u)} \beta^{(u)} \tilde{p}_t^{\text{OFF}^{(u)}} - r_{t-1}^{(u)} \beta^{(u)} \tilde{p}_{t-1}^{\text{OFF}^{(u)}}|.$$

Furthermore, recalling the definitions of $\tilde{p}_t^{\text{ON}^{(u)}}$ and $\tilde{p}_t^{\text{OFF}^{(u)}}$, we have the following:

$$\mathbb{E}\left[\text{Cost}(\tilde{\mathbf{p}}_t, \tilde{\mathbf{p}}_{t-1})\right] = \sum_{u \in X} f_t^{(u)} p_t^{\text{ON}^{(u)}} + \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \sum_{u \in X} \beta^{(u)} |p_t^{\text{OFF}^{(u)}} - p_{t-1}^{\text{OFF}^{(u)}}|.$$

Recalling that $\mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) = \mathbb{W}^1(\mathbf{r}_t, \mathbf{r}_{t-1}) + \sum_{u \in X} \beta^{(u)} |p_t^{\text{OFF}^{(u)}} - p_{t-1}^{\text{OFF}^{(u)}}|$ by the structure of the spatial and temporal switching costs completes the proof, since $\mathbb{E}\left[\text{Cost}(\mathbf{p}_t, \mathbf{p}_{t-1})\right] = \mathbb{E}\left[\text{Cost}(\tilde{\mathbf{p}}_t, \tilde{\mathbf{p}}_{t-1})\right]$, and thus the expected cost is equivalent if a point (location) is first chosen probabilistically and the ON / OFF probabilities at that point are then interpreted as deterministic (fractional) allocations. □

D Proofs for Section 3 (A Competitive Online Algorithm)

D.1 Convexity of the pseudo-cost minimization problem in PCM

In this section, we prove Theorem 3.1, which states that the pseudo-cost minimization problem central to the design of PCM is a convex minimization problem, implying that it can be solved efficiently.

For convenience, let $h_t(\mathbf{k}) : t \in [T]$ represent the pseudo-cost minimization problem's objective for a single arbitrary time step:

$$h_t(\mathbf{k}) = f_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\bar{c}(\mathbf{k})} \psi(u) du. \tag{9}$$

PROOF OF THEOREM 3.1. We prove the statement by contradiction. By definition, the sum of two convex functions gives a convex function. Since $\|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})}$ is a norm and $\mathbf{k}_{t-1}$ is fixed, by definition it is convex. We have also assumed as part of the problem setting that each $f_t(\mathbf{k})$ is linear. Thus, $f_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})}$ must be convex. The remaining term is the negation of $\int_{z^{(t-1)}}^{z^{(t-1)}+\bar{c}(\mathbf{k})} \psi(u) du$. Let $w(\bar{c}(\mathbf{k})) = \int_{z^{(t-1)}}^{z^{(t-1)}+\bar{c}(\mathbf{k})} \psi(u) du$. By the fundamental theorem of calculus, we have

$$\nabla w(\bar{c}(\mathbf{k})) = \psi(z^{(t-1)} + \bar{c}(\mathbf{k}))\nabla\bar{c}(\mathbf{k}).$$

Let $b(\bar{c}(\mathbf{k})) = \psi(z^{(t-1)} + \bar{c}(\mathbf{k}))$. Then we have

$$\nabla^2 w(\bar{c}(\mathbf{k})) = \nabla^2\bar{c}(\mathbf{k})w(\bar{c}(\mathbf{k})) + \nabla\bar{c}(\mathbf{k})b'(\bar{c}(\mathbf{k}))\nabla\bar{c}(\mathbf{k})^\top.$$

Since $\bar{c}(\mathbf{k})$ is piecewise linear by the definition of SOAD, we know that $\nabla^2\bar{c}(\mathbf{k})b(\bar{c}(\mathbf{k})) = 0$. Since ψ is monotonically decreasing on the interval $[0, 1]$, we know that $b'(\bar{c}(\mathbf{k})) < 0$, and thus $\nabla\bar{c}(\mathbf{k})b'(\bar{c}(\mathbf{k}))\nabla\bar{c}(\mathbf{k})^\top$ is negative semidefinite. This implies that $w(\bar{c}(\mathbf{k}))$ is concave in $\mathbf{k}$.

Since the negation of a concave function is convex, this causes a contradiction, because the sum of two convex functions gives a convex function. Thus, $h_t(\cdot) = f_t(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\bar{c}(\mathbf{k})} \psi(u) du$ is always convex under the assumptions of SOAD. □

By showing that $h_t(\cdot)$ is convex, it follows that the pseudo-cost minimization (2) in PCM is a convex minimization problem (i.e., it can be solved efficiently using numerical methods).

D.2 Proof of Theorem 3.2

In the following, we prove Theorem 3.2. In what follows, we let $\mathcal{I} \in \Omega$ denote an arbitrary valid SOAD instance. Let $z^{(j)} = \sum_{t \in [T]} \overline{c}(\mathbf{k}_t)$ denote the final utilization before the mandatory allocation. Also note that $z^{(t)} = \sum_{m \in [t]} \overline{c}(\mathbf{k}_m)$ is non-decreasing over t.

In what follows, we let η be defined as the solution to $\ln\left(\frac{U-L-D-2\tau}{U-U/\eta-D}\right) = \frac{1}{\eta}$, which has a closed form given by:

$$\eta \coloneqq \left[W\left(\frac{(D+L-U+2\tau)\exp\left(\frac{D-U}{U}\right)}{U} \right) + \frac{U-D}{U} \right]^{-1}. \tag{10}$$

Note that setting η as above satisfies the following equality within the pseudo-cost function ψ (defined in Def. 3.1):

$$\psi(1) = U - \tau + (U/\eta - U + D + \tau)\exp(z/\eta) = L + D.$$

We start by proving Lemma 3.4, which states that OPT is lower bounded by

$$\text{OPT}(\mathcal{I}) \geq \frac{\max\left\{\psi(z^{(j)}) - D, L\right\}}{O(\log n)}.$$

Proof of Lemma 3.4.   Without loss of generality, denote the minimum gradient of any cost function (excluding OFF states) by $\nabla_{\min}$. Suppose that a cost function f_m with $\nabla_{\min}$ gradient (at a dimension corresponding to any ON state) arrives at time step m.
Recall that PCM solves the following pseudo-cost minimization problem at time m:

$$\mathbf{k}_m = \underset{\mathbf{k} \in K : \overline{c}(\mathbf{k}) \leq 1 - z^{(t-1)}}{\arg\min} f_m(\mathbf{k}) + \|\mathbf{k} - \mathbf{k}_{m-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(m-1)}}^{z^{(m-1)}+\overline{c}(\mathbf{k})} \psi(u)du$$

By assumption, since $f_m(\cdot)$ is linear and satisfies $\nabla f_m < \psi(z^{(j)}) - D$, there must exist a dimension in f_m (i.e., a service cost associated with an ON state) that satisfies the following. Let $\text{ON}^{[d]} \subset [d]$ denote the index set (i.e., the dimensions in $\mathbf{k}$) that correspond to allocations in ON states.

$$\exists i \in \text{ON}^{[d]} : f_m(\mathbf{k})_i \leq [\nabla_{\min} \cdot \overline{c}(\mathbf{k})]_i.$$

Also note that $\|\mathbf{k} - \mathbf{k}_{m-1}\|_{\ell_1(\mathbf{w})}$ is upper bounded by $(D+\tau)\overline{c}(\mathbf{k})$, since in the worst-case, PCM must pay the max movement and switching cost to move the allocation to the "furthest" point and make a decision $\mathbf{k}$.

Since ψ is monotone decreasing on the interval $z \in [0, 1]$, by definition we have that $\mathbf{k}_m$ solving the true pseudo-cost minimization problem is lower-bounded by the $\check{\mathbf{k}}_m$ solving the following minimization problem (specifically, the constraint satisfaction satisfies $\overline{c}(\check{\mathbf{k}}_m) \leq \overline{c}(\mathbf{k}_m)$):

$$\check{\mathbf{k}}_m = \underset{\mathbf{k} \in K : \overline{c}(\mathbf{k}) \leq 1 - z^{(t-1)}}{\arg\min} \nabla_{\min} \cdot \overline{c}(\mathbf{k}) + D\overline{c}(\mathbf{k}) + \tau\overline{c}(\mathbf{k}) - \int_{z^{(m-1)}}^{z^{(m-1)}+\overline{c}(\mathbf{k})} \psi(u)du. \tag{11}$$

By expanding the right hand side, we have:

$$\nabla_{\min} \cdot \overline{c}(\mathbf{k}) + D\overline{c}(\mathbf{k}) + \tau\overline{c}(\mathbf{k}) - \int_{z^{(m-1)}}^{z^{(m-1)}+\overline{c}(\mathbf{k})} \psi(u)du$$

$$\nabla_{\min} \cdot \overline{c}(\mathbf{k}) + D\overline{c}(\mathbf{k}) + \tau\overline{c}(\mathbf{k}) - \int_{z^{(m-1)}}^{z^{(m-1)}+\overline{c}(\mathbf{k})} \left[U - \tau + (U/\eta - U + D + \tau)\exp(z/\eta) \right] du$$

$$(\nabla_{\min} - U + D + \tau)\overline{c}(\mathbf{k}) - \left[(\tau - U + D)\eta + U \right] \left(\exp\left(\frac{z^{(m-1)} + \overline{c}(\mathbf{k})}{\eta}\right) - \exp\left(\frac{z^{(m-1)}}{\eta}\right) \right)$$

Letting $\overline{c}(\mathbf{k})$ be some scalar y (which is valid since we assume there is at least one dimension $i \in [d]$ where the growth rate of $f_m(\cdot)$ is at most $\nabla_{\min}$), the pseudo-cost minimization problem finds the value y that minimizes the following quantity:

$$(\nabla_{\min} - U + D + \tau)y - \left[(\tau - U + D)\eta + U \right] \left(\exp\left(\frac{z^{(m-1)} + y}{\eta}\right) - \exp\left(\frac{z^{(m-1)}}{\eta}\right) \right)$$

Taking the derivative of the above with respect to y yields the following:

$$= \nabla_{\min} + D + \tau - U - \frac{\left[(\tau - U + D)\eta + U \right] \exp\left(\frac{z^{(m-1)}+y}{\eta}\right)}{\eta} \quad = \nabla_{\min} + D - \psi(z^{(m-1)} + y) \quad (12)$$

Note that since the minimization problem is convex by [Theorem 3.1](#), the unique solution to the above coincides with a point y where the derivative is zero. This implies that PCM will increase $\overline{c}(\mathbf{k})$ until $\nabla_{\min} = \psi(z^{(m-1)} + y) - D$, which further implies that $\psi(z^{(m)}) = \nabla_{\min} + D$.

Since this minimization $\check{\mathbf{k}}_m$ is a lower bound on the true value of $\mathbf{k}_m$, this implies that $\psi(z^{(t)}) - D$ is a lower bound on the minimum service cost coefficient (excluding OFF states) seen *so far* at time t. Further, the *final utilization* $z^{(j)}$ gives that the minimum service cost coefficient over any cost function over the entire sequence is lower bounded by $\psi(z^{(j)}) - D$.

Note that the best choice for OPT is to service the entire workload at the minimum service cost if it is feasible. Since the vector space $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ used by PCM has at most $O(\log n)$ distortion with respect to the underlying metric used by OPT (see [Definition 2.3](#)), this implies that $\text{OPT}(\mathcal{I}) \geq \frac{\max\{\psi(z^{(j)}) - D, L\}}{O(\log n)}$. $\qquad\qquad\square$

Next, we prove [Lemma 3.5](#), which states that the expected cost of $\text{PCM}(\mathcal{I})$ is upper bounded by $\mathbb{E}[\text{PCM}(\mathcal{I})] \leq \int_0^{z^{(j)}} \psi(u)du + (1 - z^{(j)})U + \tau z^{(j)}$.

Proof of [Lemma 3.5](#).   Recall that $z^{(t)} = \sum_{m \in [t]} \overline{c}(\mathbf{k}_m)$ is non-decreasing over t.

Observe that whenever $\overline{c}(\mathbf{k}_t) > 0$, we have that $f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} < \int_{z^{(t-1)}}^{z^{(t-1)}+\overline{c}(\mathbf{k}_t)} \psi(u)du$. Then, if $\overline{c}(\mathbf{k}_t) = 0$, which corresponds to the case when $\mathbf{k}_t$ allocates all of the marginal probability mass to OFF states, we have the following:

$$f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\overline{c}(\mathbf{k}_t)} \psi(u)du = 0 + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - 0 = \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} \quad (13)$$

This gives that for any time step where $\overline{c}(\mathbf{k}_t) = 0$, we have the following inequality, which follows by observing that from [Assumption 2.2](#), any marginal probability mass assigned to ON states in the previous time step can be moved to OFF states at a cost of at most $\tau\overline{c}(\mathbf{k}_{t-1})$.

$$f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} \leq \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} \leq \tau\overline{c}(\mathbf{k}_{t-1}), \forall t \in [T] : \overline{c}(\mathbf{k}_t) = 0. \quad (14)$$

Since any time step where $\bar{c}(\mathbf{k}_t) > 0$ implies that $f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} < \int_{z^{(t-1)}}^{z^{(t-1)}+\bar{c}(\mathbf{k}_t)} \psi(u)du$, we have the following inequality across all time steps (*i.e., an upper bound on the excess cost not accounted for by the pseudo-cost*):

$$f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\bar{c}(\mathbf{k}_t)} \psi(u)du \leq \tau\bar{c}(\mathbf{k}_{t-1}), \forall t \in [T]. \tag{15}$$

Thus, we have

$$\tau z^{(j)} = \sum_{t\in[j]} \tau\bar{c}(\mathbf{k}_{t-1}) \geq \sum_{t\in[j]} \left[f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} - \int_{z^{(t-1)}}^{z^{(t-1)}+\bar{c}(\mathbf{k}_t)} \psi(u)du \right] \tag{16}$$

$$= \sum_{t\in[j]} \left[f_t(\mathbf{k}_t) + \|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} \right] - \int_0^{z^{(j)}} \psi(u)du \tag{17}$$

$$= \text{PCM}(\mathcal{I}) - (1 - z^{(j)})U - \int_0^{z^{(j)}} \psi(u)du. \tag{18}$$

Combining Lemma 3.4 and Lemma 3.5 gives

$$\text{CR} \leq \frac{\mathbb{E}[\text{PCM}(\mathcal{I})]}{\text{OPT}(\mathcal{I})} \leq \frac{\int_0^{z^{(j)}} \psi(u)du + (1 - z^{(j)})U + \tau z^{(j)}}{\max\{\psi(z^{(j)}) - D, L\}} \leq \eta, \tag{19}$$

where the last inequality holds since for any $z \in [0, 1]$:

$$\int_0^z \psi(u)du + \tau z + (1 - z)U = \int_0^z \left[U - \tau + (U/\eta - U + D + \tau)\exp(z/\eta) \right] du + (1 - z)U + \tau z \tag{20}$$

$$= \left[((\tau - U + D)\eta + U) \, e^{\frac{u}{\eta}} + Uu - \tau u \right]_0^z + (1 - z)U + \tau z \tag{21}$$

$$= ((\tau - U + D)\eta + U) \, e^{\frac{z}{\eta}} - (\tau - U + D)\eta \tag{22}$$

$$= \eta \left[\left((\tau - U + D) + \frac{U}{\eta} \right) e^{\frac{z}{\eta}} - \tau + U - D \right] \tag{23}$$

$$= \eta[\psi(z) - D]. \tag{24}$$

This completes the proof of Theorem 3.2.   □

D.3   Proof of Theorem 3.3

In this section, we prove Theorem 3.3, which states that η (as defined in (4)) is the optimal competitive ratio for SOAD. To show this lower bound, we first define a family of special two-stage adversaries, a corresponding metric space X, and then show that the competitive ratio for any algorithm is lower bounded under the instances provided by these adversaries.

Prior work has shown that difficult instances for online search problems with a minimization objective occur when inputs arrive at the algorithm in an decreasing order of cost [27, 42, 50, 70]. For SOAD, we extend this idea and additionally consider how adaptive adversaries can strategically present good service cost functions at distant points in the metric first, followed by good service costs at the starting point (e.g., "at home"), to create a family of sequences that simultaneously penalize the online player for moving "too much" and for not moving enough.

We now formalize two such families of adversaries, namely $\{\mathcal{G}_y\}_{y\in[L,U]}$ and $\{\mathcal{A}_y\}_{y\in[L,U]}$, where $\mathcal{A}_y$ and $\mathcal{G}_y$ are both called y-adversaries.
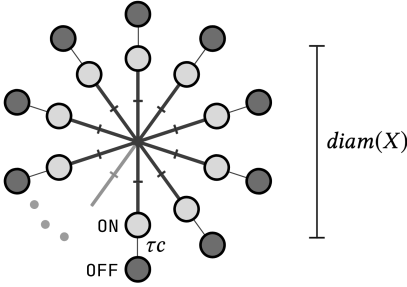
Fig. 21. A motivating illustration of the lower bound star metric considered in Definition D.3. Light gray circles represent the points of the metric space (X, d), and darker circles represent the OFF states of SOAD. Note that the distance between any two points in the metric is $\text{diam}(X) = Dc$.

DEFINITION D.1 (y-ADVERSARIES FOR SOAD). *Let $m \in \mathbb{N}$ be sufficiently large, and $\sigma := (U-L)/m$. The metric space X is a weighted star metric with n points, each with an ON and OFF state. For the constraint function $c(\cdot)$, we set one throughput constant for all ON states such that $c^{(u)} \ll 1 : u \in X$. This value is henceforth simply denoted by c. See Fig. D.3 for an illustration.*

The movement cost can be represented by a weighted ℓ_1 norm $\| \cdot \|_{\ell_1(\mathbf{w})}$ that combines the spatial distances given by the metric with the temporal switching cost. Recall Assumption 2.2 – at any single point u, the switching cost between the ON and OFF states is given by $\tau c \cdot |x_t^{\text{ON}^{(u)}} - x_{t-1}^{\text{ON}^{(b)}}|$ (i.e., for two arbitrary allocations $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$) for $\tau > 0$. Furthermore, for any two disjoint points in the metric $u, v : u \neq v$, the distance between $\text{ON}^{(u)}$ and $\text{ON}^{(v)}$ is exactly $Dc = \text{diam}(X)$.

Let y denote a value on the interval $[L, U]$, which represents the "best service cost function" presented by the y-adversary in their sequence. We define two distinct stages of the input.

*In **Stage 1**, the adversary presents two types of cost functions such that "good cost functions" are always at points that are distant to the online player. These cost functions are denoted (for convenience) as $Up(x) = Ux$, and $Down^i(x) = (U - i\sigma)x$, where one such cost function is delivered at each point's ON state, at each time step.*

Without loss of generality, let the starting point be $s \in X$ and the start state be $\text{OFF}^{(s)}$ (for both ALG and OPT). In the first time step, the adversary presents cost function $Up(x) = Ux$ at the starting point's ON state (i.e., $\text{ON}^{(s)}$), and $Down^1(x)$ at all of the other $(n - 1)$ ON states.

If ALG ever moves a non-zero fractional allocation to an ON state other than the starting point, that point becomes inactive, meaning that the adversary will present $Up(x)$ at that location in the next time step and for the rest of the sequence.

*In the second time step, the adversary presents cost function $Up(x) = Ux$ at the starting point and any inactive points, and $Down^1(x)$ at all of the other ON states. The adversary continues to sequentially present $Down^1(x)$ in this manner until it has presented it at least μ times (where $\mu := 1/c$). It then moves on to present $Down^2(x)$, $Down^3(x)$, and so forth. The adversary follows the above pattern, presenting "good cost functions" to a shrinking subset of ON states until they present $Down^{m_y}(x) = yx$ up to μ times at any remaining active states. In the time step after the last $Down^{m_y}(x) = yx$ is presented, the adversary presents $Up(x)$ everywhere, and **Stage 1** ends.*

*In **Stage 2**, the adversary presents $Up(x)$ and $Down^i(x)$ cost functions at the starting point s, in an alternating fashion. All other ON states are considered inactive in this stage, so they only receive $Up(x)$. In the first time step, the adversary presents $Down^1(x)$ at the starting point, followed by $Up(x)$ in the following time step. In the third time step, the adversary presents $Down^2(x)$ at the starting point, followed by $Up(x)$ in the subsequent time step. The adversary continues alternating in this manner until they present $Down^{m_y}(x) = yx$ at the starting point. In the $\mu - 1$ time steps after $Down^{m_y}(x) = yx$ is presented, the adversary presents $Down^{m_y}(x) = yx$ at the starting point (allowing OPT to reduce their switching cost). Finally, the adversary presents $Up(x)$ everywhere for the final μ time steps, and **Stage 2** ends.*

The first family of y-adversaries, $\{\mathcal{G}_y\}_{y \in [L,U]}$, only uses **Stage 1** – the sequence ends when **Stage 1** ends. The second family, $\{\mathcal{A}_y\}_{y \in [L,U]}$, sequentially uses both stages – cost functions are presented using the full **Stage 1** sequence first, which is then followed by the full **Stage 2** sequence. This concatenated sequence ends when **Stage 2** ends.

Note that the final cost function for any point in any y-adversary instance is always $Up(x)$.

PROOF OF THEOREM 3.3. Let $s(y)$ and $t(y)$ denote *constraint satisfaction functions* mapping $[L, U] \rightarrow [0, 1]$, that fully describe ALG's expected deadline constraint satisfaction (i.e., $\mathbb{E}\left[\sum_{t \in [j]} \bar{c}(\mathbf{k})\right]$ before the mandatory allocation) during **Stage 1** and **Stage 2**, respectively.

Note that for large m, **Stage 1** and **Stage 2** for $y = z - \sigma$ are equivalent to first processing **Stage 1** and **Stage 2** for $y = z$, and then processing an additional batch of cost functions such that the best cost function observed is $\text{Down}^{m_{z-\sigma}}(x) = (z - \sigma)x$.

As the expected deadline constraint satisfactions at each time step are unidirectional (irrevocable), we must have that $s(y)$ and $t(y)$ both satisfy $s(y - \sigma) \geq s(y)$ and $t(y - \sigma) \geq t(y)$, i.e. $s(y)$ and $t(y)$ are non-decreasing on $[L, U]$. Note that the optimal solutions for adversaries $\mathcal{G}_y$ and $\mathcal{A}_y$ are not the same.

In particular, we have that $\text{OPT}(\mathcal{G}_y) = \min\{y + D + \tau, U\}$. For relatively large y, the optimal solution may choose to satisfy the constraint at the starting point, but for sufficiently small y, the optimal solution on $\mathcal{G}_y$ may choose to move to a distant point. For $\mathcal{A}_y$, since the "good cost functions" arrive at the starting point, we have $\text{OPT}(\mathcal{A}_y) = y$ for any $y \in [L, U]$.

Due to the adaptive nature of the y-adversary, any ALG incurs expected movement cost proportional to $s(y)$ during **Stage 1**. Furthermore, since $c \ll 1$ for all ON states, note that as soon as $s(y) > c$, in expectation, ALG has *moved away from* the starting point. Thus, during **Stage 2**, ALG must also incur expected movement cost proportional to $t(y)$ as it "moves back" to the starting point. Let $\mathbf{l} = U/\eta^\star - 2\tau$ denote the worst marginal cost that an $\eta^\star$-competitive ALG should be willing to accept in either stage. The total expected cost incurred by an $\eta^\star$-competitive online algorithm ALG on adversaries $\mathcal{A}_y$ and $\mathcal{G}_y$ can be expressed as follows:

$$\mathbb{E}[\text{ALG}(\mathcal{G}_y)] = s(\mathbf{l})\mathbf{l} - \int_1^y uds(u) + Ds(y) + (1 - s(y))U + 2\tau s(y) \tag{25}$$

$$\mathbb{E}[\text{ALG}(\mathcal{A}_y)] = s(\mathbf{l})\mathbf{l} - \int_1^y uds(u) + Ds(y) + t(\mathbf{l})\mathbf{l} - \int_1^y udt(u) + Dt(y) + \tag{26}$$

$$(1 - s(y) - t(y))U + 2\tau\left[s(y) + t(y)\right] \tag{27}$$

In the above expressions, $uds(u)$ is the expected cost of buying $ds(u)$ constraint satisfaction at cost u. The same convention extends to $udt(u)$. $Ds(y)$ and $Dt(y)$ represent the movement cost paid by ALG during **Stage 1** and **Stage 2**, respectively, given that "good service cost functions" arrive at distant points (i.e. at distance D). $(1 - s(y))U$ and $(1 - s(y) - t(y))U$ give the expected cost of the mandatory allocation on adversary $\mathcal{G}_y$ and $\mathcal{A}_y$, respectively. Similarly, 2τ gives the expected temporal switching cost due to allocation decisions made (excepting the mandatory allocation).

For any $\eta^\star$-competitive algorithm, the constraint satisfaction functions $s(\cdot)$ and $t(\cdot)$ must simultaneously satisfy $\mathbb{E}[\text{ALG}(\mathcal{G}_y)] \leq \eta^\star \text{OPT}(\mathcal{G}_y)$ and $\mathbb{E}[\text{ALG}(\mathcal{A}_y)] \leq \eta^\star \text{OPT}(\mathcal{A}_y)$ for all $y \in [L, U]$. This gives a necessary condition that the functions must satisfy as follows:

$$s(\mathbf{l})\mathbf{l} - \int_1^y uds(u) + Ds(y) + (1 - s(y))U + 2\tau s(y) \leq \eta^\star\left[y + D + 2\tau\right]$$

$$s(\mathbf{l})\mathbf{l} - \int_1^y uds(u) + t(\mathbf{l})\mathbf{l} - \int_1^y udt(u) + (1 - s(y) - t(y))U + [D + 2\tau](s(y) + t(y)) \leq \eta^\star[y]$$

By integral by parts, the above expressions imply that the constraint satisfaction functions $s(y)$ and $t(y)$ must satisfy the following conditions:

$$s(y) \geq \frac{U - \eta^\star y - \eta^\star D - 2\eta^\star \tau}{U - y - D - 2\tau} - \frac{1}{U - y - D - 2\tau} \int_1^y s(u)du \tag{28}$$

$$t(y) \geq \frac{(y + D - U + 2\tau)s(y)}{U - y - D - 2\tau} + \frac{U - \eta^\star y}{U - y - D - 2\tau} - \frac{1}{U - y - D - 2\tau} \int_1^y s(u) + t(u)du \tag{29}$$

$$\geq \frac{\eta^\star D + \eta^\star 2\tau}{U - y - D - 2\tau} - \frac{1}{U - y - D - 2\tau} \int_1^y t(u)du \tag{30}$$

In what follows, we substitute $\mathbf{l} = \frac{U}{\eta} - 2\tau$. By Grönwall's Inequality [58, Theorem 1, p. 356], we have the following:

$$s(y) \geq \frac{U - \eta^\star y - \eta^\star D - 2\eta^\star \tau}{U - y - D - 2\tau} - \int_1^y \frac{U - \eta^\star u - \eta^\star D - 2\eta^\star \tau}{(U - u - D - 2\tau)^2} du$$

$$\geq \frac{U - \eta^\star y - \eta^\star D - 2\eta^\star \tau}{U - y - D - 2\tau} - \left[\frac{2\tau\eta^\star + U\eta^\star - 2\tau - U}{u - U + D + 2\tau} - \eta^\star \ln(U - u - D - 2\tau) \right]_1^y$$

$$\geq \eta^\star \ln(U - y - D - 2\tau) - \eta^\star \ln(U - \mathbf{l} - D - 2\tau) - \frac{\eta^\star D + \eta^\star 2\tau}{\frac{U}{\eta^\star} - U + D}, \quad \forall y \in [L, U].$$

$$t(y) \geq \frac{\eta^\star D + \eta^\star 2\tau}{U - y - D - 2\tau} - \int_1^y \frac{\eta^\star D + \eta^\star 2\tau}{(U - u - D - 2\tau)^2} du$$

$$\geq \frac{\eta^\star D + \eta^\star 2\tau}{U - y - D - 2\tau} - \left[\frac{-\eta^\star D - \eta^\star 2\tau}{u - U + D + 2\tau} \right]_1^y$$

$$\geq \frac{\eta^\star D + \eta^\star 2\tau}{\mathbf{l} - U + D + 2\tau}.$$

By the definition of the adversaries $\mathcal{G}_y$ and $\mathcal{A}_y$, we have that $t(L) \leq 1 - s(L)$, and thus $s(L) \leq 1 - t(L)$. We combine this inequality with the above bounds to give the following condition for any $\eta^\star$-competitive online algorithm:

$$\eta^\star \ln(U - L - D - 2\tau) - \eta^\star \ln(U - \mathbf{l} - D - 2\tau) - \frac{\eta^\star D + \eta^\star 2\tau}{\frac{U}{\eta^\star} - U + D} \leq s(L) \leq 1 - t(L) \leq 1 - \frac{\eta^\star D + \eta^\star 2\tau}{\mathbf{l} - U + D + 2\tau}.$$

The optimal $\eta^\star$ is obtained when the above inequality is binding, and is given by the solution to the following transcendental equation (after substituting $\frac{U}{\eta} - 2\tau$ for $\mathbf{l}$):

$$\ln \left[\frac{U - L - D - 2\tau}{U - U/\eta^\star - D} \right] = \frac{1}{\eta^\star}. \tag{31}$$

The solution to the above is given by the following (note that $W(\cdot)$ denotes the Lambert W function):

$$\eta^\star \rightarrow \left[W \left(\frac{(D + L - U + 2\tau) \exp\left(\frac{D-U}{U}\right)}{U} \right) + \frac{U - D}{U} \right]^{-1} \tag{32}$$

□

E   Proofs for Section 4 (A Learning-augmented Algorithm)

E.1   Proof of Theorem 4.3

In this section, we prove Theorem 4.3, which states that ST-CLIP is $(1 + \varepsilon)$-consistent for any $\varepsilon \in (0, \eta - 1]$, and $O(\log n)\gamma^{(\varepsilon)}$-robust, where $\gamma^{(\varepsilon)}$ is the solution to (5).

Proof. We show the result by separately considering *consistency* (the competitive ratio when advice is correct) and *robustness* (the competitive ratio when advice is not correct) in turn.

Recall that the black-box advice Adv is denoted by a decision $\mathbf{a}_t$ at each time t. Throughout the proof, we use shorthand notation SC_t to denote the expected cost of ST-CLIP up to time t, and Adv_t to denote the cost of Adv up to time t. We start by proving Lemma 4.5 to show that ST-CLIP is $(1 + \varepsilon)$-consistent.

Proof of Lemma 4.5. First, we note that the constrained optimization enforces that the expected cost of ST-CLIP so far plus a term that forecasts the mandatory allocation is always within $(1 + \varepsilon)$ of the advice. There is always a feasible $\mathbf{p}_t$ that satisfies the constraint, because setting $\mathbf{k}_t = \Phi \mathbf{a}_t$ is always within the feasible set. Formally, if time step $j \in [T]$ denotes the time step marking the start of the mandatory allocation, the constraint in (7) holds for every time step $t \in [j]$.

Thus, to show $(1 + \varepsilon)$ consistency, we must resolve the cost of any actions during the *mandatory allocation* and show that the final expected cost of ST-CLIP is upper bounded by $(1 + \varepsilon)\text{Adv}$.

Let $\mathcal{I} \in \Omega$ be an arbitrary valid SOAD sequence. If the mandatory allocation begins at time step $j < T$, both ST-CLIP and Adv must greedily satisfy the constraint during the last m time steps $[j, T]$. This is assumed to be feasible, and the cost due to switching in and out of ON / OFF states is assumed to be negligible as long as m is sufficiently large.

Let $(1 - z^{(j-1)})$ denote the remaining deadline constraint that must be satisfied by ST-CLIP in expectation at these final m time steps, and let $(1 - A^{(j-1)})$ denote the remaining deadline constraint to be satisfied by Adv. We consider two cases, corresponding to the cases where ST-CLIP has *underprovisioned* with respect to Adv (i.e., it has completed less of the deadline constraint in expectation) and *overprovisioned* (i.e., completed more of the deadline constraint), respectively.

Case 1: ST-CLIP$(\mathcal{I})$ has "underprovisioned" $((1 - z^{(j-1)}) > (1 - A^{(j-1)}))$.

In this case, ST-CLIP must satisfy *more* of the deadline constraint (in expectation) during the mandatory allocation compared to Adv. From the previous time step, we know that the following constraint holds:

$$\text{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L + (A^{(j-1)} - z^{(j-1)})U \leq$$
$$(1 + \varepsilon) \left[\text{Adv}_{j-1} + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L \right].$$

Let $\{\mathbf{p}_t\}_{t \in [j,T]}$ and $\{\mathbf{a}_t\}_{t \in [j,T]}$ denote the decisions made by ST-CLIP and Adv during the mandatory allocation, respectively. Conditioned on the fact that ST-CLIP has completed $z^{(j-1)}$ fraction of the deadline constraint in expectation, we have that $\mathbb{E}\left[\sum_{t=j}^{T} c(\mathbf{p}_t)\right] = (1 - z^{(j-1)})$ and $\sum_{t=j}^{T} c(\mathbf{a}_t) = (1 - A^{(j-1)})$.

Considering $\{f_t(\cdot)\}_{t \in [j,T]}$, by assumption we have a lower bound based on L, namely $\sum_{t=j}^{T} f_t(\mathbf{a}_t) \geq L \sum_{t=j}^{T} c(\mathbf{a}_t)$. For the service costs that ST-CLIP must incur *over and above* what Adv incurs, we have a upper bound based on U, so $\mathbb{E}\left[\sum_{t=j}^{T} f_t(\mathbf{p}_t)\right] \leq \sum_{t=j}^{T} f_t(\mathbf{a}_t) + U(\sum_{t=j}^{T} c(\mathbf{p}_t) - \sum_{t=j}^{T} c(\mathbf{a}_t))$

Note that the worst case for ST-CLIP occurs when $\sum_{t=j}^{T} f_t(\mathbf{a}_t)$ exactly matches this lower bound, i.e., $\sum_{t=j}^{T} f_t(\mathbf{a}_t) = L \sum_{t=j}^{T} c(\mathbf{a}_t)$, as Adv is able to satisfy the rest of the deadline constraint at the best possible marginal price. Furthermore, note that if ST-CLIP and Adv are in different points of the metric at time j, the term $\mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1})$ in the left-hand-side of the constraint allows ST-CLIP to "move back" and follow Adv just before the mandatory allocation begins, thus leveraging the

same cost functions as ADV. By the constraint in the previous time step, we have the following:

$$\text{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L + (A^{(j-1)} - z^{(j-1)})U \tag{33}$$

$$\leq (1 + \varepsilon)[\text{ADV}_{j-1} + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L], \tag{34}$$

$$\text{SC}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L \sum_{t=j}^{T} c(\mathbf{a}_t) + U \left(\sum_{t=j}^{T} c(\mathbf{p}_t) - \sum_{t=j}^{T} c(\mathbf{a}_t) \right) \tag{35}$$

$$\leq (1 + \varepsilon) \left[\text{ADV}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L \sum_{t=j}^{T} c(\mathbf{a}_t) \right] \leq (1 + \varepsilon)\text{ADV}(\mathcal{I}). \tag{36}$$

$$\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right] \leq (1 + \varepsilon)\text{ADV}(\mathcal{I}). \tag{37}$$

Case 2: ST-CLIP($\mathcal{I}$) has "overprovisioned" ($(1 - z^{(j-1)}) \leq (1 - A^{(j-1)})$).

In this case, ST-CLIP must satisfy *less* of the deadline constraint (in expectation) during the mandatory allocation compared to ADV.

From the previous time step, we know that the following constraint holds:

$$\text{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + (1 - z^{(j-1)})L \leq (1 + \varepsilon) \left[\text{ADV}_{j-1} + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L \right].$$

Let $\{\mathbf{p}_t\}_{t \in [j,T]}$ and $\{\mathbf{a}_t\}_{t \in [j,T]}$ denote the decisions made by ST-CLIP and ADV during the mandatory allocation, respectively. As previously, we have that $\mathbb{E}\left[\sum_{t=j}^{T} c(\mathbf{p}_t)\right] = (1 - z^{(j-1)})$ and $\sum_{t=j}^{T} c(\mathbf{a}_t) = (1 - A^{(j-1)})$.

Considering $\{f_t(\cdot)\}_{t \in [j,T]}$, we have a lower bound on $\sum_{t=j}^{T} f_t(\cdot)$ based on L, namely $\sum_{t=j}^{T} f_t(\mathbf{p}_t) \geq L \sum_{t=j}^{T} c(\mathbf{p}_t)$ and $\sum_{t=j}^{T} f_t(\mathbf{a}_t) \geq L \sum_{t=j}^{T} c(\mathbf{a}_t)$. Since ST-CLIP has "overprovisioned", we know $\mathbb{E}\left[\sum_{t=j}^{T} c(\mathbf{p}_t)\right] \leq \sum_{t=j}^{T} c(\mathbf{a}_t)$, and thus it follows that $\mathbb{E}\left[\sum_{t=j}^{T} f_t(\mathbf{p}_t)\right] \leq \sum_{t=j}^{T} f_t(\mathbf{a}_t)$.

By the constraint in the previous time step, we have:

$$\frac{\text{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + (1 - z^{(j-1)})L}{\text{ADV}_{j-1} + \tau c(\mathbf{a}_{j-1}) + (1 - A^{(j-1)})L} =$$

$$\frac{\text{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + L \sum_{t=j}^{T} c(\mathbf{p}_t)}{\text{ADV}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L \sum_{t=j}^{T} c(\mathbf{a}_t)} \leq (1 + \varepsilon).$$

Let $y = \mathbb{E}\left[\sum_{t=j}^{T} f_t(\mathbf{p}_t)\right] - L \sum_{t=j}^{T} c(\mathbf{p}_t)$, and let $y' = \sum_{t=j}^{T} f_t(\mathbf{a}_t) - L \sum_{t=j}^{T} c(\mathbf{a}_t)$. By definition, $y \geq 0$ and $y' \geq 0$. Note that by resolving the mandatory allocation and by definition, we have that the final expected cost $\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right] \leq \text{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + L \sum_{t=j}^{T} c(\mathbf{p}_t) + y$, and $\text{ADV}(\mathcal{I}) \geq \text{ADV}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L \sum_{t=j}^{T} c(\mathbf{a}_t) + y'$.

Furthermore, since ST-CLIP has "overprovisioned" and by the linearity of the cost functions $f_t(\cdot)$, we have that $y \leq y'$. Combined with the constraint from the previous time step, we have the following bound:

$$\frac{\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right]}{\text{ADV}(\mathcal{I})} \leq \frac{\text{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + L \sum_{t=j}^{T} c(\mathbf{p}_t) + y}{\text{ADV}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L \sum_{t=j}^{T} c(\mathbf{a}_t) + y'} \tag{38}$$

$$\leq \frac{\text{SC}_{j-1} + \mathbb{W}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1}) + \tau c(\mathbf{a}_{j-1}) + L \sum_{t=j}^{T} c(\mathbf{p}_t)}{\text{ADV}_{j-1} + \tau c(\mathbf{a}_{j-1}) + L \sum_{t=j}^{T} c(\mathbf{a}_t)} \leq (1 + \varepsilon). \tag{39}$$

Thus, by combining the bounds in each of the above two cases, the result follows, and we conclude that ST-CLIP is $(1 + \varepsilon)$-consistent with accurate advice. □

Having proved consistency, we next prove Lemma 4.5 to show that ST-CLIP is $O(\log n)\gamma^{(\varepsilon)}$-robust.

Proof of Lemma 4.6. Let $\varepsilon \in (0, \eta^{\star} - 1]$ be the target consistency (recalling that ST-CLIP is $(1+\varepsilon)$-consistent), and let $\mathcal{I} \in \Omega$ denote an arbitrary valid SOAD sequence. To prove the robustness of ST-CLIP, we consider two "bad cases" for the advice $\text{ADV}(\mathcal{I})$, and show that in the worst-case, ST-CLIP's competitive ratio is bounded by $O(\log n)\gamma^{(\varepsilon)}$.

Case 1: $\text{ADV}(\mathcal{I})$ is "inactive".

Consider the case where ADV accepts nothing during the main sequence and instead satisfies the entire deadline constraint at the end of the sequence immediately before the mandatory allocation, incurring the worst possible movement & switching cost in the process. In the worst-case, this gives that $\text{ADV}(\mathcal{I}) = U + D + \tau$.

Based on the consistency constraint (and using the fact that ST-CLIP will always be "overprovisioning" with respect to ADV throughout the main sequence), we can derive an upper bound on the constraint satisfaction that ST-CLIP is "allowed to accept" from the robust pseudo-cost minimization. Recall the following constraint:

$$\text{SC}_{t-1} + f_t(\mathbf{p}_t) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + \tau c(\mathbf{a}_t) + (1 - z^{(t-1)} - c(\mathbf{p}_t))L$$
$$\leq (1 + \varepsilon) \left[\text{ADV}_t + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L \right].$$

PROPOSITION E.1. *Under "inactive" advice, z_{PCM} is an upper bound on the amount that ST-CLIP can accept from the pseudo-cost minimization in expectation without violating $(1 + \varepsilon)$-consistency, and is defined as:*

$$z_{PCM} = \gamma^{(\varepsilon)} \ln \left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau} \right]$$

PROOF. Consider an arbitrary time step t. If ST-CLIP is *not* allowed to make a decision that makes progress towards the constraint (i.e., it cannot accept anything more from the robust pseudo-cost minimization), we have that $c(\mathbf{p}_t)$ is restricted to be 0. Recall that $\mathbf{a}_t = \delta_s$ (where δ_s is the Dirac measure at the starting OFF state) for any time steps before the mandatory allocation, because the advice is assumed to be inactive. By definition, since any cost functions accepted so far in expectation (i.e., in SC_{t-1}) can be attributed to the robust pseudo-cost minimization, we have the following in the worst-case, using the same techniques used in the proof of Theorem 3.2:

$$\text{SC}_{t-1} = \int_0^{z^{(t-1)}} \psi^{(\varepsilon)}(u) du + \tau z^{(t-1)}.$$

Combining the above with the left-hand side of the consistency constraint, we have the following. Observe that $\mathbf{p}_t$ is in an OFF state, $\mathbf{a}_t = \delta_s$, and any prior movement costs to make progress towards the constraint can be absorbed into the pseudo-cost ψ since $\|\mathbf{k}_t - \mathbf{k}_{t-1}\|_{\ell_1(\mathbf{w})} \geq \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1})$. Furthermore, in the worst-case, $\mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) = Dz^{(t-1)}$ (i.e., the pseudo-cost chooses to move to a point in the metric that is a distance D away from ADV).

$$\text{SC}_{t-1} + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + (1 - z^{(t-1)})L = \int_0^{z^{(t-1)}} \psi^{(\varepsilon)}(u) du + \tau z^{(t-1)} + Dz^{(t-1)} + (1 - z^{(t-1)})L.$$

As stated, let $z^{(t-1)} = z_{\text{PCM}}$. Then by properties of the pseudo-cost,

$$
\begin{aligned}
\text{SC}_{t-1} + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + (1 - z_{\text{PCM}})L &= \int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})U + \\
&\qquad (1 - z_{\text{PCM}})L + Dz_{\text{PCM}} - (1 - z_{\text{PCM}})U, \\
&= \gamma^{(\varepsilon)} \left[\psi^{(\varepsilon)}(z_{\text{PCM}}) - D\right] + (1 - z_{\text{PCM}})L + Dz_{\text{PCM}} - (1 - z_{\text{PCM}})U, \\
&= \gamma^{(\varepsilon)}L + (L - U)\left(1 - \gamma^{(\varepsilon)}\ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right]\right) + Dz_{\text{PCM}}, \\
&= \gamma^{(\varepsilon)}L + L - U + (U - L + D)\gamma^{(\varepsilon)}\ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right].
\end{aligned}
$$

Substituting for the definition of $\gamma^{(\varepsilon)}$, we obtain:

$$
\begin{aligned}
\text{SC}_{t-1} + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + (1 - z_{\text{PCM}})L &= \gamma^{(\varepsilon)}L + L - U + (U - L + D)\gamma^{(\varepsilon)}\ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right], \\
&= \left[\varepsilon L + U - \gamma^{(\varepsilon)}(U - L + D)\ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right]\right] + \\
&\qquad L - U + (U - L + D)\gamma^{(\varepsilon)}\ln\left[\frac{U - L - D - 2\tau}{U - U/\gamma^{(\varepsilon)} - D - 2\tau}\right], \\
&= \varepsilon L + L = (1 + \varepsilon)L.
\end{aligned}
$$

This completes the proposition, since $(1 + \varepsilon)L$ is exactly the right-hand side of the consistency constraint (note that $(1 + \varepsilon)\left[\text{Adv}_t + \tau c(\mathbf{a}_t) + (1 - A_t)L\right] = (1 + \varepsilon)L$). $\qquad\square$

If ST-CLIP is constrained to use at most z_{PCM} of its utilization to be robust, the remaining $(1 - z_{\text{PCM}})$ utilization must be used for the mandatory allocation and/or to follow Adv. Note that if ST-CLIP has moved away from Adv's point in the metric, and Adv turns out to be "inactive" bad advice that incurs sub-optimal service cost late in the sequence, the consistency constraint will become non-binding and ST-CLIP will not have to move back to follow Adv in the metric. Thus, we have the following worst-case competitive ratio for ST-CLIP, specifically for Case 1, where we assume $\text{Opt}(\mathcal{I}) \to \psi^{(\varepsilon)}(z_{\text{PCM}})/O(\log n) = L/O(\log n)$, as in, e.g., Lemma 3.4:

$$
\frac{\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right]}{\text{Opt}(\mathcal{I})} \leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})U}{L/O(\log n)} \tag{40}
$$

$$
\leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})D + (1 - z_{\text{PCM}})U}{L/O(\log n)} \tag{41}
$$

By the definition of $\psi^{(\varepsilon)}(\cdot)$, we have the following:

$$
\begin{aligned}
\frac{\mathbb{E}\left[\text{ST-CLIP}(\mathcal{I})\right]}{\text{Opt}(\mathcal{I})} &\leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})D + (1 - z_{\text{PCM}})U}{L/O(\log n)} \\
&\leq \frac{\gamma^{(\varepsilon)}\left[\psi^{(\varepsilon)}(z_{\text{PCM}}) - 2D\right]}{L/O(\log n)} \leq \frac{\gamma^{(\varepsilon)}\left[L + 2D - 2D\right]}{L/O(\log n)} \leq O(\log n)\gamma^{(\varepsilon)}.
\end{aligned}
$$

Case 2: Adv($\mathcal{I}$) is "overactive".

We now consider the case where Adv incurs bad service cost due to "accepting" cost functions which it "should not" (i.e. $\text{Adv}(\mathcal{I}) \gg \text{Opt}(\mathcal{I})$). Let $\text{Adv}(\mathcal{I}) = \mathbb{V} \gg \text{Opt}(\mathcal{I})$ (i.e. the final total cost of Adv is $\mathbb{V}$ for some $\mathbb{V} \in [L, U]$, and $\mathbb{V}$ is much greater than the optimal solution).

This is without loss of generality, since we can assume that $\mathbb{V}$ is the "best marginal service and movement cost" incurred by Adv at a particular time step and the consistency ratio changes strictly in favor of Adv. Based on the consistency constraint, we can derive a lower bound on the amount that ST-CLIP *must* accept from Adv in expectation to stay $(1+\varepsilon)$-consistent. To do this, we consider the following sub-cases:

• **Sub-case 2.1**: Let $\mathbb{V} \geq \frac{U+D}{1+\varepsilon}$.

In this sub-case, ST-CLIP can fully ignore the advice, because the following consistency constraint is never binding (note that $\text{Adv}_t \geq \frac{U+D}{1+\varepsilon} A^{(t)}$):

$$SC_{t-1} + f_t(\mathbf{p}_t) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L + (A^{(t)} - z^{(t-1)} - c(\mathbf{p}_t))U$$
$$\leq (1+\varepsilon)\left[\text{Adv}_t + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L\right],$$
$$(D + \tau)c(\mathbf{a}_t) + (1 - A^{(t)})L + (A^{(t)})U \leq (1+\varepsilon)\left[\mathbb{V}c(\mathbf{a}_t) + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L\right],$$
$$(D + \tau)A^{(t)} + (1 - A^{(t)})L + UA^{(t)} \leq (1+\varepsilon)\left[\frac{U+D}{1+\varepsilon}A^{(t)} + \tau A^{(t)} + (1 - A^{(t)})L\right]$$

• **Sub-case 2.2**: Let $\mathbb{V} \in (L, \frac{U+D}{1+\varepsilon})$.

In this case, in order to remain $(1+\varepsilon)$-consistent, ST-CLIP must follow Adv and incur some "bad cost", denoted by $\mathbb{V}$. We derive a lower bound that describes the minimum amount that ST-CLIP must follow Adv in order to always satisfy the consistency constraint.

PROPOSITION E.2. *Under "overactive" advice, z_{ADV} is a lower bound on the amount that* ST-CLIP *must accept from the advice in order to always satisfy the consistency constraint, and is defined as:*

$$z_{ADV} \geq 1 - \frac{\mathbb{V}\varepsilon}{U + D - \mathbb{V}}.$$

PROOF. For the purposes of showing this lower bound, we assume there are no marginal service costs in the instance that would otherwise be accepted by the robust pseudo-cost minimization. Based on the consistency constraint, we have the following:

$$SC_{t-1} + f_t(\mathbf{p}_t) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L + (A^{(t)} - z^{(t-1)} - c(\mathbf{p}_t))U$$
$$\leq (1+\varepsilon)\left[\text{Adv}_t + \tau c(\mathbf{a}_t) + (1 - A^{(t)})L\right].$$

We let $f_t(\mathbf{p}_t) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) + \mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) + \tau c(\mathbf{a}_t) \leq vc(\mathbf{p}_t)$ for any $\mathbf{p}_t : c(\mathbf{p}_t) < c(\mathbf{a}_t)$, which holds by linearity of the cost functions $f_t(\cdot)$ and a prevailing condition that $c(\mathbf{p}_t) \leq c(\mathbf{a}_t)$ for the "bad service costs" accepted by Adv. Note that ST-CLIP must "follow" Adv to distant points in the metric to avoid violating consistency, and recall that $\mathbf{p}_t = \mathbf{a}_t$ is always in the feasible set. Under this condition that ST-CLIP follows Adv, $\mathbb{W}^1(\mathbf{p}_t, \mathbf{p}_{t-1}) + \tau c(\mathbf{a}_t)$ is upper bounded by the movement cost of Adv and absorbed into $\mathbb{V}$. The term $\mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t)$ is upper bounded by $D(A^{(t)} - c(\mathbf{p}_t))$ by Assumption 2.2 of the metric.

$$\text{SC}_{t-1} + \mathbb{V}c(\mathbf{p}_t) + L - LA^{(t)} + UA^{(t)} + DA^{(t)} - Uz^{(t-1)} - Uc(\mathbf{p}_t) - Dc(\mathbf{p}_t) \leq$$
$$(1 + \varepsilon) \left[\mathbb{V}A^{(t-1)} + \mathbb{V}c(\mathbf{a}_t) + L - LA^{(t)} \right],$$

$$\mathbb{V}c(\mathbf{p}_t) - Dc(\mathbf{p}_t) - Uc(\mathbf{p}_t) \leq (1 + \varepsilon) \left[\mathbb{V}A^{(t-1)} + \mathbb{V}c(\mathbf{a}_t) + L - LA^{(t)} \right] -$$
$$\text{SC}_{t-1} - L + LA^{(t)} - UA^{(t)} - DA^{(t)} + Uz^{(t-1)},$$

$$\mathbb{V}c(\mathbf{p}_t) - Dc(\mathbf{p}_t) - Uc(\mathbf{p}_t) \leq \mathbb{V}A^{(t)} - DA^{(t)} - UA^{(t)} - \text{SC}_{t-1} + Uz^{(t-1)} +$$
$$\varepsilon \left[\mathbb{V}A^{(t-1)} + \mathbb{V}c(\mathbf{a}_t) + L - LA^{(t)} \right],$$

$$c(\mathbf{p}_t) \geq \frac{\mathbb{V}A^{(t)} - DA^{(t)} - UA^{(t)} - \text{SC}_{t-1} + Uz^{(t-1)} + \varepsilon \left[\mathbb{V}A^{(t)} + L - LA^{(t)} \right]}{\mathbb{V} - D - U}.$$

In the event that $A^{(t-1)} = 0$ (i.e., nothing has been accepted so far by either ADV or ST-CLIP), we have:

$$c(\mathbf{p}_t) \geq \frac{\mathbb{V}c(\mathbf{a}_t) - Dc(\mathbf{a}_t) - Uc(\mathbf{a}_t) + \varepsilon \left[\mathbb{V}c(\mathbf{a}_t) + L - Lc(\mathbf{a}_t) \right]}{\mathbb{V} - D - U},$$

$$c(\mathbf{p}_t) \geq c(\mathbf{a}_t) - \frac{\varepsilon \left[\mathbb{V}c(\mathbf{a}_t) + L - Lc(\mathbf{a}_t) \right]}{U + D - \mathbb{V}}.$$

Through a recursive definition, we can show that for any $A^{(t)}$, given that ST-CLIP has satisfied $z^{(t-1)}$ of the deadline constraint by following ADV so far, it must set $\mathbf{p}_t$ such that:

$$z^{(t)} \geq z^{(t-1)} + c(\mathbf{a}_t) - \frac{\varepsilon \left[\mathbb{V}c(\mathbf{a}_t) + L - Lc(\mathbf{a}_t) \right]}{U + D - \mathbb{V}}.$$

Continuing the assumption that $\mathbb{V}$ is constant, if ST-CLIP has accepted $z^{(t-1)}$ thus far, we have the following if we assume that all of the constraint satisfaction up to this point happened in a single previous time step m:

$$c(\mathbf{p}_t) \geq A^{(t)} - \frac{Uc(\mathbf{p}_m) + Dc(\mathbf{p}_m) - \text{SC}_{t-1} + \varepsilon \left[\mathbb{V}A^{(t)} + L - LA^{(t)} \right]}{U + D - \mathbb{V}},$$

$$c(\mathbf{p}_t) \geq c(\mathbf{a}_t) + c(\mathbf{a}_m) - c(\mathbf{p}_m) - \frac{\varepsilon \left[\mathbb{V}(c(\mathbf{a}_t) + c(\mathbf{a}_m)) + L - L(c(\mathbf{a}_t) + c(\mathbf{a}_m)) \right]}{U + D - \mathbb{V}},$$

$$c(\mathbf{p}_t) + c(\mathbf{p}_m) \geq c(\mathbf{a}_t) + c(\mathbf{a}_m) - \frac{\varepsilon \left[\mathbb{V}(c(\mathbf{a}_t) + c(\mathbf{a}_m)) + L - L(c(\mathbf{a}_t) + c(\mathbf{a}_m)) \right]}{U + D - \mathbb{V}},$$

$$z^{(t)} \geq A^{(t)} - \frac{\varepsilon \left[\mathbb{V}A^{(t)} + L - LA^{(t)} \right]}{U + D - \mathbb{V}}.$$

This gives intuition into the desired z_{ADV} bound. The above motivates that the *aggregate* expected constraint satisfaction by ST-CLIP at any given time step t must satisfy a lower bound. Consider that the worst case for Sub-case 2.2 occurs when all of the v prices accepted by ADV arrive first, before any prices that would be considered by the pseudo-cost minimization. Then let $A^{(t)} = 1$ for some arbitrary time step t, and we have the stated lower bound on z_{ADV}. □

If ST-CLIP is forced to use z_{ADV} of its utilization to be $(1 + \varepsilon)$ consistent against ADV, that leaves at most $(1 - z_{\text{ADV}})$ utilization for robustness. We define $z' = \min(1 - z_{\text{ADV}}, z_{\text{PCM}})$ and consider the following two cases.

• **Sub-case 2.2.1**: if $z' = z_{\text{PCM}}$, the worst-case competitive ratio is bounded by the following. Note that if $z' = z_{\text{PCM}}$, the amount of utilization that ST-CLIP can use to "be robust" is exactly the same

as in **Case 1**, and we again have that $\text{OPT}(I) \to \psi^{(\varepsilon)}(z_{\text{PCM}})/O(\log n) = L/O(\log n)$:

$$\frac{\mathbb{E}\left[\text{ST-CLIP}(I)\right]}{\text{OPT}(I)} \leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{Adv}} - z_{\text{PCM}})U + z_{\text{Adv}}\mathbb{V}}{L/O(\log n)}, \tag{42}$$

$$\leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})D + (1 - z_{\text{PCM}})U}{L/O(\log n)}, \tag{43}$$

$$\leq O(\log n)\gamma^{(\varepsilon)}. \tag{44}$$

• **Sub-case 2.2.2**: if $z' = 1 - z_{\text{Adv}}$, the worst-case competitive ratio is bounded by the following. Note that ST-CLIP *cannot* use z_{PCM} of its utilization for robustness, so the following bound assumes that the "robust service costs" accepted by ST-CLIP are bounded by the *worst* $(1 - z_{ADV})$ *fraction* of the pseudo-cost function $\psi^{(\varepsilon)}$ (note that $\psi^{(\varepsilon)}$ is non-increasing on $z \in [0, 1]$):

$$\frac{\mathbb{E}\left[\text{ST-CLIP}(I)\right]}{\text{OPT}(I)} \leq \frac{\int_0^{1-z_{\text{Adv}}} \psi^{(\varepsilon)}(u)du + \tau(1 - z_{\text{Adv}}) + z_{\text{Adv}}\mathbb{V}}{L/O(\log n)}.$$

Note that if $z' = 1 - z_{\text{Adv}}$, we know that $1 - z_{\text{Adv}} < z_{\text{PCM}}$, which further gives the following by definition of z_{Adv}:

$$1 - z_{\text{PCM}} < 1 - \frac{\mathbb{V}\varepsilon}{U + D - \mathbb{V}},$$

$$\mathbb{V}\varepsilon < (U + D - \mathbb{V})z_{\text{PCM}},$$

$$\mathbb{V} < \frac{U + D}{(1 + \frac{\varepsilon}{z_{\text{PCM}}})}.$$

By plugging $\mathbb{V}$ back into the definition of z_{Adv}, we have that $z_{\text{Adv}}\mathbb{V} \leq \left(\frac{(1-z_{\text{PCM}})(U+D)}{1+\frac{\varepsilon}{z_{\text{PCM}}}}\right)$, giving the following:

$$\frac{\mathbb{E}\left[\text{ST-CLIP}(I)\right]}{\text{OPT}(I)} \leq \frac{\int_0^{1-z_{\text{Adv}}} \psi^{(\varepsilon)}(u)du + \tau(1 - z_{\text{Adv}}) + \left(\frac{(1-z_{\text{PCM}})(U+D)}{1+\frac{\varepsilon}{z_{\text{PCM}}}}\right)}{L/O(\log n)}, \tag{45}$$

$$\leq \frac{\int_0^{z_{\text{PCM}}} \psi^{(\varepsilon)}(u)du + \tau z_{\text{PCM}} + (1 - z_{\text{PCM}})D + (1 - z_{\text{PCM}})U}{L/O(\log n)}, \tag{46}$$

$$\leq O(\log n)\gamma^{(\varepsilon)}. \tag{47}$$

Thus, by combining the bounds in each of the above two cases, the result follows, and we conclude that ST-CLIP is $O(\log n)\gamma^{(\varepsilon)}$-robust. □

Having proven Lemma 4.5 (consistency) and Lemma 4.6 (robustness), the statement of Theorem 4.3 follows: ST-CLIP is $(1+\varepsilon)$-consistent and $O(\log n)\gamma^{(\varepsilon)}$-robust given any advice for SOAD. □

E.2 Proof of Theorem 4.4

In this section, we prove Theorem 4.4, which states that $\gamma^{(\varepsilon)}$ (as defined in (5)) is the optimal robustness for any $(1 + \varepsilon)$-consistent learning-augmented SOAD algorithm.

PROOF. To show this result, we build off the same y-adversaries for SOAD defined in Definition D.3, where $y \in [L, U]$. For the purposes of showing consistency, we define a slightly tweaked adversary $\mathcal{A}'_y$:

DEFINITION E.3 ($\mathcal{A}'_y$ ADVERSARY FOR LEARNING-AUGMENTED SOAD). *Recall the $\mathcal{A}_y$ adversary defined in Definition D.3. During **Stage 1** of the adversary's sequence, $\mathcal{A}_y$ and $\mathcal{A}'_y$ are identical. In*

Stage 2, $\mathcal{A}'_y$ presents $Up(x)$ at the starting point's ON state $\text{ON}^{(s)}$ once, followed by $Down^{m_y}(x) = yx$ at $\text{ON}^{(s)}$. All other ON states are considered inactive in this stage, so they only receive $Up(x)$. In the $\mu - 1$ time steps after $Down^{m_y}(x) = yx$ is presented, the adversary presents $Down^{m_y}(x) = yx$ at the starting point in the metric (allowing Opt and Adv to reduce their switching cost). Finally, the adversary presents $Up(x)$ everywhere for the final μ time steps, and **Stage 2** ends.

As in the proof of Theorem 3.3, for adversary $\mathcal{A}'_y$, the optimal offline objective is $\text{Opt}(\mathcal{A}'_y) \to y$. Against these adversaries, we consider two types of advice – the first is *bad* advice, which stays with their full allocation at the starting OFF state (i.e., $\mathbf{a}_t = \delta_s$) for all time steps $t < j$ before the mandatory allocation, incurring a final cost of $U + 2\tau$.

On the other hand, *good* advice sets $\mathbf{a}_t = \delta_s$ for all time steps up to the first time step when y is revealed at the starting point in the metric, after which it sets $a_t^{\text{ON}^{(s)}} = 1/\mu$ to achieve final cost $\text{Adv}(\mathcal{A}'_y) = \text{Opt}(\mathcal{A}'_y) = y + 2\tau/\mu$.

We let $(s(y) + t(y))$ denote a *robust constraint satisfaction function* $[L, U] \to [0, 1]$, that fully quantifies the actions of a learning-augmented algorithm Alg playing against adaptive adversary $\mathcal{A}'_y$, where $(s(y) + t(y))$ gives the progress towards the deadline constraint under the instance $\mathcal{A}'_y$ before (either) the mandatory allocation or the black-box advice sets $a_t^{\text{ON}^{(s)}} > 0$. Since the conversion is unidirectional (irrevocable), we must have that $s(y - \sigma) + t(y - \sigma) \geq (s(y) + t(y))$, i.e. $(s(y) + t(y))$ is non-increasing in $[L, U]$.

As in the proof of Theorem 3.3, the adaptive nature of $\mathcal{A}'_y$ forces any algorithm to incur a movement and switching cost proportional to $(s(y) + t(y))$ during the robust phase, specifically denoted by $(D + 2\tau)(s(y) + t(y))$. Recall that by the proof of Theorem 3.3, for any γ-competitive online algorithm Alg, we have the following condition on $(s(y) + t(y))$:

$$(s(y) + t(y)) \geq \gamma \ln (U - y - D - 2\tau) - \gamma \ln (U - \mathbf{l} - D - 2\tau), \quad \forall y \in [L, U]. \tag{48}$$

Furthermore, we have that the expected cost of Alg on adversary $\mathcal{A}'_y$ is given by:

$$\mathbb{E}\left[\text{Alg}(\mathcal{A}'_y)\right] = s(\mathbf{l})\mathbf{l} - \int_1^y u ds(u) + Ds(y) + t(\mathbf{l})\mathbf{l} - \int_1^y u dt(u) + Dt(y) + \tag{49}$$

$$(1 - s(y) - t(y))U + 2\tau [s(y) + t(y)] \tag{50}$$

To simultaneously be α-consistent when the advice *is* correct, Alg must satisfy $\mathbb{E}\left[\text{Alg}(\mathcal{A}'_L)\right] \leq \alpha \text{Opt}(\mathcal{A}'_L) = \alpha L$. If the advice is correct, Alg must pay an additional factor of D to move back and follow Adv in the worst case – but can satisfy the rest of the deadline constraint at the best cost functions L. It must also still pay for switching incurred by the robust algorithm (recall that Opt pays no switching cost). Using integral by parts, we have:

$$\int_1^L s(u) + t(u) du + [2D + 2\tau](s(L) + t(L)) + (1 - s(y) - t(y))L + L(s(y) + t(y)) \leq \alpha L, \tag{51}$$

$$\int_1^L s(u) + t(u) du + [2D + 2\tau](s(L) + t(L)) \leq \alpha L - L. \tag{52}$$

By combining equations (48) and (52), and substituting $\mathbf{l} = U/\gamma$, the robust constraint satisfaction function $(s(y) + t(y))$ of any γ-robust and α-consistent online algorithm must satisfy:

$$\gamma \int_1^L \ln \left(\frac{U - u - D - 2\tau}{U - U/\gamma - D - 2\tau}\right) du + [2D + 2\tau]\left[\gamma \ln \left(\frac{U - L - D - 2\tau}{U - U/\gamma - D - 2\tau}\right)\right] \leq \alpha L - L. \tag{53}$$

When all inequalities are binding, this equivalently gives that

$$\alpha \geq \gamma + 1 - \frac{U}{L} + \frac{\gamma(U - L + D)}{L} \ln \left(\frac{U - L - D - 2\tau}{U - U/\gamma - D - 2\tau} \right). \tag{54}$$

We define α such that $\alpha \coloneqq (1 + \varepsilon)$. By substituting for α into (54), we recover the definition of $\gamma^{(\varepsilon)}$ as given by (5), which subsequently completes the proof. Thus, we conclude that any $(1 + \varepsilon)$-consistent algorithm for SOAD is at least $\gamma^{(\varepsilon)}$-robust. □

F   Proofs for Section 5 (Generalization to Time-varying Metrics)

F.1   Proof of Corollary 5.1

In this section, we prove Corollary 5.1, which states that PCM is $O(\log n)\eta$-competitive for SOAD-T, the variant of SOAD where distances in the metric (X, d) are allowed to be time-varying (see Section 5).

We recall a few mild assumptions on the SOAD-T problem that directly imply the result. Let $d_t(\cdot, \ \cdot)$ denote the distance between points in X at time $t \in [T]$. We redefine D as $D = \sup_{t \in [T]} \left(\max_{u,v \in X : u \neq v} \frac{d_t(u, \ v)}{\min\{c^{(u)}, c^{(v)}\}} \right)$, i.e., it is an upper bound on distance between any two points in the metric at any time over the horizon T. Recall that the temporal switching cost between ON and OFF states at a single point is defined as *non-time-varying*, so $\| \cdot \|_{\ell_1(\beta)}$ gives the temporal switching cost for all $t \in [T]$.

We also assume that the tree embedding-based vector space $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ defined by Definition 2.3 is appropriately reconstructed at each step, and that PCM has knowledge of the current distances (i.e., $\| \cdot \|_{\ell_1(\mathbf{w})}$ accurately reflects $d_t(\cdot, \ \cdot)$ at time t).

Under these assumptions, every step in the proof of Theorem 3.2 holds. In particular, note that in Lemma 3.4, the only fact about the distance function that is used is the fact that the distance between two ON states is upper bounded by D, in (11), and that the vector space $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ has expected $O(\log n)$ distortion with respect to the underlying metric, which follows by definition. In Lemma 3.5, most of PCM's movement cost is absorbed into the integral over the pseudo-cost function ψ, and the only other fact about the distance that is used is that the distance between ON and OFF states at a single point $u \in X$ is fixed and bounded by $\tau c^{(u)}$, which follows by definition. Thus, we conclude that PCM is $O(\log n)\eta$-competitive for SOAD-T. □

F.2   Proof of Corollary 5.2

In this section, we prove Corollary 5.2, which states that a minor change to the consistency constraint enables ST-CLIP to be $(1 + \varepsilon)$-consistent and $O(\log n)\gamma^{(\varepsilon)}$-robust for SOAD-T, given any $\varepsilon \in (0, \eta - 1]$.

We start by recalling assumptions on the SOAD-T problem that inform the result. Recall that we redefine D as $D = \sup_{t \in [T]} \left(\max_{u,v \in X : u \neq v} \frac{d_t(u, \ v)}{\min\{c^{(u)}, c^{(v)}\}} \right)$, i.e., it is an upper bound on distance between any two points at any time over the horizon T, and the temporal switching cost between ON and OFF states at a single point is defined as *non-time-varying*, so $\| \cdot \|_{\ell_1(\beta)}$ gives the temporal switching cost for all $t \in [T]$.

We also assume that the tree embedding-based vector space $(K, \| \cdot \|_{\ell_1(\mathbf{w})})$ defined by Definition 2.3 is appropriately reconstructed at each step, and that ST-CLIP has knowledge of the current distances (i.e., $\| \cdot \|_{\ell_1(\mathbf{w})}$ and $\mathbb{W}^1(\cdot, \ \cdot)$ accurately reflect $d_t(\cdot, \ \cdot)$ at time t).

For the consistency constraint, we define a modified Wasserstein-1 distance function $\overline{\mathbb{W}}^1$ that will be used in the consistency constraint to hedge against the time-varying properties of the metric. This distance computes the optimal transport between two distributions on Δ_S while

assuming that the underlying distances are given by $\overline{d}(\cdot, \ \cdot)$, which is itself defined such that $\overline{d}(u, \ v) = D \min\{c^{(u)}, c^{(v)}\} : u, v \in X : u \neq v$.

$$\overline{\mathbb{W}}^1(\mathbf{p}, \mathbf{p}') \coloneqq \min_{\pi \in \Pi(\mathbf{p}, \mathbf{p}')} \mathbb{E}\left[\overline{d}(\mathbf{x}, \mathbf{x}')\right], \tag{55}$$

where $(\mathbf{x}, \mathbf{x}') \sim \pi_t$ and $\Pi(\mathbf{p}, \mathbf{p}')$ is the set of distributions over X^2 with marginals $\mathbf{p}$ and $\mathbf{p}'$.

Intuitively, the purpose of $\overline{\mathbb{W}}^1$ is to leverage the D upper bound between points in the metric to give a "worst-case optimal transport" distance between distributions, assuming that the time-varying distances increase in future time steps.

To this end, within the definition of the consistency constraint (7), ST-CLIP for SOAD-T replaces the term $\mathbb{W}^1(\mathbf{p}, \mathbf{a}_t)$ with $\overline{\mathbb{W}}^1(\mathbf{p}, \mathbf{a}_t)$ – this term hedges against the case where ST-CLIP must move to follow ADV in a future time step, and in the time-varying distances case, we charge ST-CLIP an extra amount to further hedge against the case where the underlying distances between ST-CLIP and ADV grow in future time steps.

Paralleling the proof of Theorem 4.3, we consider consistency and robustness independently.

Consistency.   For consistency, according to the proof of Lemma 4.5, we show that resolving the mandatory allocation remains feasible in the time-varying case. First, note that there is always a feasible $\mathbf{p}_t$ that satisfies the consistency constraint, since even in the time-varying case, setting $\mathbf{k}_t = \Phi\mathbf{a}_t$ is always within the feasible set – this follows by observing that at a given time t, if ST-CLIP has moved away from ADV, it has already "prepaid" a worst-case movement cost of $\overline{\mathbb{W}}^1(\mathbf{p}_m, \mathbf{a}_m)$ (for some previous time step m) in order to move back and follow ADV. The remainder of the proof of Lemma 4.5 only uses the fact that at the beginning of the mandatory allocation (at time $j \in [T]$), $\overline{\mathbb{W}}^1(\mathbf{p}_{j-1}, \mathbf{a}_{j-1})$ is an upper bound on the movement cost paid by ST-CLIP (if necessary) while migrating to ADV's points in the metric to take advantage of the same service cost functions. By definition of $\overline{\mathbb{W}}^1$, this follows for any time-varying distance $d_t(\cdot, \ \cdot)$, and the remaining steps in the proof hold.   □

Robustness.   For robustness, following the proof of Lemma 4.6, we show that a certain amount of ST-CLIP's utilization can be "set aside" for robustness. First, note that in Case 1 (i.e., "inactive" advice), the proof of Proposition E.1 only requires that $\mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t)$ is bounded by D, which follows immediately by the definition of $\overline{\mathbb{W}}^1$ – the remaining steps for Case 1 follow. In Case 2 (i.e., "overactive" advice), note that the proof of Sub-case 2.1 similarly only requires that $\mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t)$ is bounded by D, which follows because $\mathbb{W}^1(\mathbf{p}_t, \mathbf{a}_t) \leq \overline{\mathbb{W}}^1(\mathbf{p}_t, \mathbf{a}_t) \leq D \max\{c(\mathbf{p}_t), c(\mathbf{a}_t)\}$. Likewise, the remaining steps for Case 2 follow.   □

Thus, we conclude that ST-CLIP is $(1 + \varepsilon)$-consistent and $O(\log n)\gamma^{(\varepsilon)}$-robust for SOAD-T, given any $\varepsilon \in (0, \eta - 1]$.