Optimal Hidden-Target Learning for Online Inventory Optimization on General Convex Sets

Anthony Pineci
UIUC
apineci2@illinois.edu

Yunzong Xu
UIUC
xyz@illinois.edu

Abstract

Online inventory optimization (OIO) is online convex optimization with physical memory: inventory can be raised immediately, but it can be reduced only by demand. A natural principle, used in stochastic inventory learning and recently in OIO under a single linear capacity constraint, is to maintain a hidden target chosen by an online learner and implement its projection onto the currently feasible order-up-to set. We prove that this simple principle is optimal for OIO on arbitrary bounded convex capacity sets. With online gradient descent as the base learner, the method improves the best known regret guarantee for OIO on general convex sets from inverse to inverse-square-root dependence on the common-demand probability, and we prove a matching lower bound. The analysis identifies the right geometric state variable: the Euclidean distance between the hidden target and the implementable set. This distance evolves pathwise as a scalar queue, with target movement as arrivals and common demand as service, reducing the state-dependent feasibility cost to the control of a one-dimensional queue. The same reduction gives the first polylogarithmic regret guarantee for strongly convex losses and the first dynamic regret guarantee adapting to Euclidean path variation on general convex capacity sets.

1 Introduction

Inventory control is a fundamental problem in operations research, stochastic control, and online learning. A retailer, warehouse, or fulfillment system repeatedly chooses order-up-to levels while demand, costs, and constraints evolve. This resembles an online convex optimization (OCO) problem, where at each round the learner picks an action, observes local cost information, and has its performance measured against a good comparator. The key difference is that inventory carries over: stock can be replenished up to a chosen level, but only disappears when demand consumes it. A gradient step may point a learner toward a lower stock level, but the items already on the shelf make that target physically impossible.

Online inventory optimization (OIO), introduced by [1], isolates this obstruction in an adversarial online-learning model. The learner chooses an order-up-to vector in a capacity set $\mathcal{Y} \subseteq \mathbb{R}^n_+$, but the action must also dominate the current inventory state. If the inventory state is always zero, OIO reduces to standard OCO on $\mathcal{Y}$ [2, 3, 4, 5]. Otherwise, past actions restrict the feasible set at future rounds. This is the learning-theoretic form of the carryover effects studied in classical periodic-review inventory models [6] and in data-driven inventory learning with censored demand [7, 8, 9, 10].

The first general-convex OIO algorithm, MaxCOSD [1], handles this state dependence through adaptive cycles: it waits for moments when a proposed update to the implemented order level is compatible with the inventory state. This idea is robust enough to work for arbitrary bounded convex capacity sets and adversarial losses. Under uniformly probably positive demand (UPPD), a non-degeneracy condition on the common demand (i.e., minimum demand over all products) with

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Reference / Method	Capacity Set	Demand Primitive	Regret type	Bound
[1] / MaxCOSD	general	UPPD μ	static	$\widetilde{O}(\sqrt{T}/\mu)$
[11] / HT	linear	sell-out period $L_{\max}$	static	$\widetilde{O}(n^{1/4}\sqrt{L_{\max}T})$
			dynamic	$\widetilde{O}(n^{1/4}\sqrt{L_{\max}(1+P_{T,1})T})$
[Ours] / HT	general	UPPD μ	static (cvx)	$\widetilde{O}(\sqrt{T/\mu})$ / $\Omega(\sqrt{T/\mu})$
			static (s.c.)	$O(\log(T/\delta)\log(T)/\mu)$
			dynamic	$\widetilde{O}(\sqrt{(1+P_{T,2})T/\mu})$

Table 1: Closest adversarial-demand OIO guarantees, suppressing problem-dependent constants. HT abbreviates hidden target. For [11], the $n^{1/4}$ term appears explicitly in intermediate bounds and is absorbed into the big-O notation of their final stated regret. Here $P_{T,1}$ and $P_{T,2}$ denote comparator path variation in ℓ_1 and ℓ_2, respectively. For *[Ours]*, the static convex row lists the upper bound and matching lower bound; the strongly convex (s.c.) and dynamic bounds appear on separate rows. Appendix D.2 relates UPPD and $L_{\max}$.

probability parameter μ, its high-probability static regret is $\widetilde{O}(\sqrt{T}/\mu)$. The $1/\mu$ factor controls how long the system may wait before all products receive enough demand to reduce excess stock, and in large or intermittent-demand systems this waiting time can dominate the guarantee.

A different algorithmic principle has long been attractive in inventory learning. Rather than modifying the implemented level directly, a *base learner* can keep a desired stock level in the background, and implement it only as far as the current state permits. This *hidden-target* idea appears in queueing analyses for single-product learning [7], in multiproduct algorithms under a single linear warehouse constraint [8], and most explicitly in the recent hidden-target OIO method of [11]. The principle also has practical appeal: [1] suggests using it to improve MaxCOSD's empirical performance in large multiproduct systems with capacity constraints. However, extending existing hidden-target analyses to general convex capacity sets faces a geometric barrier. These analyses rely on a single linear warehouse constraint and productwise sell-out structure, but Appendix C shows why such productwise control does not extend uniformly to general convex capacity sets. The reliance on productwise control also leads to additional polynomial dependence on the number of products n, which is undesirable from the norm-based viewpoint standard in OCO.

Our contributions. This paper shows that the simple and natural hidden-target projection method is already powerful enough for the full general-convex OIO model. The missing ingredient is not a more complicated algorithm, but a better geometric state variable. At each round, the base learner produces hidden targets $z_t \in \mathcal{Y}$. Given the current implementable set J_t, the algorithm implements $y_t = \mathrm{Proj}_{J_t}(z_t)$. We track the Euclidean target-implementation gap $q_t = \mathrm{dist}(z_t, J_t) = \|z_t - y_t\|_2$. Letting $d_t^- := \min_i d_{t,i}$ denote common demand, the inventory dynamics imply the pathwise recursion

$$q_{t+1} \le \left[q_t + \|z_{t+1} - z_t\|_2 - d_t^-\right]^+.$$

Thus the target-implementation gap evolves as a one-dimensional queue. Moving the learning target creates arrivals, while common demand provides service. Queueing intuition is familiar in inventory learning [7, 8], but here the queue is geometric: it is not an itemwise sell-out clock, but the distance from the hidden target to the moving convex feasible set. We provide distribution-free control on this one-dimensional process, establishing a powerful reduction theorem that converts a base learner's regret into sharp guarantees for hidden-target learning in general settings.

The consequences are summarized in Table 1. With online gradient descent (OGD) as the base learner, our static regret result improves the μ dependence of the best known general-convex OIO guarantee from μ^{-1} to $\mu^{-1/2}$ and establishes optimality via a new lower bound. Even under linear capacity constraints, our Euclidean analysis removes the explicit $n^{1/4}$ factor in the hidden-target analysis of [11]. Under strongly convex losses, the same algorithm obtains polylogarithmic regret, addressing an open direction raised by [1]. In dynamic environments, the bound depends on the Euclidean path variation $P_{T,2}$ rather than the larger ℓ_1 variation $P_{T,1}$, sharpening [11] on more general geometry. We instantiate this with OGD when $P_{T,2}$ is known a priori and smoothed OGD (SOGD) [12] when it is not; both achieve the same rate up to logarithmic factors. All our guarantees are *dimension-free* in the

standard norm-based sense: dimension enters through problem parameters such as the capacity set diameter and μ, but not through an additional explicit factor of dimension n. While the parameter μ can still deteriorate with dimension, our result isolates this difficulty in the demand primitive instead of adding a separate algorithmic dimension penalty.

Notation and Organization. We denote $\mathbb{N}$ as the set of positive integers and write $[m] := \{1, \dots, m\}$ for any $m \in \mathbb{N}$, with $[m] = \varnothing$ for integers $m \leq 0$. We use $a \succeq b$ for coordinatewise inequality and Euclidean norms throughout. Probabilistic statements are with respect to a filtration $(\mathcal{F}_t)_{t \geq 0}$. The learner's action at round t is $\mathcal{F}_{t-1}$-measurable, whereas the demand and loss chosen after the action are $\mathcal{F}_t$-measurable. We write $O(\cdot), \Omega(\cdot)$, and $\Theta(\cdot)$ for standard asymptotic upper, lower, and tight bounds respectively. We additionally write $\widetilde{O}(\cdot)$ to hide polylogarithmic factors in T. All proofs are deferred to the appendix.

2 Model and hidden-target projection

2.1 The OIO model

We work in the model of [1], generalized to adaptive adversarial settings for the environment's choice of demand and loss function. The environment first sets the initial inventory state to zero, i.e. $x_1 = \mathbf{0}$, then for every round $t = 1, \dots, T$:

1. The learner observes an inventory state $x_t \in \mathbb{R}^n$ and must choose an order level $y_t \in \mathcal{Y}$ that obeys the *inventory feasibility constraint*:
$$y_t \succeq x_t \tag{1}$$
The inventory is instantly replenished to y_t.
2. The environment observes the learner's action and chooses a demand $d_t \in \mathbb{R}^n_+$, then a loss function $\ell_t : \mathcal{Y} \to \mathbb{R}$. The next inventory state x_{t+1} is updated so that it adheres to the *inventory dynamics constraint*:
$$x_{t+1} \preceq (y_t - d_t)^+ \tag{2}$$
3. The learner incurs a loss $\ell_t(y_t)$ and observes a subgradient $g_t \in \partial \ell_t(y_t)$ that may help them make a choice in the next round.

We use the following convexity and boundedness assumptions, as in [1]:

Assumption 2.1 (Convex and bounded OIO problem)**.** The capacity set $\mathcal{Y} \subseteq \mathbb{R}^n_+$ is nonempty, closed, convex, and bounded, with Euclidean diameter $D := \sup_{u,v \in \mathcal{Y}} \|u - v\|_2$. Each loss $\ell_t : \mathcal{Y} \to \mathbb{R}$ is convex, and every observed subgradient $g_t \in \partial \ell_t(y_t)$ satisfies $\|g_t\|_2 \leq G$.

We assume $D, G > 0$ when specifying stepsizes; when $D = 0$ or $G = 0$, regret is nonpositive.

We allow a general convex capacity set $\mathcal{Y}$ [1], rather than restricting attention to the single linear warehouse constraint common in inventory learning [8, 11]. Appendix D.1 gives motivating examples involving multiple shared-resource constraints and chance constraints that yield curved capacity sets.

Equality in (2) gives the usual lost-sales dynamics. The inequality allows for alternative dynamics incorporating additional depletion or perishability, as in [1]. If the state never binds, the only constraint is $y_t \in \mathcal{Y}$ and the model is ordinary OCO on $\mathcal{Y}$. The difficulty is precisely that the feasible action set at time t is shaped by earlier order-up-to decisions.

Since the demand is chosen prior to the loss function, this framework accommodates the standard newsvendor loss used in the inventory literature:

$$\ell_t(y; d_t) = \sum_{i=1}^{n} \Big(h_{t,i}(y_i - d_{t,i})^+ + p_{t,i}(d_{t,i} - y_i)^+ \Big). \tag{3}$$

where $h_t, p_t \in \mathbb{R}^n_+$ correspond to the holding cost (overage cost) and lost-sales penalty (underage cost) respectively. Following the OIO framework [1], we do not permit the loss function ℓ_t to depend on the current inventory x_t. However, under a lost-sales dynamic, newsvendor loss (3) can naturally accommodate purchasing costs using a standard cost transformation [6, Section 4.3.2.4].

The framework's requirement for subgradient observability does not require full demand observation. A realistic assumption is that the learner may observe only sales $s_t = \min\{y_t, d_t\}$ taken coordinatewise. In this case, the vector with coordinates given by

$$h_{t,i}\mathbb{1}\{y_{t,i} > s_{t,i}\} - p_{t,i}\mathbb{1}\{y_{t,i} = s_{t,i}\}$$

is an observable subgradient of (3) at y_t. This shows that our subgradient requirement is sufficient for the canonical inventory loss, while the analysis permits arbitrary convex losses with local subgradient feedback.

We measure algorithm performance using *Regret*. For a comparator sequence $u_{1:T} \in \mathcal{Y}^T$, dynamic regret is

$$R_T(u_{1:T}) := \sum_{t=1}^{T} \big(\ell_t(y_t) - \ell_t(u_t)\big).$$

Static regret is the special case $u_t \equiv u$.

The implementable set at round t is $J_t := \mathcal{Y} \cap \{y \in \mathbb{R}^n \mid y \succeq x_t\}$. It is always nonempty: $J_1 = \mathcal{Y}$ since $x_1 = \mathbf{0}$, and (2) implies $y_t \in J_{t+1}$ whenever $y_t \in J_t$.

2.2 Why OIO is harder than OCO

In OCO, the learner can choose any point in the fixed feasible set $\mathcal{Y}$ at each round. In OIO, inventory carried over from earlier rounds restricts which points in $\mathcal{Y}$ can actually be implemented. The following counterexample shows that applying OGD [2, 4] directly to the currently implementable set can incur linear expected regret.

Take one product with $\mathcal{Y} = [0, M]$, $M > 0$, lost-sales dynamics, and fixed newsvendor coefficients $h, p > 0$. Let OGD update the order-up-to level by $y_{t+1} = \mathrm{Proj}_{J_{t+1}}(y_t - \eta_t g_t)$. Demands are i.i.d.: $d_t = \rho > M$ with probability $\mu \in (0, 1)$ and $d_t = 0$ otherwise. On zero-demand rounds, the update with $g_t = h$ is blocked by the inventory constraint. High-demand rounds clear the inventory, but the update with $g_t = -p$ points upward. Hence

$$y_{t+1} = \begin{cases} \min\{M, y_t + \eta_t p\}, & d_t = \rho, \\ y_t, & d_t = 0. \end{cases}$$

The order-up-to level never decreases and reaches M almost surely for either $\eta_t \equiv \eta$ or $\eta_t = c/\sqrt{t}$, with fixed $\eta, c > 0$. Yet if $\mu < h/(h + p)$, the expected loss is increasing on $[0, M]$, so the best fixed order-up-to level is 0 and direct OGD incurs linear expected regret.

In this example, direct projection discards the blocked downward updates. To retain this information, we maintain a desired stock level separately from the implemented order-up-to level. This desired level is the *hidden target*, which continues to be updated even when it cannot yet be implemented.

2.3 Hidden-target projection and the regret decomposition

We work in the standard projection-oracle model of OCO, assuming Euclidean projections onto $\mathcal{Y}$ and onto J_t can be computed. Appendix D.3 discusses projection tractability and the additional regret incurred when projections are computed approximately.

Hidden-target projection lets an online learner move in the fixed set $\mathcal{Y}$ while a separate inventory layer enforces physical feasibility. When the base learner proposes $z_t \in \mathcal{Y}$, the implemented order-up-to level is

$$y_t = \mathrm{Proj}_{J_t}(z_t). \tag{4}$$

This is the algorithmic principle behind projected base-stock methods such as DDM [8] and the hidden-target OIO method of [11]. After implementing y_t, we observe only a subgradient $g_t \in \partial \ell_t(y_t)$ at the implemented point. The linear loss $f_t(z) = \langle g_t, z \rangle$ is therefore the most informative function we can construct from this feedback, and so we take the base learner to be an online linear optimizer (OLO). The implementation is formalized in Algorithm 1.

Algorithm 1: Hidden-Target Projection Meta-Algorithm

Input: Capacity set $\mathcal{Y}$, base learner (OLO) on $\mathcal{Y}$, initial target $z_1 \in J_1$
for $t = 1, 2, \ldots, T$ **do**
  Observe inventory state x_t and set $J_t = \mathcal{Y} \cap \{y \in \mathbb{R}^n \mid y \succeq x_t\}$;
  Implement $y_t = \mathrm{Proj}_{J_t}(z_t)$;
  Observe $g_t \in \partial \ell_t(y_t)$ and the next state x_{t+1};
  Feed the linearized loss $f_t(z) = \langle g_t, z \rangle$ to the base learner and receive $z_{t+1} \in \mathcal{Y}$;
end

The regret analysis begins with a decomposition inherited from earlier hidden-target arguments. Define

$$\Delta_T^{\text{base}}(u_{1:T}) := \sum_{t=1}^{T} \langle g_t, z_t - u_t \rangle, \qquad \Delta_T^{\text{gap}} := \sum_{t=1}^{T} \|z_t - y_t\|_2, \qquad \Delta_T^{\text{mv}} := \sum_{t=1}^{T-1} \|z_{t+1} - z_t\|_2.$$

The first quantity is the ordinary OCO regret of the base learner. The second is the cost of physical implementation of the target. The third tracks target movement, which will become a switching cost.

Lemma 2.2 (Target regret plus implementation error)**.** *Let $(z_t)_{t=1}^T \in \mathcal{Y}^T$ be any hidden-target sequence, and let $(y_t)_{t=1}^T$ be defined by* (4)*. Then for every comparator sequence $u_{1:T} \in \mathcal{Y}^T$,*

$$R_T(u_{1:T}) \le \Delta_T^{\textit{base}}(u_{1:T}) + G\Delta_T^{\textit{gap}}. \tag{5}$$

For linear warehouse constraints, [8] and [11] bound the implementation term through itemwise sell-out quantities. That is exactly where the general-convex problem becomes delicate. While the residual $z_t - y_t$ still measures the gap between the target and its implementation, on an arbitrary convex set it no longer admits an itemwise shortage interpretation due to coordinate coupling during the projection. Section 3 shows that its Euclidean length is nevertheless governed by a scalar recursion.

3 The target-implementation queue

The inventory-specific part of our analysis is pathwise; neither stationarity nor independence is required. We use UPPD only to obtain a high-probability clearing bound. Let

$$q_t := \mathrm{dist}(z_t, J_t) = \|z_t - y_t\|_2, \qquad d_t^- := \min_{i \in [n]} d_{t,i}. \tag{6}$$

Proposition 3.1 (Target-implementation queue)**.** *Under the inventory dynamics constraint* (2)*, for every $t \in [T-1]$,*

$$q_{t+1} \le \left[q_t + \|z_{t+1} - z_t\|_2 - d_t^-\right]^+. \tag{7}$$

The proof is geometric. If z_{t+1} lies within distance d_t^- of the implemented point y_t, common demand makes z_{t+1} feasible at the next round and the queue clears. Otherwise, move from y_t toward z_{t+1} by exactly distance d_t^-. That interpolation point is feasible after demand, and Euclidean projection can only bring z_{t+1} closer. This point's feasibility is provided by the following lemma.

Lemma 3.2 (Short moves and inventory feasibility)**.** *Let $y, y' \in \mathbb{R}_+^n$ and $d \in \mathbb{R}_+^n$. If $\|y' - y\|_2 \le d^- := \min_i d_i$, then $y' \succeq (y - d)^+$.*

To obtain high-probability regret, we use the non-degeneracy demand condition introduced by [1].

Assumption 3.3 (Uniformly probably positive demand)**.** There exist $\mu \in (0, 1]$ and $\rho > 0$ such that, for every round t, almost surely

$$\mathbb{P}(d_t^- \ge \rho \mid \mathcal{F}_{t-1}) \ge \mu.$$

UPPD says that, no matter the past, there is conditional probability at least μ of a common demand shock of size ρ. So over any sufficiently long window of rounds, at least one such shock occurs with high probability. To exploit this condition for queueing control, our analysis partitions $[T]$ into queue-clearing windows where the shocks can clear the accumulated queue, reducing the cumulative gap to a bound on the target's movement. Appendix D.2 discusses the realism of this assumption and alternative window clearing conditions in more detail. If the base learner moves by at most ρ over each such window, the queue cannot persist across windows. This is made precise in the following theorem:

Theorem 3.4 (Regret reduction under UPPD)**.** *Assume UPPD and set* $B_\delta := \lceil \mu^{-1} \log(T/\delta) \rceil$ *for a fixed* $\delta \in (0,1)$*. Suppose* $q_1 = 0$ *and, for every* t *with* $1 \le t \le T - B_\delta$*,*

$$\sum_{r=t}^{t+B_\delta-1} \|z_{r+1} - z_r\|_2 \le \rho. \tag{8}$$

Then, with probability at least $1-\delta$*,*

$$\Delta_T^{gap} \le B_\delta \Delta_T^{mv}, \qquad R_T(u_{1:T}) \le \Delta_T^{base}(u_{1:T}) + G B_\delta \Delta_T^{mv}, \tag{9}$$

for every comparator sequence $u_{1:T} \in \mathcal{Y}^T$*.*

Theorem 3.4 is a reduction from OIO to OCO with switching cost. The base learner must control two ordinary quantities, linearized regret and cumulative movement, while the inventory layer contributes only the clearing-window factor B_δ. This modularity is the advantage of the Euclidean queue: the inventory proof is agnostic to how z_t was generated, and the base learner never reasons about products or sell-out events.

Relation to sell-out reductions. The linear-capacity hidden-target analysis of [11] tracks productwise sell-out quantities and controls switching in ℓ_1. In contrast, our reduction uses the state variable $\mathrm{dist}(z_t, J_t)$, which makes it applicable to arbitrary convex feasible sets $\mathcal{Y}$. This change also eliminates the explicit $n^{1/4}$ factor arising from the ℓ_1 movement control in [11], and replaces $P_{T,1}$ in dynamic regret by the smaller Euclidean path variation $P_{T,2}$. The tradeoff is the local movement condition (8) on the base learner, which is not required in [11]. This movement condition can be enforced through the base learner's tuning. The OGD stepsizes and SOGD switching parameter in Section 4 satisfy it and yield the stated regret bounds.

4 Optimal and adaptive guarantees

We now instantiate the reduction. Hidden-Target OGD uses ordinary projected OGD on the hidden targets and lets the inventory layer implement the projection. Its update is

$$y_t = \mathrm{Proj}_{J_t}(z_t), \qquad z_{t+1} = \mathrm{Proj}_{\mathcal{Y}}(z_t - \eta_t g_t). \tag{10}$$

Algorithm 2: Hidden-Target OGD

Input: Capacity set $\mathcal{Y}$, initial target $z_1 \in J_1$, stepsizes (η_t)
for $t = 1, 2, \ldots, T$ **do**
  Observe inventory state x_t and set $J_t = \mathcal{Y} \cap \{y \in \mathbb{R}^n \mid y \succeq x_t\}$;
  Implement $y_t = \mathrm{Proj}_{J_t}(z_t)$;
  Observe $g_t \in \partial \ell_t(y_t)$ and the next state x_{t+1};
  Update $z_{t+1} = \mathrm{Proj}_{\mathcal{Y}}(z_t - \eta_t g_t)$;
end

4.1 Convex losses and the optimal dependence on μ

Theorem 4.1 (Static regret for convex losses)**.** *Assume Assumption 2.1, the dynamics constraint* (2)*, and UPPD. Fix* $\delta \in (0,1)$ *and set* $B_\delta = \lceil \mu^{-1} \log(T/\delta) \rceil$*. Run HT-OGD (Algorithm 2) from* $z_1 \in J_1$ *with stepsize* $\eta_t = \gamma D/(G\sqrt{t})$*, where* $0 < \gamma \le \rho/(2D\sqrt{B_\delta})$*. Then, with probability at least* $1-\delta$*, simultaneously for all* $u \in \mathcal{Y}$*,*

$$R_T(u) \le DG\Big(\frac{1}{2\gamma} + \gamma + 2\gamma B_\delta\Big)\sqrt{T}.$$

Choosing γ *as a sufficiently small constant multiple of* $B_\delta^{-1/2}$ *gives*

$$R_T(u) = O\big(DG\sqrt{B_\delta T}\big) = \widetilde{O}\left(DG\sqrt{\frac{T}{\mu}}\right).$$

The hidden constants in these asymptotic bounds may depend on D/ρ. Moreover, for $T \geq 2$, taking $\delta = 1/T$ and $\gamma = \rho/(2D\sqrt{B_{1/T}})$ gives

$$\mathbb{E}[R_T(u)] \leq DG\Big(\frac{D}{\rho} + \frac{3\rho}{2D}\Big)\sqrt{B_{1/T}T} + DG.$$

The theorem balances two costs. OGD contributes $O(DG\sqrt{T}/\gamma)$ target regret, while the queue reduction charges $O(DG\gamma B_\delta\sqrt{T})$ for target movement. The clearing-scale choice $\gamma = \Theta(B_\delta^{-1/2})$ gives the optimal $\widetilde{O}(DG\sqrt{T/\mu})$ rate.[1]

Theorem 4.2 (Lower bound under UPPD)**.** *There exists a universal constant $c > 0$ such that, for every $D, G > 0$, $\mu \in (0, 1]$, $\rho > 0$, and $T \geq \mu^{-1}$, there is a two-product OIO instance with capacity set*

$$\mathcal{Y}_D := \Big\{ y \in \mathbb{R}^2_+ \;\Big|\; y_1 + y_2 = \tfrac{D}{\sqrt{2}} \Big\} \tag{11}$$

of diameter D, gradient bound G, and UPPD parameters (μ, ρ), for which every possibly randomized inventory algorithm satisfies

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(y_t) - \min_{u \in \mathcal{Y}_D} \sum_{t=1}^{T} \ell_t(u)\right] \geq c\, DG\sqrt{\frac{T}{\mu}}.$$

The instance uses only the two linear losses $\ell_t(y) = Gy_1$ and $\ell_t(y) = Gy_2$.

Proof idea. The construction uses Bernoulli common-demand shocks. After a no-demand round, inventory feasibility and the equality constraint on $y_1 + y_2$ pin the next order-up-to level to the previous one, so the learner plays the same action between any two shocks. This decomposes the period into random blocks of constant action. Each block then receives an independent random loss direction. The learner cannot exploit the direction within the block, while the best static comparator chooses the endpoint favored by the aggregate signed block length. Since the sum of squared block lengths is typically of order T/μ, this aggregate has an imbalance of order $\sqrt{T/\mu}$.

Theorem 4.2 holds for every $\rho > 0$. When $\rho \geq D/\sqrt{2}$, the stepsize condition of Theorem 4.1 allows $\gamma = \Theta(B_\delta^{-1/2})$ with a universal constant, giving $O(DG\sqrt{B_\delta T}) = \widetilde{O}(DG\sqrt{T/\mu})$ regret. The upper and lower bounds therefore match in D, G, μ, and T, up to logarithmic factors. Determining the sharp dependence on D/ρ for smaller ρ remains open.

This lower bound is different from the $L_{\max}$-based lower bound of [11]. Their construction uses deterministic cycles with long zero-demand stretches, so it is not a UPPD lower bound with a positive μ. Our instance itself satisfies UPPD with parameter μ, and the randomness of the clearing times is exactly what produces the $\sqrt{1/\mu}$ factor. Together, Theorems 4.1 and 4.2 show that the $\mu^{-1/2}$ dependence is minimax optimal under UPPD.

4.2 Strongly convex losses

The same algorithm gives a fast rate when the losses have curvature. The base learner uses the standard strongly-convex OGD stepsize, which gives logarithmic cumulative target movement.

Theorem 4.3 (Static regret for strongly convex losses)**.** *Assume Assumption 2.1, the dynamics constraint* (2)*, and UPPD. Suppose each ℓ_t is α-strongly convex on $\mathcal{Y}$ for some $\alpha > 0$, and let $T \geq 2$. Fix $\delta \in (0, 1)$ and define*

$$B_\delta := \Big\lceil \frac{1}{\mu} \log\Big(\frac{T}{\delta}\Big) \Big\rceil, \qquad s_\delta := \max\left\{1, \left\lceil \frac{B_\delta}{e^{\alpha\rho/(2G)} - 1} \right\rceil \right\}.$$

Run HT-OGD (Algorithm 2) from $z_1 \in J_1$ with $\eta_t = 2/(\alpha(t + s_\delta))$. Then, with probability at least $1 - \delta$, for all $u \in \mathcal{Y}$,

$$R_T(u) = O\left(\frac{\log(T/\delta)\log T}{\mu}\right),$$

where the hidden constant depends only on α, ρ, D, and G.

[1] The tuning above uses μ and ρ, but their exact values are unnecessary. Known positive lower bounds $\bar{\mu} \leq \mu$ and $\bar{\rho} \leq \rho$ also satisfy UPPD, and the regret bounds then apply with these conservative values. Appendix D.2 discusses parameter selection and parameter-free guarantees.

We retain the negative quadratic term from strong convexity when applying the OGD inequality. The terms introduced by the changing stepsize then cancel; after bounding the target–implementation gap, we obtain $O(B_\delta \log T)$ regret. This gives the first polylogarithmic-in-T adversarial-OIO guarantee for strongly convex losses on general convex capacity sets, resolving a major open direction raised by [1]. The result also complements the logarithmic rates known under more stochastic inventory assumptions [10].

4.3 Dynamic regret

For a comparator sequence $u_{1:T}$, define

$$P_{T,1} := \sum_{t=2}^{T} \|u_t - u_{t-1}\|_1, \qquad P_{T,2} := \sum_{t=2}^{T} \|u_t - u_{t-1}\|_2. \tag{12}$$

We measure comparator variation in the Euclidean norm, matching the target-movement term in the reduction.

Theorem 4.4 (Dynamic regret with known path variation)**.** *Assume Assumption 2.1, the dynamics constraint* (2)*, and UPPD. Fix $\delta \in (0,1)$ and set $B_\delta := \lceil \mu^{-1} \log(T/\delta) \rceil$. For a comparator sequence $u_{1:T} \in \mathcal{Y}^T$ with known path variation $P_{T,2}$, let*

$$\eta^* := \frac{1}{G}\sqrt{\frac{D(D + 2P_{T,2})}{(2B_\delta + 1)T}}.$$

If $\eta^* \le \rho/(B_\delta G)$ and HT-OGD (Algorithm 2) is run with constant stepsize $\eta_t \equiv \eta^*$, then with probability at least $1-\delta$,

$$R_T(u_{1:T}) \le G\sqrt{(2B_\delta + 1)D(D + 2P_{T,2})T} = \widetilde{O}\left(G\sqrt{\frac{D(D + P_{T,2})T}{\mu}}\right).$$

Note that the best known dynamic regret guarantee for OIO was achieved by [11] using the same hidden-target principle, with path variation measured by ℓ_1 rather than ℓ_2, an additional $n^{1/4}$ factor, and the restriction on a linear capacity constraint. For dense comparator movement, replacing $P_{T,1}$ by $P_{T,2}$ can improve the path-variation term by a factor of $n^{1/4}$; together with removal of the explicit $n^{1/4}$ factor, the total improvement can be as large as $\sqrt{n}$.

When $P_{T,2}$ is unknown, we use the switching-aware SOGD base learner of [12], following the hidden-target architecture of [11]. SOGD combines a grid of OGD experts with a smoothed meta-learner and satisfies a regret-plus-switching-cost guarantee. Appendix F.4.2 gives the base learner details and proves that its movement is slow enough for the queue reduction.

For $\lambda \ge 1$, define

$$\begin{aligned}\mathcal{R}_T^{\mathrm{SOGD}}(u_{1:T};\lambda) := {} & 2GD\sqrt{(1+\lambda)T\Big(1 + \frac{2P_{T,2}}{D}\Big)} \\ & + 120GD\sqrt{\lambda T\Big(1 + \frac{2P_{T,2}}{D}\Big)\log T}.\end{aligned} \tag{13}$$

Theorem 4.5 (Dynamic regret with unknown path variation)**.** *Assume Assumption 2.1, the dynamics constraint* (2)*, and UPPD. Fix $\delta \in (0,1)$ and set*

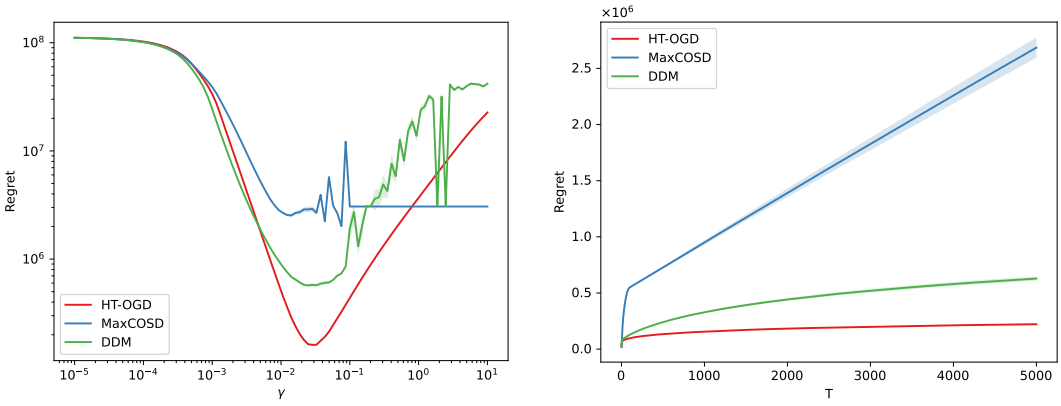
$$B_\delta := \Big\lceil \frac{1}{\mu}\log\Big(\frac{T}{\delta}\Big)\Big\rceil, \qquad \lambda_\delta := \Big\lceil B_\delta \max\Big\{1, \frac{2D}{\rho}\Big\}\Big\rceil.$$

Assume $T \ge \max\{32\lambda_\delta \log T, e\}$. Run the hidden-target projection layer with the SOGD base learner and switching parameter λ_δ. Then, with probability at least $1-\delta$, for every comparator sequence $u_{1:T} \in \mathcal{Y}^T$,

$$R_T(u_{1:T}) \le \mathcal{R}_T^{\mathrm{SOGD}}(u_{1:T};\lambda_\delta). \tag{14}$$

In particular,

$$R_T(u_{1:T}) = O\left(GD\sqrt{\Big(1 + \frac{P_{T,2}}{D}\Big)T\max\Big\{1, \frac{D}{\rho}\Big\}\frac{1}{\mu}\log\Big(\frac{T}{\delta}\Big)\log T}\right).$$



(a) Learning-rate sweep.

(b) Regret over time.

Figure 1: Static regret on Setting 3 of [1]. Panel (a) reports regret at $T = 5 \times 10^3$ over logarithmically spaced learning-rate multipliers. Panel (b) uses the best multiplier for each algorithm and plots regret over time. Envelopes show ± 2 standard errors.

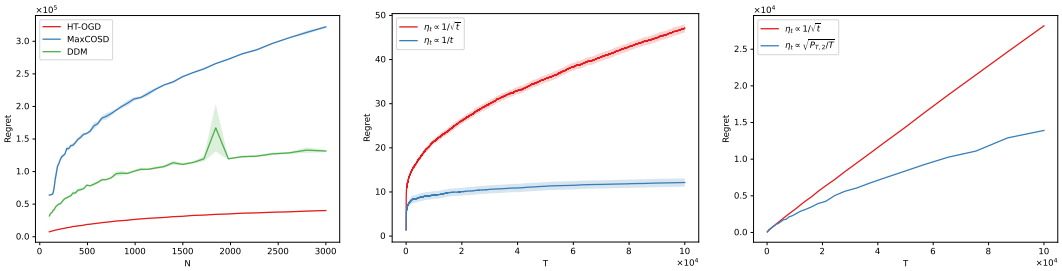
5 Experiments

The experiments compare the three algorithmic principles available in the literature. MaxCOSD waits for a new proposed target to become feasible before committing to it [1]. DDM is a projected base-stock method designed for stochastic multiproduct systems with a linear warehouse constraint that only updates its target when unbiased gradient estimates become available [8]. HT-OGD uses the simple every-round hidden-target projection principle. Since DDM was designed for linear warehouse constraints, the common testing environment is $\mathcal{Y}_C = \{y \in \mathbb{R}^n_+ : \|y\|_1 \le C\}$ with lost-sales dynamics and newsvendor loss. While this setup does not display the full general-convex scope of our theorems, it is the setting where all three methods can be run side by side. The closest hidden-target OIO experiments in [11] are single-item, whereas our algorithmic comparisons are multiproduct. In experiments where we compare different stepsizes for HT-OGD suggested by our theoretical results in Section 4, we use a single product setting to establish a baseline for these new experimental settings. In this case, the linear feasible set $\mathcal{Y}_C$ reduces to an interval and we choose sufficiently large C to be non-binding.

Unless stated otherwise, $h_i = 1$ and $p_i = 200$ in newsvendor loss. We tune a scalar learning-rate multiplier γ for each method, compute static comparators offline by optimizing over the realized demand sequence, and plot ± 2 standard-error envelopes over repeated runs. Appendix B gives the full parameter choices.

Figure 1 reproduces Setting 3 of [1]: $n = 100$ products, independent Poisson demands with rates drawn from $\mathrm{Uniform}[1, 2]$, and capacity $C = 175$. HT-OGD attains lower regret at its best tuning and is less sensitive to the learning-rate multiplier. In the time-series panel, MaxCOSD accumulates regret much faster on the plotted horizon, DDM behaves sublinearly, and HT-OGD remains the most stable. This stochastic instance is not meant to be a worst-case construction, it is a common benchmark showing that the hidden-target projection principle also behaves competitively in the capacitated multiproduct regime where cycle methods were originally tested.

The remaining experiments probe different aspects of our theoretical results. Figure 2a uses a power-law demand experiment to test explicit dimension dependence while keeping aggregate demand and loss scale comparable. While regret may still grow based on other problem parameters like D and G, the plot demonstrates the dimensional-scaling advantage our analysis predicts. The strongly convex loss experiment in Figure 2b distinguishes the performance of HT-OGD under the $1/t$ schedule from the general convex $1/\sqrt{t}$ schedule. This separation is consistent with our polylogarithmic regret bound for the $1/t$ schedule and $\widetilde{O}(\sqrt{T})$ bound for the $1/\sqrt{t}$ schedule. Lastly, the dynamic regret experiment in Figure 2c shows the advantage of using a known-variation constant stepsize for HT-OGD when the comparator moves with $P_{T,2} = O(\sqrt{T})$. The plot shows that this adapted stepsize is consistent with the $\widetilde{O}(\sqrt{P_{T,2}T}) = \widetilde{O}(T^{3/4})$ dynamic regret predicted by Theorem 4.4.



(a) Dimension scaling. (b) Strong convexity. (c) Dynamic regret.

Figure 2: Additional experiments. (a) Static regret as the number of products grows under power-law demand $d_{t,i} \propto i^{-1.1}$. (b) Regret performance for quadratic strongly convex loss, comparing $\eta_t \propto 1/\sqrt{t}$ and $\eta_t \propto 1/t$. (c) Dynamic regret for blockwise-constant demand with $P_{T,2} = O(\sqrt{T})$, comparing the static schedule with the known-variation constant stepsize.

6 Conclusion

We prove that hidden-target learning can achieve the optimal regret for OIO on general convex capacity sets under UPPD. In our experiments, this simple method also exhibits significant empirical advantages.

The full version of this paper [13] extends our main reduction to mirror-descent base learners through a norm alignment principle: the same norm must be used for projection, the target–implementation gap, and target movement. This broadens the use of mirror descent in multiproduct inventory learning from stochastic demand and a single linear warehouse constraint [14] to adversarial demand and general convex capacity sets.

UPPD requires demand of at least ρ for every product in the same round with conditional probability at least μ. With many products and nearly independent intermittent demand, this requirement can make μ very small. It remains open whether sublinear regret on general convex capacity sets can be obtained under a weaker assumption that requires demand only on products where the implemented action exceeds the hidden target.

Acknowledgments and Disclosure of Funding

The authors formulated the research questions, established the theoretical results, and designed the experiments. ChatGPT and Claude were used for editorial assistance, proof simplification, and coding. The authors take full responsibility for the paper's content.

Anthony Pineci was supported by the Amazon AI PhD Fellowship and the National Science Foundation Graduate Research Fellowship Program (NSF GRFP). This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 21-46756. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Massil Hihat, Stéphane Gaïffas, Guillaume Garrigos, and Simon Bussy. Online inventory problems: Beyond the i.i.d. setting with online convex optimization. In *Advances in Neural Information Processing Systems*, volume 36, pages 20421–20440, 2023.

[2] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, pages 928–936, 2003.

[3] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[4] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3–4):157–325, 2016.

[5] Francesco Orabona. Online learning: A modern introduction using convex optimization. *arXiv preprint arXiv:1912.13213*, 2019.

[6] Lawrence V. Snyder and Zuo-Jun Max Shen. Stochastic Inventory Models: Periodic Review. In *Fundamentals of Supply Chain Theory*, pages 87–154. John Wiley & Sons, Ltd, second edition, 2019.

[7] Woonghee Tim Huh and Paat Rusmevichientong. A Nonparametric Asymptotic Analysis of Inventory Planning with Censored Demand. *Mathematics of Operations Research*, 34(1): 103–123, 2009.

[8] Cong Shi, Weidong Chen, and Izak Duenyas. Technical note—nonparametric data-driven algorithms for multiproduct inventory systems with censored demand. *Operations Research*, 64 (2):362–370, 2016.

[9] Huanan Zhang, Xiuli Chao, and Cong Shi. Technical Note—Perishable Inventory Systems: Convexity Results for Base-Stock Policies and Learning Algorithms Under Censored Demand. *Operations Research*, 66(5):1276–1286, 2018.

[10] Jiameng Lyu, Jinxing Xie, Shilin Yuan, and Yuan Zhou. A Minibatch Stochastic Gradient Descent-Based Learning Metapolicy for Inventory Systems with Myopic Optimal Policy. *Management Science*, 71(7):5572–5588, 2025.

[11] Koji Ichikawa, Kei Takemura, and Tatsuya Matsuoka. Online inventory optimization in non-stationary environment. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

[13] Anthony Pineci and Yunzong Xu. Optimal hidden-target learning for online inventory optimization on general convex sets. *arXiv preprint arXiv:2606.14679*, 2026. URL `https://arxiv.org/abs/2606.14679`. Full version.

[14] Sichen Guo, Cong Shi, Chaolin Yang, and Christos Zacharias. An online mirror descent learning algorithm for multiproduct inventory systems. *Operations Research*, 74(5):2310–2328, 2026.

[15] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2–3):169–192, 2007.

[16] Eric C. Hall and Rebecca M. Willett. Dynamical models and tracking regret in online convex programming. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 579–587, 2013.

[17] Ali Jadbabaie, Alexander Rakhlin, Shahin Shahrampour, and Karthik Sridharan. Online optimization: Competing with dynamic comparators. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 398–406, 2015.

[18] Dhruv Madeka, Kari Torkkola, Carson Eisenach, Anna Luo, Dean P. Foster, and Sham M. Kakade. Deep inventory management. *arXiv preprint arXiv:2210.03137*, 2022.

[19] Matias Alvo, Daniel Russo, Yash Kanoria, and Minuk Lee. Deep reinforcement learning for inventory networks: Toward reliable policy optimization. *arXiv preprint arXiv:2306.11246*, 2025.

[20] Alvaro Maggiar, Sohrab Andaz, Akhil Bagaria, Carson Eisenach, Dean Foster, Omer Gottesman, and Dominique Perrault-Joncas. Structure-informed deep reinforcement learning for inventory management. *arXiv preprint arXiv:2507.22040*, 2025.

[21] Michael Bloem, Song Zhou, Kai He, Zhunyou Hua, and Yan Xia. Simulation-based online retailer supply chain inbound node arrival capacity control. In *Proceedings of the 2025 Winter Simulation Conference*, pages 438–449. IEEE, 2025.

[22] Luis F. Cardona and Kevin R. Gue. How to determine slot sizes in a unit-load warehouse. *IISE Transactions*, 51(4):355–367, 2019.

[23] Vineet Goyal and R. Ravi. A PTAS for the chance-constrained knapsack problem with random item sizes. *Operations Research Letters*, 38(3):161–164, 2010.

[24] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, 1999.

Appendix Contents

A Additional related work **14**

B Detailed experiment setup **14**

C Why productwise dynamics fail on general convex sets **15**

D Discussion of assumptions **16**

D.1 Examples of general capacity sets 16

D.2 UPPD, sell-out windows, and tuning 17

D.3 Projection computation and approximation errors 18

E Proofs for Sections 2 and 3 **19**

E.1 Deterministic clearing and pathwise transfer 19

E.2 Proof of Lemma 2.2 19

E.3 Proof of Proposition 3.1 19

E.4 Proof of Theorem E.2 20

E.5 Proof of Lemma E.4 21

E.6 Proof of Theorem 3.4 21

F Proofs for Section 4 **22**

F.1 General convex losses 22

F.2 A matching lower bound under UPPD 23

F.3 Strongly convex losses 24

F.4 Dynamic regret 26

A Additional related work

Online convex optimization. Projected online gradient descent gives the classical $O(\sqrt{T})$ regret guarantee for Lipschitz convex losses [2, 3, 4, 5], and logarithmic regret under curvature appears in the strongly convex and exp-concave OCO literature [15]. Dynamic regret compares against a changing comparator sequence and is controlled by path variation [16, 17]. Our OGD and strongly-convex analyses are standard OCO arguments; the contribution is to show that the inventory feasibility layer can be reduced to a switching-cost term. For unknown path variation, we use the smoothed OCO meta-learner of [12].

Stochastic inventory learning. The infeasible-target problem appeared before OIO. AIM [7] used a queueing argument for single-product censored-demand learning. Later, [8] developed DDM for multiproduct systems with a linear warehouse-capacity constraint. [9] used cycle subgradients and a bridging construction for perishable inventory. [10] proposed a minibatch SGD metapolicy for several stochastic inventory systems, including strongly convex settings. These methods exploit i.i.d. demand and problem-specific inventory structure. Our setting permits adversarial demand and arbitrary bounded convex capacity sets, with the probabilistic assumption entering only through UPPD clearing.

Online inventory optimization. To generalize previous settings, [1] introduced OIO to move beyond i.i.d. demand, fixed dynamics, and newsvendor-specific losses. Their MaxCOSD algorithm works for general convex capacity sets but controls learning through cycle boundaries, leading to a μ^{-1} weak-demand dependence under UPPD. [11] analyzed hidden-target projection and obtained dynamic regret guarantees in nonstationary environments, but their proof uses linear capacity and productwise sell-out quantities. The present paper keeps the hidden-target principle and replaces the sell-out analysis by a Euclidean queue, giving optimal weak-demand dependence on general convex sets.

Learning-based inventory control. Deep reinforcement learning and simulation-based methods have recently become effective for large inventory systems with rich dynamics [18, 19, 20, 21]. These methods emphasize empirical performance, differentiable simulation, network structure, or supply-chain-scale policies. They are complementary to the present work, whose goal is a sharp adversarial regret theory for simple projection-based principles.

B Detailed experiment setup

All experiments use the linear warehouse set $\mathcal{Y}_C = \{y \in \mathbb{R}^n_+ : \|y\|_1 \le C\}$ so that HT-OGD, MaxCOSD, and DDM can be compared on the same feasible region. The inventory dynamics are lost sales, $x_{t+1} = (y_t - d_t)^+$, and the basic loss is the newsvendor loss (3). A scalar multiplier γ rescales the nominal learning rate in each method. For the convex static experiments, HT-OGD and DDM use $\eta_t \propto 1/\sqrt{t}$, while MaxCOSD uses the adaptive stepsize prescribed by [1]. Static regret is computed against an offline fixed comparator obtained by projected OGD on the realized demand sequence until convergence. When applicable, tests are rerun to reduce variance from stochastic demand generation and we display ± 2 standard error bars in the plots.

All experimental results were computed on an Apple Mac Studio using an M4 Max chip with 48GB of memory. Each experiment finished execution in under an hour.

General convex setting. For Figure 1, we use Setting 3 of [1]: $n = 100$, independent demands $d_{t,i} \sim \mathrm{Poisson}(\lambda_i)$, independent rates $\lambda_i \sim \mathrm{Uniform}[1, 2]$, and capacity $C = 175$. The learning-rate sweep uses horizon $T = 5 \times 10^3$ and logarithmically spaced multipliers $\gamma \in [10^{-5}, 10]$, with 200 reruns. The time-series panel plots regret up to the same horizon, using the best multiplier for each algorithm from the sweep.

Dimension-scaling. Demands are computed as $d_{t,i} = i^{-\alpha} u_{t,i}$ with $\alpha = 1.1$ and independent $u_{t,i} \sim \mathrm{Uniform}[0, 1]$. Since $\alpha > 1$, the total demand magnitude remains comparable as n grows. This experiment is a stress test for algorithmic dimension scaling rather than a direct verification of UPPD, because common clearing can become rare in sparse high-dimensional systems. At all

dimensions, we use a capacity of $C = 2.75$. We evaluate logarithmically spaced product counts $n \in [100, 3000]$ with 10 reruns.

Strong convexity. For the strongly convex experiment, we use a single-product quadratic loss $\ell_t(y) = \frac{1}{2}(y - d_t)^2$ with i.i.d. Poisson(λ) demand with $\lambda = 1.5$. The capacity $C = 20$ is chosen to be sufficiently large to be non-binding. We compare HT-OGD with the general convex schedule $\eta_t \propto 1/\sqrt{t}$ and the strongly convex schedule $\eta_t \propto 1/t$, tuning the scalar multiplier on a shorter horizon of $T = 10^4$ before evaluating at the plotted horizon up to $T = 10^5$ over 20 reruns.

Dynamic regret. In the dynamic-regret panel, the single-product demand sequence consists of constant blocks, with block length $\lceil \sqrt{T} \rceil$ and demand values cycling through the list $\{2, 3, 4\}$. Under lost-sales, the comparator $u_t = d_t$ is optimal since it is feasible and has zero newsvendor loss, yielding a path variation of $P_{T,2} = O(\sqrt{T})$. We compare to the static-regret schedule $\eta_t \propto 1/\sqrt{t}$ with the constant stepsize suggested by Theorem 4.4. Note that since the constant stepsize depends on T, we rerun the algorithm at many different values of T rather than plotting the cumulative regret from one run with large T. The underage cost is set to $p = 2$ to avoid a degenerate myopic ordering rule. Since this is a deterministic demand sequence and the algorithm is also deterministic, there is no variance in the results so we do not display error bars.

C Why productwise dynamics fail on general convex sets

Earlier hidden-target analyses track excess inventory in a single product [7] or aggregate mismatch under a single warehouse constraint [8, 11]. To see the difficulty in extending such arguments to general capacity sets, we examine how the inventory lower bound changes a hidden target. A small violation in one product can force a much larger change in another product to preserve capacity feasibility. We show that this displacement cannot be bounded uniformly by the amount inventory exceeds the target.

Let $\hat{y} \in \mathcal{Y}$ be a hidden target and $x \in \mathcal{Y}$ an inventory state. The vector $(x - \hat{y})^+$ records the coordinatewise violation of the inventory lower bound. The implemented action is the projection of $\hat{y}$ onto $\mathcal{Y} \cap \{y \succeq x\}$, which is nonempty because it contains x.

Proposition C.1 (No uniform coordinatewise bound)**.** *Already for $n = 2$, there is no finite constant C such that, for every closed, convex, bounded, downward-closed set $\mathcal{Y} \subseteq \mathbb{R}^n_+$ and every $\hat{y}, x \in \mathcal{Y}$, the projection*

$$y^\star = \mathrm{Proj}_{\mathcal{Y} \cap \{y \succeq x\}}(\hat{y})$$

satisfies

$$|y^\star_i - \hat{y}_i| \le C(x_i - \hat{y}_i)^+ \qquad \textit{for all } i \in [n].$$

Proof. Take the closed, bounded, convex, downward-closed set

$$\mathcal{Y} = \{(u, v) \in \mathbb{R}^2_+ \mid u + v^2 \le 1\}.$$

For $\delta \in (0, 1]$, choose

$$\hat{y}^{(\delta)} = (1 - \delta^2, \delta), \qquad x^{(\delta)} = (1, 0).$$

Both points belong to $\mathcal{Y}$, and $\mathcal{Y} \cap \{y \succeq x^{(\delta)}\} = \{(1, 0)\}$. Thus $y^\star = (1, 0)$ and

$$|y^\star_2 - \hat{y}^{(\delta)}_2| = \delta, \qquad (x^{(\delta)}_2 - \hat{y}^{(\delta)}_2)^+ = 0.$$

The capacity constraint forces the second coordinate to change even though it has no inventory lower-bound violation. □

Summing over products does not give a bound independent of the capacity geometry either.

Proposition C.2 (No uniform aggregate mismatch bound)**.** *Already for $n = 2$, there is no finite constant C such that, for every closed, convex, bounded, downward-closed set $\mathcal{Y} \subseteq \mathbb{R}^n_+$ and every $\hat{y}, x \in \mathcal{Y}$, the projection*

$$y^\star = \mathrm{Proj}_{\mathcal{Y} \cap \{y \succeq x\}}(\hat{y})$$

satisfies

$$\|y^\star - \hat{y}\|_1 \le C \|(x - \hat{y})^+\|_1.$$

Proof. For the same set, target, and state,

$$\|y^\star - \hat{y}^{(\delta)}\|_1 = \delta + \delta^2, \qquad \|(x^{(\delta)} - \hat{y}^{(\delta)})^+\|_1 = \delta^2.$$

Their ratio is $1 + \delta^{-1}$, which diverges as $\delta \downarrow 0$. □

The required increase in the first coordinate is only δ^2, but maintaining capacity feasibility forces a decrease of δ in the second coordinate. Thus the total coordinatewise violation can be arbitrarily smaller than the target–implementation gap.

The same failure already occurs with two linear resource constraints. Fix $\delta \in (0, 1/2]$ and consider

$$\mathcal{Y}_\delta := \{(u, v) \in \mathbb{R}^2_+ \mid u + \delta v \le 1,\ \delta u + v \le 1\}.$$

Each product uses one unit of one resource and δ units of the other. This downward-closed polytope contains both $\hat{y}^{(\delta)} = (1 - \delta^2, \delta)$ and $x^{(\delta)} = (1, 0)$. The conditions $u \ge 1$ and $u + \delta v \le 1$ again force $(u, v) = (1, 0)$. Both projection displacements and the ratio above are therefore unchanged. In particular, the two propositions remain valid when the capacity sets are restricted to polytopes with two resource constraints.

Changing product units cannot remove both large ratios. With only the constraint $u + \delta v \le 1$, the change of units $v' = \delta v$ gives $u + v' \le 1$, and the ratio for this instance becomes 2. With both constraints, consider any change of units $(u, v) \mapsto (au, bv)$, where $a, b > 0$. The instance above then has ratio $1 + (b/a)\delta^{-1}$, while the mirrored target $(\delta, 1 - \delta^2)$ and state $(0, 1)$ have ratio $1 + (a/b)\delta^{-1}$. At least one ratio is at least $1 + \delta^{-1}$.

The corresponding Euclidean estimate also fails, since the ℓ_1 and ℓ_2 norms are equivalent in dimension two. These examples explain why we control the evolution of the geometric distance $q_t = \mathrm{dist}(z_t, J_t)$ directly. The short-move argument of Lemma 3.2 relates common positive demand to a reduction in this distance without assigning the projection displacement to individual products.

D Discussion of assumptions

D.1 Examples of general capacity sets

Multiple shared resources and uncertain resource consumption give capacity sets beyond a single warehouse constraint.

Multiple resources. A single resource constraint with positive consumption coefficients can be written as a warehouse constraint after rescaling product quantities. With m shared resources, the constraints take the form $Ay \le b$, where $A \in \mathbb{R}^{m \times n}_+$ gives consumption rates and $b \in \mathbb{R}^m_+$ gives resource limits. The resources may include storage volume, weight, refrigerated space, and inventory capital. Productwise upper bounds can be added to ensure boundedness. Lyu et al. [10] discuss the operational importance of multiple constraints and provide guarantees under i.i.d. demand; Appendix C explains the difficulty for earlier hidden-target analyses based on a single warehouse constraint.

Random resource consumption. When resource use is random, one may require a capacity constraint to hold with high probability. Such a requirement is called a *chance constraint*; applications in warehouse design include choosing storage-slot heights [22]. For a simple example, suppose the vector of resource coefficients is $c \sim \mathcal{N}(\bar{c}, \Sigma)$, with mean $\bar{c}$ and covariance matrix Σ. For a fixed inventory level y, resource budget b, and $0 < \epsilon < 1/2$, the requirement

$$\mathbb{P}(c^\top y \le b) \ge 1 - \epsilon$$

is equivalent to

$$\bar{c}^\top y + \Phi^{-1}(1 - \epsilon)\|\Sigma^{1/2} y\|_2 \le b,$$

where Φ is the standard normal cumulative distribution function. This follows because $c^\top y$ is normal with mean $\bar{c}^\top y$ and standard deviation $\|\Sigma^{1/2} y\|_2$. The norm term can produce a curved capacity set even when the coefficients are independent, so that Σ is diagonal. The same geometry also appears in chance-constrained knapsack problems [23].

Estimated resource coefficients. Uncertainty can also arise when fixed resource coefficients are estimated from data. Suppose the estimates are independent and satisfy $\hat{c}_i \sim \mathcal{N}(c_i^\star, s_i^2)$, where $c_i^\star$ is the true coefficient and $s_i > 0$ is a known standard error. A confidence region with coverage $1-\alpha$, for $0 < \alpha < 1$, is

$$\mathcal{C}_\alpha := \left\{ c \in \mathbb{R}^n \;\middle|\; \sum_{i=1}^n \frac{(c_i - \hat{c}_i)^2}{s_i^2} \le \chi^2_{n,1-\alpha} \right\},$$

where $\chi^2_{n,1-\alpha}$ is the $(1-\alpha)$-quantile of the chi-squared distribution with n degrees of freedom. Indeed, the sum at $c = c^\star$ is the sum of squares of n independent standard normal variables. Requiring $c^\top y \le b$ for every $c \in \mathcal{C}_\alpha$ gives

$$\hat{c}^\top y + \sqrt{\chi^2_{n,1-\alpha}}\, \| \operatorname{diag}(s_1, \dots, s_n) y \|_2 \le b.$$

This is a second-order-cone constraint, consisting of a Euclidean norm and a linear term, as in the chance constraint above [24]. Independence makes the confidence ellipsoid axis-aligned; it does not turn the ellipsoid into a box. Hence independent estimation errors can also lead to curved capacity sets.

D.2 UPPD, sell-out windows, and tuning

Interpretation and limitations. UPPD requires a conditional probability of at least μ that every product receives demand of at least ρ in the same round. This can be plausible for an assortment of frequently purchased products or when each round spans a sufficiently long period. Perishable goods provide one reason to exclude very slow-moving products. A common demand factor, such as store-wide traffic or a promotion, can also generate positive demand across the assortment. With many products and nearly independent intermittent demands, however, the probability of common positive demand may become very small.

The pathwise reduction needs only windowed clearing (Assumption E.1): every window must contain a round with common demand of at least ρ. UPPD implies this condition with high probability, as shown in Lemma E.4. Any other demand assumption that gives the same window bound can therefore be used in the reduction. Conditions that require demand only on products where inventory exceeds the target may be less restrictive, but would require a different clearing argument.

Comparison with sell-out windows. The same UPPD condition is used in [1], whereas [11] formulate their guarantees using the maximum sell-out window $L_{\max}$. The following proposition shows that UPPD gives a high-probability sell-out-window bound: every product receives at least C units of cumulative demand in every window of length $O\big(\mu^{-1}(\lceil C/\rho \rceil + \log(nT/\delta))\big)$ contained in the horizon.

Proposition D.1. *Consider the linear-capacity model*

$$\mathcal{Y}_{\text{lin}} := \left\{ y \in \mathbb{R}^n_+ \mid \|y\|_1 \le C \right\}, \qquad C > 0.$$

Suppose Assumption 3.3 holds with parameters (μ, ρ), and set $m := \lceil C/\rho \rceil$. For every product $i \in [n]$, window length $L \in \mathbb{N}$, and start time t with $t + L - 1 \le T$,

$$\mathbb{P}\left(\sum_{s=t}^{t+L-1} d_{s,i} < C \right) \le \exp\left(m - \frac{\mu L}{2} \right). \tag{15}$$

For $\delta \in (0,1)$, define

$$L_\delta := \left\lceil \frac{2}{\mu} \left(\left\lceil \frac{C}{\rho} \right\rceil + \log\left(\frac{nT}{\delta} \right) \right) \right\rceil.$$

With probability at least $1-\delta$, every product receives at least C units of demand over every length-L_δ window contained in $[T]$. For windows contained in the horizon, this is the high-probability sell-out condition used in Remark 3 of [11].

Proof. We count rounds of common positive demand; each contributes at least ρ demand to every product. Fix a product i and a window $\{t, \dots, t+L-1\}$, and let $X_s := \mathbb{1}\{d_s^- \ge \rho\}$. Since $d_{s,i} \ge \rho X_s$, the event $\sum_{s=t}^{t+L-1} d_{s,i} < C$ implies $\sum_{s=t}^{t+L-1} X_s \le m - 1$.

UPPD gives $\mathbb{E}[e^{-\theta X_s} \mid \mathcal{F}_{s-1}] \le 1 - \mu + \mu e^{-\theta}$ for every $\theta > 0$. Iterating conditional expectations and applying the exponential Markov inequality yields

$$\mathbb{P}\left(\sum_{s=t}^{t+L-1} X_s \le m - 1 \right) \le e^{\theta m}(1 - \mu + \mu e^{-\theta})^L.$$

Taking $\theta = \log 2$ bounds the right-hand side by

$$2^m(1 - \mu/2)^L \le \exp\left(m - \frac{\mu L}{2} \right),$$

which proves (15). For $L = L_\delta$, this bound is at most $\delta/(nT)$. A union bound over at most nT product–window pairs proves the claim. □

The converse need not hold. For capacity $C = 1$ and two products, demands $d_{2k-1} = (1, 0)$ and $d_{2k} = (0, 1)$ give each product enough demand in every two-round window, but $d_t^- = 0$ in every round. Thus bounded sell-out windows do not imply UPPD or our windowed-clearing condition, which requires common positive demand in a single round.

Information needed for tuning. The stated tuning can use any known valid UPPD pair, including conservative lower bounds on μ and ρ. If several valid pairs are available, their regret bounds can be compared. Under adaptive demand, past frequencies need not give valid lower bounds on future conditional demand probabilities. Obtaining the same regret guarantees without knowing a valid pair in advance remains open.

D.3 Projection computation and approximation errors

Our method uses Euclidean projections onto the capacity set $\mathcal{Y}$ and the implementable set $J_t = \mathcal{Y} \cap \{y \in \mathbb{R}^n \mid y \succeq x_t\}$. These projections are tractable for the capacity examples above. Boxes and a single linear warehouse constraint admit simple projection procedures. With multiple linear constraints, projection minimizes a convex quadratic objective subject to linear constraints. The curved examples in Appendix D.1 have constraints that bound a Euclidean norm by an affine expression; their projections can be written as second-order-cone programs. Adding the inventory lower bounds $y \succeq x_t$ preserves these formulations, so standard solvers can compute the projections to a prescribed accuracy.

Approximate base-learner projections. Theorem 3.4 applies regardless of how the base learner generates its targets. Approximate projections within the base learner are therefore allowed, provided the resulting targets remain in $\mathcal{Y}$ and satisfy the required movement bound. Their effect on the base learner's regret must be included in Δ_T^{base}.

Approximate implementation projections. Suppose the implementation layer returns a feasible point $y_t \in J_t$, and define its projection error and the distance from the target to the implementable set by

$$e_t := \|y_t - \mathrm{Proj}_{J_t}(z_t)\|_2, \qquad \widehat{q}_t := \mathrm{dist}(z_t, J_t).$$

Write $a_t := \|z_{t+1} - z_t\|_2$ for $t < T$. Then $\|y_t - z_t\|_2 \le \widehat{q}_t + e_t$. Applying the geometric argument of Proposition 3.1 to the actual implemented point y_t gives

$$\widehat{q}_{t+1} \le \left[\widehat{q}_t + a_t + e_t - d_t^-\right]^+.$$

Thus implementation errors add to the queue arrivals. Since $x_1 = 0$ and $z_1 \in \mathcal{Y}$, we have $\widehat{q}_1 = 0$. Under windowed clearing (Assumption E.1), suppose

$$\sum_{r=t}^{t+B-1} (a_r + e_r) \le \rho \qquad \text{for every } t \in [T - B].$$

The proof of Theorem E.2, with $a_t + e_t$ in place of target movement, then bounds the cumulative distance $\sum_t \widehat{q}_t$. Combining this bound with the regret decomposition gives

$$R_T(u_{1:T}) \le \Delta_T^{\text{base}}(u_{1:T}) + GB \sum_{t=1}^{T-1} (a_t + e_t) + G \sum_{t=1}^{T} e_t.$$

Under UPPD, windowed clearing with $B = B_\delta$ holds with probability at least $1 - \delta$. Retaining the stated regret rates therefore requires controlling both cumulative projection errors and their contribution within each clearing window.

E Proofs for Sections 2 and 3

E.1 Deterministic clearing and pathwise transfer

The main text works directly with UPPD. The proofs pass through the following deterministic clearing event.

Assumption E.1 (Windowed clearing)**.** There exist $B \in \mathbb{N}$ and $\rho > 0$ such that every interval of B consecutive rounds contained in $[T]$ contains a time s with $d_s^- \geq \rho$.

Theorem E.2 (Pathwise implementation bound)**.** *Suppose $q_1 = 0$, Assumption E.1 holds, and for every $t \in [T - B]$,*

$$\sum_{r=t}^{t+B-1} \|z_{r+1} - z_r\|_2 \leq \rho. \tag{16}$$

Then

$$\Delta_T^{gap} = \sum_{t=1}^{T} q_t \leq B \sum_{t=1}^{T-1} \|z_{t+1} - z_t\|_2 = B\Delta_T^{mv}. \tag{17}$$

Theorem E.3 (OIO regret as target regret plus movement)**.** *Suppose $q_1 = 0$, Assumption E.1 holds, and* (16) *is satisfied. Then for every comparator sequence $u_{1:T} \in \mathcal{Y}^T$,*

$$R_T(u_{1:T}) \leq \Delta_T^{base}(u_{1:T}) + GB\Delta_T^{mv}. \tag{18}$$

Lemma E.4 (Probabilistic clearing under UPPD)**.** *Under Assumption 3.3, for every $B \in \mathbb{N}$,*

$$\mathbb{P}\Big(\exists t \in [T - B + 1] \text{ such that } d_s^- < \rho \text{ for all } s = t, \ldots, t + B - 1\Big) \leq Te^{-\mu B}.$$

In particular, for any $\delta \in (0, 1)$, set $B_\delta := \lceil \mu^{-1} \log(T/\delta) \rceil$. With probability at least $1 - \delta$, Assumption E.1 holds with $B = B_\delta$.

E.2 Proof of Lemma 2.2

Proof. For any $t \in [T]$, convexity of ℓ_t at y_t and the subgradient $g_t \in \partial\ell_t(y_t)$ gives

$$\ell_t(y_t) - \ell_t(u_t) \leq \langle g_t, y_t - u_t \rangle = \langle g_t, z_t - u_t \rangle + \langle g_t, y_t - z_t \rangle.$$

Since $\|g_t\|_2 \leq G$ by Assumption 2.1 and using Cauchy-Schwarz, we have

$$\langle g_t, y_t - z_t \rangle \leq G\|y_t - z_t\|_2.$$

Summing over t proves (5). □

E.3 Proof of Proposition 3.1

The proof of the target-implementation gap recursion (Proposition 3.1) relies on the geometric Lemma 3.2 relating the distance between order levels and the inventory dynamics constraint (2). We first present a proof of this lemma and then proceed to a proof of the proposition:

Proof of Lemma 3.2. For each coordinate i,

$$y_i' \geq y_i - |y_i' - y_i| \geq y_i - \|y' - y\|_2 \geq y_i - d^- \geq y_i - d_i.$$

Since $y_i' \geq 0$, this implies

$$y_i' \geq \max\{y_i - d_i, 0\} = (y_i - d_i)^+$$

□

Proof of Proposition 3.1. Fix $t \in [T-1]$ and set $r_t := \|z_{t+1} - y_t\|_2$. We break the proof into cases dependent on the size of r_t relative to d_t^-:

Case 1. If $r_t \le d_t^-$, then Lemma 3.2 yields

$$z_{t+1} \succeq (y_t - d_t)^+ \succeq x_{t+1},$$

where the second inequality uses (2). Since $z_{t+1} \in \mathcal{Y}$, we have $z_{t+1} \in J_{t+1}$ and therefore $q_{t+1} = 0$.

Case 2. Now suppose $r_t > d_t^-$. Define

$$\alpha_t := \frac{d_t^-}{r_t} \in [0, 1) \qquad \text{and} \qquad w_t := (1-\alpha_t)y_t + \alpha_t z_{t+1}.$$

Because $\mathcal{Y}$ is convex and $y_t, z_{t+1} \in \mathcal{Y}$, we have $w_t \in \mathcal{Y}$. Also,

$$\|w_t - y_t\|_2 = \|\alpha_t(z_{t+1} - y_t)\|_2 = \alpha_t r_t = d_t^-.$$

Then applying Lemma 3.2 gives

$$w_t \succeq (y_t - d_t)^+ \succeq x_{t+1},$$

so $w_t \in J_{t+1}$. By optimality of Euclidean projection,

$$q_{t+1} \le \|z_{t+1} - w_t\|_2 = \|(1-\alpha_t)z_{t+1} - (1-\alpha_t)y_t\|_2 = (1-\alpha_t)r_t = r_t - d_t^-$$

Finally,

$$r_t = \|z_{t+1} - y_t\|_2 \le \|z_{t+1} - z_t\|_2 + \|z_t - y_t\|_2 = \|z_{t+1} - z_t\|_2 + q_t.$$

So we may write:

$$q_{t+1} \le q_t + \|z_{t+1} - z_t\|_2 - d_t^-$$

Combining the two cases proves (7). □

E.4 Proof of Theorem E.2

Proof. We divide the horizon into excursions of the queue away from zero. On each excursion, repeated use of the queue recursion bounds the queue by cumulative hidden movement. The windowed-clearing assumption then forces every full excursion to end within at most B rounds, and the last partial excursion has length at most B as well.

For $t \in [T-1]$, set $a_t := \|z_{t+1} - z_t\|_2$. We first show that if the queue is empty and at least B rounds remain, then it empties again within the next B rounds. Fix $\tau \le T - B$ with $q_\tau = 0$. Repeated use of (7), together with $d_r^- \ge 0$, gives

$$q_{t+1} \le \sum_{r=\tau}^{t} a_r \qquad \text{for every } t \in \{\tau, \dots, \tau + B - 1\}. \tag{19}$$

By Assumption E.1, there exists $s \in \{\tau, \dots, \tau + B - 1\}$ with $d_s^- \ge \rho$. Applying (7) at time s, then using $q_s = 0$ if $s = \tau$ or (19) otherwise, and then finally applying (16),

$$q_{s+1} \le \big[q_s + a_s - d_s^-\big]^+ \le \Big[\sum_{r=\tau}^{s} a_r - d_s^-\Big]^+ \le \Big[\sum_{r=\tau}^{\tau+B-1} a_r - \rho\Big]^+ = 0.$$

Thus, if $q_\tau = 0$ and $\tau \le T - B$, then there exists $\tau' \in \{\tau + 1, \dots, \tau + B\}$ with $q_{\tau'} = 0$. In other words, whenever the queue is empty, it must empty again within the next B rounds.

Define the times at which the queue is zero recursively. Set $\tau_1 = 1$. If $\tau_j \le T - B$, let τ_{j+1} be the smallest index in $\{\tau_j + 1, \dots, \tau_j + B\}$ with $q_{\tau_{j+1}} = 0$. Otherwise, set $\tau_{j+1} = T + 1$ and stop. Let m be the final index before this stopping step, so

$$1 = \tau_1 < \tau_2 < \cdots < \tau_m < \tau_{m+1} = T + 1.$$

By construction, $q_{\tau_j} = 0$ for every $j \in [m]$, and $\tau_m > T - B$. Each interval has length at most B, including the last one because $\tau_{m+1} - \tau_m = T + 1 - \tau_m \le B$.

Fix $j \in [m]$. Starting from $q_{\tau_j} = 0$, repeated use of the queue recursion bounds the sum over this interval by

$$\sum_{t=\tau_j}^{\tau_{j+1}-1} q_t \le \sum_{t=\tau_j}^{\tau_{j+1}-1} \sum_{r=\tau_j}^{t-1} a_r = \sum_{r=\tau_j}^{\tau_{j+1}-2} (\tau_{j+1} - 1 - r) a_r \le B \sum_{r=\tau_j}^{\tau_{j+1}-2} a_r.$$

where the last inequality used $\tau_{j+1} - \tau_j \le B$. The index sets $\{\tau_j, \dots, \tau_{j+1} - 2\}$ are disjoint subsets of $[T-1]$, so summing over j yields

$$\Delta_T^{\text{gap}} = \sum_{j=1}^{m} \sum_{t=\tau_j}^{\tau_{j+1}-1} q_t \le B \sum_{t=1}^{T-1} a_t,$$

which is exactly (17). □

E.5 Proof of Lemma E.4

Proof. For each $s \in [T]$, define the event $A_s := \{d_s^- < \rho\}$. Then since d_s is $\mathcal{F}_s$-measurable, so is d_s^- and hence A_s. By UPPD, we have

$$\mathbb{P}(A_s | \mathcal{F}_{s-1}) = 1 - P(d_s^- \ge \rho | \mathcal{F}_{s-1}) \le 1 - \mu \quad \text{a.s.} \tag{20}$$

Fixing any $t \in [T - B + 1]$, we first show by induction on $k \in [B]$ that $\mathbb{P}(\cap_{s=t}^{t+k-1} A_s) \le (1-\mu)^k$. The case for $k = 1$ follows from (20) and the tower property:

$$\mathbb{P}(A_t) = \mathbb{E}[\mathbb{P}(A_t | \mathcal{F}_{t-1})] \le 1 - \mu$$

For the inductive step with $1 \le k < B$, we have

$$\begin{aligned}
\mathbb{P}\left(\bigcap_{s=t}^{t+k} A_s\right) &= \mathbb{E}[\mathbb{1}_{A_t} \cdots \mathbb{1}_{A_{t+k}}] \\
&= \mathbb{E}\left[\mathbb{E}[\mathbb{1}_{A_t} \cdots \mathbb{1}_{A_{t+k}} | \mathcal{F}_{t+k-1}]\right] \\
&= \mathbb{E}[\mathbb{1}_{A_t} \cdots \mathbb{1}_{A_{t+k-1}} \cdot \mathbb{P}(A_{t+k} | \mathcal{F}_{t+k-1})] \\
&\le \mathbb{E}[\mathbb{1}_{A_t} \cdots \mathbb{1}_{A_{t+k-1}} (1-\mu)] \\
&= (1-\mu)\mathbb{P}\left(\bigcap_{s=t}^{t+k-1} A_s\right) \\
&\le (1-\mu)(1-\mu)^k = (1-\mu)^{k+1}
\end{aligned}$$

where the second equality is due to the tower property, the first inequality follows from (20), and the last bound follows by the inductive hypothesis. This completes the induction. Now setting $k = B$ and using $1 - \mu \le e^{-\mu}$, we have

$$\mathbb{P}\left(\bigcap_{s=t}^{t+B-1} A_s\right) \le (1-\mu)^B \le e^{-\mu B}$$

The event that we wish to show has low probability is then $\mathcal{A} := \cup_{t=1}^{T-B+1} \cap_{s=t}^{t+B-1} A_s$. Combining the previous bound with a union bound, we have

$$\mathbb{P}(\mathcal{A}) \le \sum_{t=1}^{T-B+1} \mathbb{P}\left(\bigcap_{s=t}^{t+B-1} A_s\right) \le \sum_{t=1}^{T-B+1} e^{-\mu B} \le T e^{-\mu B}$$

which completes the first part of the proof.

For the second part, since $\mu B_\delta \ge \log(T/\delta)$, we may bound $T e^{-\mu B_\delta}$ by $T e^{-\log(T/\delta)} = \delta$ thereby proving the second part of the lemma. □

E.6 Proof of Theorem 3.4

Proof. By Lemma E.4, windowed clearing with $B = B_\delta$ holds with probability at least $1 - \delta$. On this event, the movement condition (8) is exactly (16) with $B = B_\delta$. Theorem E.2 gives $\Delta_T^{\text{gap}} \le B_\delta \Delta_T^{\text{mv}}$, and combining this with Lemma 2.2 proves the regret bound. □

F Proofs for Section 4

F.1 General convex losses

We first record the standard movement and regret estimates for projected OGD:

Lemma F.1. *Assume $0 < \eta_T \le \cdots \le \eta_1$. For the target states z_t and subgradients g_t in HT-OGD (Algorithm 2) and every comparator $u \in \mathcal{Y}$,*

$$\sum_{t=1}^{T} \langle g_t, z_t - u\rangle \le \frac{D^2}{2\eta_T} + \frac{1}{2}\sum_{t=1}^{T} \eta_t \|g_t\|_2^2. \tag{21}$$

Proof. Using the linearity of the inner product, we may write:

$$\langle g_t, z_t - u\rangle = \frac{1}{2\eta_t}\left(\|z_t - u\|_2^2 + \eta_t^2\|g_t\|_2^2 - \|(z_t - u) - \eta_t g_t\|_2^2\right)$$

We relate this to z_{t+1} using nonexpansiveness of the Euclidean projection:

$$\|z_{t+1} - u\|_2^2 = \|\operatorname{Proj}_{\mathcal{Y}}(z_t - \eta_t g_t) - \operatorname{Proj}_{\mathcal{Y}}(u)\|_2^2 \le \|(z_t - \eta_t g_t) - u\|_2^2$$

Combining the previous two lines results in:

$$\langle g_t, z_t - u\rangle \le \frac{1}{2\eta_t}\left(\|z_t - u\|_2^2 + \eta_t^2\|g_t\|_2^2 - \|z_{t+1} - u\|_2^2\right)$$

Summing over $t \in [T]$ yields:

$$\begin{aligned}
\sum_{t=1}^{T}\langle g_t, z_t - u\rangle &\le \sum_{t=1}^{T}\frac{1}{2\eta_t}\left(\|z_t - u\|_2^2 - \|z_{t+1} - u\|_2^2\right) + \frac{1}{2}\sum_{t=1}^{T}\eta_t\|g_t\|_2^2 \\
&= \frac{1}{2}\left(\frac{\|z_1 - u\|_2^2}{\eta_1} - \frac{\|z_{T+1} - u\|_2^2}{\eta_T} + \sum_{t=2}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)\|z_t - u\|_2^2\right) \\
&\quad + \frac{1}{2}\sum_{t=1}^{T}\eta_t\|g_t\|_2^2 \\
&\le \frac{1}{2}\left(\frac{D^2}{\eta_1} + \sum_{t=2}^{T}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}\right)D^2\right) + \frac{1}{2}\sum_{t=1}^{T}\eta_t\|g_t\|_2^2 \\
&= \frac{D^2}{2\eta_T} + \frac{1}{2}\sum_{t=1}^{T}\eta_t\|g_t\|_2^2
\end{aligned}$$

where the last inequality uses the nonincreasing stepsizes and Assumption 2.1. □

We first prove the regret bound under windowed clearing.

Theorem F.2 (Convex losses under windowed clearing)**.** *Assume the bounded convex OIO setting (Assumption 2.1), the inventory dynamics constraint* (2)*, and windowed clearing (Assumption E.1). Choose a learning rate parameter γ and define step sizes η_t according to:*

$$0 < \gamma \le \frac{\rho}{2D\sqrt{B}}, \qquad \eta_t = \frac{\gamma D}{G\sqrt{t}}. \tag{22}$$

Run HT-OGD (Algorithm 2) with this choice of η_t from an initial feasible target $z_1 \in J_1$. Then

$$\Delta_T^{gap} \le 2\gamma DB\sqrt{T}, \tag{23}$$

and, for every $u \in \mathcal{Y}$,

$$R_T(u) \le DG\Big(\frac{1}{2\gamma} + \gamma + 2\gamma B\Big)\sqrt{T}. \tag{24}$$

In particular, choosing $\gamma = c/\sqrt{B}$ with fixed $0 < c \le \rho/(2D)$ gives regret $O(DG\sqrt{BT})$, with a constant depending on c.

Proof. We start by showing that this algorithm satisfies (16). By nonexpansiveness of Euclidean projection and (22),

$$\|z_{t+1} - z_t\|_2 = \|\operatorname{Proj}_{\mathcal{Y}}(z_t - \eta_t g_t) - \operatorname{Proj}_{\mathcal{Y}}(z_t)\|_2 \le \eta_t \|g_t\|_2 \le \frac{\gamma D}{\sqrt{t}}.$$

Then for $t \in [T - B]$, we can bound the sum over the next B periods as:

$$\sum_{r=t}^{t+B-1} \|z_{r+1} - z_r\|_2 \le \gamma D \sum_{r=t}^{t+B-1} \frac{1}{\sqrt{r}} \le \gamma D \sum_{r=1}^{B} \frac{1}{\sqrt{r}} \le 2\gamma D\sqrt{B} \le \rho$$

where the last inequality used (22). Then Theorem E.2 applies with this value of B, yielding (23) by bounding $\Delta_T^{\text{mv}} \le \gamma D \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \le 2\gamma D\sqrt{T}$.

For the base-learner term, Lemma F.1 and the choice of η_t gives

$$\sum_{t=1}^{T} \langle g_t, z_t - u\rangle \le \frac{D^2}{2\eta_T} + \frac{1}{2}\sum_{t=1}^{T} \eta_t \|g_t\|_2^2 \le \frac{DG}{2\gamma}\sqrt{T} + \gamma DG\sqrt{T}.$$

Combining this with Lemma 2.2 and (23) proves (24). □

Proof of Theorem 4.1. Using $B = B_\delta$ and Lemma E.4, Assumption E.1 holds with probability at least $1 - \delta$. On that event, apply Theorem F.2.

For the expectation bound, using $\delta = 1/T$ and learning rate parameter $\gamma = \rho/(2D\sqrt{B_{1/T}})$, the conditions of the high probability are satisfied. Let E_T be the event that the high probability bound holds so that we have $\mathbb{P}(E_T) \ge 1 - 1/T$. On E_T, we have the regret bound

$$R_T(u) \le DG\Big(\frac{1}{2\gamma} + \gamma + 2\gamma B_{1/T}\Big)\sqrt{T} \le DG\Big(\frac{D}{\rho} + \frac{3\rho}{2D}\Big)\sqrt{B_{1/T}T}$$

where we used $1 \le B_{1/T}$ in the second inequality. On the complement, we use the crude bound $R_T(u) \le DGT$. Therefore

$$\mathbb{E}\big[R_T(u)\big] \le DG\Big(\frac{D}{\rho} + \frac{3\rho}{2D}\Big)\sqrt{B_{1/T}T} + \frac{1}{T} \cdot DGT,$$

which proves the claim. □

F.2 A matching lower bound under UPPD

Proof of Theorem 4.2. We construct demand blocks during which the learner cannot change its action, then choose losses that make these commitments costly. Write $s := D/\sqrt{2}$, so the capacity set

$$\mathcal{Y}_D = \{y \in \mathbb{R}_+^2 \mid y_1 + y_2 = s\}$$

has diameter D. Set $x_1 = (0, 0)$ and $r := \max\{\rho, s\}$. Let $\xi_1, \dots, \xi_T$ be independent Bernoulli(μ) variables, independent of the learner's randomness, and define

$$d_t := r\xi_t(1, 1), \qquad x_{t+1} := (y_t - d_t)^+.$$

These demands satisfy UPPD with parameters (μ, ρ). If $\xi_t = 1$, the demand clears all inventory because every $y \in \mathcal{Y}_D$ satisfies $y \preceq r(1, 1)$. If $\xi_t = 0$, feasibility requires $y_{t+1} \succeq y_t$; since both actions have coordinate sum s, they must be equal.

Thus the learner plays a constant action on each block between successive clearings. Let

$$K := 1 + \sum_{t=1}^{T-1} \xi_t$$

be the number of blocks, with start times $t_1 = 1 < \cdots < t_K$ and $t_{K+1} := T + 1$. Write $L_j := t_{j+1} - t_j$ for their lengths, so $\sum_{j=1}^{K} L_j = T$. When block j begins, after the learner chooses

its action a_j, draw a fresh sign σ_j, uniformly from $\{-1, +1\}$ and independently of the demands, earlier signs, and learner randomness. For every round in this block, set

$$\ell_t(y) := G\left(\frac{1-\sigma_j}{2}y_1 + \frac{1+\sigma_j}{2}y_2\right).$$

These losses are linear and have gradient norm G.

The learner's expected loss in block j is $GsL_j/2$. Indeed, conditional on the demand path, the learner's random seed, and the earlier signs, a_j is determined, $a_{j,1} + a_{j,2} = s$, and σ_j remains uniform and independent. Summing over blocks gives

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(y_t)\right] = \frac{GsT}{2}.$$

The best static comparator can instead choose the endpoint favored by the signs. Define the signed sum of block lengths

$$S := \sum_{j=1}^{K} L_j \sigma_j.$$

The cumulative losses at $(s, 0)$ and $(0, s)$ are $Gs(T - S)/2$ and $Gs(T + S)/2$, respectively. Since cumulative loss is linear on $\mathcal{Y}_D$,

$$\min_{u\in\mathcal{Y}_D} \sum_{t=1}^{T} \ell_t(u) = \frac{Gs}{2}(T - |S|).$$

Consequently,

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(y_t) - \min_{u\in\mathcal{Y}_D} \sum_{t=1}^{T} \ell_t(u)\right] = \frac{Gs}{2}\mathbb{E}|S|.$$

We now bound this expected imbalance using the squared block lengths. Conditional on $L_{1:K}$, the signs are independent and uniform, so

$$\mathbb{E}[S^2 \mid L_{1:K}] = \sum_{j=1}^{K} L_j^2 =: V, \qquad \mathbb{E}[S^4 \mid L_{1:K}] = 3V^2 - 2\sum_{j=1}^{K} L_j^4 \le 3V^2.$$

Hölder's inequality gives

$$V \le \big(\mathbb{E}[|S| \mid L_{1:K}]\big)^{2/3}\big(\mathbb{E}[S^4 \mid L_{1:K}]\big)^{1/3}, \qquad \text{hence} \qquad \mathbb{E}[|S| \mid L_{1:K}] \ge \sqrt{V/3}.$$

Since the block lengths sum to T, Cauchy–Schwarz yields $V \ge T^2/K$. Also, $T \ge \mu^{-1}$ implies $\mathbb{E}K = 1 + (T - 1)\mu \le 2\mu T$. Taking expectations and applying Jensen's inequality to $k \mapsto k^{-1/2}$ therefore gives

$$\mathbb{E}|S| \ge \frac{1}{\sqrt{3}}\mathbb{E}\sqrt{V} \ge \frac{T}{\sqrt{3}}\mathbb{E}[K^{-1/2}] \ge \frac{T}{\sqrt{3\mathbb{E}K}} \ge \sqrt{\frac{T}{6\mu}}.$$

Substituting into the regret identity and using $s = D/\sqrt{2}$ proves

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t(y_t) - \min_{u\in\mathcal{Y}_D} \sum_{t=1}^{T} \ell_t(u)\right] \ge \frac{DG}{4\sqrt{3}}\sqrt{\frac{T}{\mu}}.$$

Thus the theorem holds with the universal constant $c = 1/(4\sqrt{3})$. □

F.3 Strongly convex losses

Theorem F.3 (Strongly convex losses under windowed clearing)**.** *Assume the bounded convex OIO setting (Assumption 2.1), the inventory dynamics constraint* (2)*, and windowed clearing (Assumption E.1). In addition, suppose that each loss function ℓ_t is α-strongly convex on $\mathcal{Y}$ for some $\alpha > 0$. Define a constant s_B and the stepsize η_t by*

$$s_B := \max\left\{1, \left\lceil \frac{B}{e^{\alpha\rho/(2G)} - 1} \right\rceil\right\}, \qquad \eta_t = \frac{2}{\alpha(t + s_B)}.$$

Run HT-OGD (Algorithm 2) from an initial target $z_1 \in J_1$. Then

$$\Delta_T^{gap} \le \frac{2GB}{\alpha} \log\Big(1 + \frac{T}{s_B}\Big), \tag{25}$$

and, for every $u \in \mathcal{Y}$,

$$R_T(u) \le \frac{\alpha s_B D^2}{4} + \frac{G^2}{\alpha} \log\Big(1 + \frac{T}{s_B}\Big) + \Big(G + \frac{\alpha D}{2}\Big)\Delta_T^{gap}. \tag{26}$$

In particular,

$$R_T(u) = O\Big(s_B + B \log\big(1 + \tfrac{T}{s_B}\big)\Big) = O\left(B(1 + \log(1 + T/B))\right), \tag{27}$$

where the asymptotic bound has absorbed constants only dependent on α, ρ, D, and G.

Proof. We first control the hidden target movement over each clearing window; the queue bound will then follow from Theorem E.2. By construction of s_B,

$$s_B \ge \frac{B}{e^{\alpha\rho/(2G)} - 1}$$

and therefore

$$\frac{2G}{\alpha} \log\Big(1 + \frac{B}{s_B}\Big) \le \rho \tag{28}$$

This will be used later in the proof. Now aiming to prove that this choice of step size satisfies (16), we first use nonexpansiveness of the Euclidean projection to yield

$$\|z_{t+1} - z_t\|_2 = \|\operatorname{Proj}_{\mathcal{Y}}(z_t - \eta_t g_t) - \operatorname{Proj}_{\mathcal{Y}}(z_t)\|_2 \le \|\eta_t g_t\|_2 \le \eta_t G = \frac{2G}{\alpha(t + s_B)}$$

Hence, for every $t \in [T - B]$,

$$\sum_{r=t}^{t+B-1} \|z_{r+1} - z_r\|_2 \le \frac{2G}{\alpha} \sum_{r=t}^{t+B-1} \frac{1}{r + s_B} \le \frac{2G}{\alpha} \sum_{r=1}^{B} \frac{1}{r + s_B} \le \frac{2G}{\alpha} \log\Big(1 + \frac{B}{s_B}\Big) \le \rho$$

where the second to last inequality used $\sum_{n=1}^{N} \frac{1}{n+A} \le \log(\frac{N+A}{A})$ and the last inequality used (28). Thus, Theorem E.2 applies and gives

$$\Delta_T^{\text{gap}} \le B \sum_{t=1}^{T-1} \|z_{t+1} - z_t\|_2 \le \frac{2GB}{\alpha} \sum_{t=1}^{T-1} \frac{1}{t + s_B} \le \frac{2GB}{\alpha} \log\Big(1 + \frac{T}{s_B}\Big)$$

which is (25). For the regret bound, we revisit the subgradient bound on the loss difference at the implemented point and the comparator. Strong convexity at the implemented point y_t and subgradient $g_t \in \partial \ell_t(y_t)$ gives

$$\ell_t(y_t) - \ell_t(u) \le \langle g_t, y_t - u \rangle - \frac{\alpha}{2}\|y_t - u\|_2^2$$

Since $y_t - u = (z_t - u) + (y_t - z_t)$ and $\|y_t - z_t\|_2 = q_t$, we have

$$\ell_t(y_t) - \ell_t(u) \le \langle g_t, z_t - u \rangle + G q_t - \frac{\alpha}{2}\|y_t - u\|_2^2$$

Using triangle inequality and AM-GM inequality,

$$\|z_t - u\|_2^2 \le 2\|y_t - u\|_2^2 + 2q_t^2$$

so rearranging,

$$\|y_t - u\|_2^2 \ge \frac{1}{2}\|z_t - u\|_2^2 - q_t^2$$

Combining,

$$\ell_t(y_t) - \ell_t(u) \le \langle g_t, z_t - u \rangle - \frac{\alpha}{4}\|z_t - u\|_2^2 + G q_t + \frac{\alpha}{2} q_t^2 \tag{29}$$

As shown in the proof of Lemma F.1, the projected-gradient update gives the bound

$$\langle g_t, z_t - u\rangle \le \frac{\|z_t - u\|_2^2 - \|z_{t+1} - u\|_2^2}{2\eta_t} + \frac{\eta_t G^2}{2}$$

Combining this with (29),

$$\ell_t(y_t) - \ell_t(u) \le \Big(\frac{1}{2\eta_t} - \frac{\alpha}{4}\Big)\|z_t - u\|_2^2 - \frac{1}{2\eta_t}\|z_{t+1} - u\|_2^2 + \frac{\eta_t G^2}{2} + Gq_t + \frac{\alpha}{2}q_t^2$$

Because $\eta_t = 2/(\alpha(t + s_B))$, we have

$$\frac{1}{2\eta_t} - \frac{\alpha}{4} = \frac{\alpha(t + s_B - 1)}{4} = \frac{1}{2\eta_{t-1}}$$

where $\eta_0 := 2/(\alpha s_B)$. With this identity, summing the preceding inequality over $t \in [T]$ telescopes and gives

$$\begin{aligned} R_T(u) &\le \frac{1}{2}\sum_{t=1}^{T}\left(\frac{\|z_t - u\|_2^2}{\eta_{t-1}} - \frac{\|z_{t+1} - u\|_2^2}{\eta_t}\right) + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t + G\sum_{t=1}^{T}q_t + \frac{\alpha}{2}\sum_{t=1}^{T}q_t^2 \\ &\le \frac{\|z_1 - u\|_2^2}{2\eta_0} + \frac{G^2}{2}\sum_{t=1}^{T}\eta_t + G\Delta_T^{\text{gap}} + \frac{\alpha}{2}\sum_{t=1}^{T}q_t^2. \end{aligned}$$

Since $z_1, u, z_t, y_t \in \mathcal{Y}$, then Assumption 2.1 implies $\|z_1 - u\|_2 \le D$ and $q_t \le D$, so

$$\frac{1}{2\eta_0}\|z_1 - u\|_2^2 \le \frac{\alpha s_B D^2}{4}, \qquad \sum_{t=1}^{T}q_t^2 \le D\sum_{t=1}^{T}q_t = D\Delta_T^{\text{gap}}$$

All that remains is to bound the sum of the stepsizes:

$$\sum_{t=1}^{T}\eta_t = \frac{2}{\alpha}\sum_{t=1}^{T}\frac{1}{t + s_B} \le \frac{2}{\alpha}\log\Big(1 + \frac{T}{s_B}\Big).$$

Substituting these estimates proves (26).

Simplifying the regret bound, we have

$$\begin{aligned} R_T(u) &\le \frac{\alpha s_B D^2}{4} + \frac{G^2}{\alpha}\log\left(1 + \frac{T}{s_B}\right) + \frac{2GB}{\alpha}\left(G + \frac{\alpha D}{2}\right)\log\left(1 + \frac{T}{s_B}\right) \\ &= \frac{\alpha D^2}{4}s_B + \left(\frac{G^2}{\alpha}(1 + 2B) + GDB\right)\log\left(1 + \frac{T}{s_B}\right) \\ &= O\big(s_B + B\log(1 + T/s_B)\big), \end{aligned}$$

where the constants depend on α, D, and G. Since $s_B = \Theta\big(B/(e^{\alpha\rho/(2G)} - 1)\big)$, this gives

$$R_T(u) = O\big(B(1 + \log(1 + T/B))\big).$$

□

Proof of Theorem 4.3. By Lemma E.4, Assumption E.1 with $B = B_\delta$ holds with probability at least $1 - \delta$. On that event, Theorem F.3 applies. Since $B_\delta = \Theta\big(\mu^{-1}\log(T/\delta)\big)$, the bound in (27) becomes $O(B_\delta \log T)$, which is the claimed rate. □

F.4 Dynamic regret

Throughout, we extend the comparator sequence as necessary by setting $u_{T+1} := u_T$

F.4.1 OGD with known path variation

Proposition F.4. *Assume the bounded convex OIO setting (Assumption 2.1), the inventory dynamics constraint* (2)*, and windowed clearing (Assumption E.1). For a constant stepsize $\eta_t \equiv \eta$ with $\eta \in (0, \rho/(BG)]$, run HT-OGD (Algorithm 2) with an initial feasible target $z_1 \in J_1$. Then for every comparator sequence $u_{1:T} \in \mathcal{Y}^T$ with path length $P_{T,2}$, we have*

$$R_T(u_{1:T}) \le \frac{D^2 + 2DP_{T,2}}{2\eta} + \Big(B + \frac{1}{2}\Big)\eta G^2 T. \tag{30}$$

Consequently, if $P_{T,2}$ is known and the corresponding tuned stepsize $\eta = \frac{1}{G}\sqrt{\frac{D(D+2P_{T,2})}{(2B+1)T}}$ for (30) *also satisfies $\eta \le \rho/(BG)$, then Hidden-Target OGD yields*

$$R_T(u_{1:T}) \le G\sqrt{(2B+1)D(D+2P_{T,2})T}. \tag{31}$$

Proof of Proposition F.4. First, we show that this stepsize satisfies (16). Since $\|z_{t+1} - z_t\|_2 \le \eta G$, the hypothesis $\eta \le \rho/(BG)$ implies

$$\sum_{r=t}^{t+B-1} \|z_{r+1} - z_r\|_2 \le B\eta G \le \rho$$

for every $t \in [T - B]$. Theorem E.3 therefore gives

$$R_T(u_{1:T}) \le \Delta_T^{\text{base}} + GB\Delta_T^{\text{mv}}$$

By using $\|z_{t+1} - z_t\|_2 \le \eta G$, we must have $\Delta_T^{\text{mv}} \le \eta GT$ so the latter sum is at most $B\eta G^2 T$.

For the base-learner regret Δ_T^{base}, as in the proof of Lemma F.1, the projected OGD update with constant stepsize gives

$$\langle g_t, z_t - u_t\rangle \le \frac{\|z_t - u_t\|_2^2 - \|z_{t+1} - u_t\|_2^2}{2\eta} + \frac{\eta G^2}{2}$$

By inserting and subtracting $\|z_{t+1} - u_{t+1}\|_2^2$ then summing over t, we may telescope the series:

$$\begin{aligned}
\sum_{t=1}^{T}\langle g_t, z_t - u_t\rangle &\le \frac{1}{2\eta}\sum_{t=1}^{T}\big(\|z_{t+1} - u_{t+1}\|_2^2 - \|z_{t+1} - u_t\|_2^2\big) \\
&\quad + \frac{1}{2\eta}\sum_{t=1}^{T}\big(\|z_t - u_t\|_2^2 - \|z_{t+1} - u_{t+1}\|_2^2\big) + \frac{\eta G^2 T}{2} \\
&= \frac{1}{2\eta}\sum_{t=1}^{T}\big(\|z_{t+1} - u_{t+1}\|_2^2 - \|z_{t+1} - u_t\|_2^2\big) \\
&\quad + \frac{1}{2\eta}\big(\|z_1 - u_1\|_2^2 - \|z_{T+1} - u_{T+1}\|_2^2\big) + \frac{\eta G^2 T}{2} \\
&\le \frac{1}{2\eta}\|z_1 - u_1\|_2^2 + \frac{1}{2\eta}\sum_{t=1}^{T}\big(\|z_{t+1} - u_{t+1}\|_2^2 - \|z_{t+1} - u_t\|_2^2\big) + \frac{\eta G^2 T}{2}
\end{aligned}$$

Since $\|z_1 - u_1\|_2 \le D$ and

$$\|z_{t+1} - u_{t+1}\|_2^2 - \|z_{t+1} - u_t\|_2^2 \le 2D\|u_{t+1} - u_t\|_2$$

using the reverse triangle inequality and the diameter bound on $\mathcal{Y}$, we obtain

$$\sum_{t=1}^{T}\langle g_t, z_t - u_t\rangle \le \frac{D^2 + 2DP_{T,2}}{2\eta} + \frac{\eta G^2 T}{2}$$

Combining the two bounds proves (30). Balancing the two terms, the unconstrained minimizer is

$$\eta^* = \frac{1}{G}\sqrt{\frac{D(D+2P_{T,2})}{(2B+1)T}}$$

Provided that $\eta^* \le \rho/(BG)$, then substituting η^* into the regret bound (30) evaluates to the regret bound in (31).

□

Proof of Theorem 4.4. By Lemma E.4, Assumption E.1 with $B = B_\delta$ holds with probability at least $1-\delta$. On that event, we may apply Proposition F.4 with the optimal stepsize η. Since $B_\delta = \Theta(\mu^{-1}\log(T/\delta))$, we assume that $\eta \in (0, \rho/(B_\delta G)]$, then (31) applies and the asymptotic regret bound follows. □

F.4.2 SOGD with unknown path variation

To apply the queue reduction when the comparator's path variation is unknown, we need two guarantees from the base learner: a regret bound that includes movement costs, and a bound on movement within each clearing window. Smoothed OGD (SOGD) of Zhang et al. [12] provides both. It combines OGD experts with different stepsizes, and was previously used for hidden-target inventory learning under a linear capacity constraint [11]. We show how its guarantees apply through our reduction on general convex capacity sets.

Algorithm 3: Hidden-target method with an SOGD base learner

Input: Domain $\mathcal{Y}$, switching parameter $\lambda > 0$, horizon T, initial target $z_1 \in J_1$
Initialize SOGD [12, Algorithm 4] on $\mathcal{Y}$, with all OGD experts initialized at z_1;
for $t = 1, 2, \dots, T$ **do**
  Observe x_t and define $J_t = \mathcal{Y} \cap \{y \in \mathbb{R}^n \mid y \succeq x_t\}$;
  Let z_t be the current target produced by SOGD and implement $y_t = \mathrm{Proj}_{J_t}(z_t)$;
  Observe $g_t \in \partial \ell_t(y_t)$ and the next state x_{t+1};
  Feed the linear loss $f_t(z) = \langle g_t, z\rangle$ to SOGD and receive z_{t+1};
end

We first state how the two guarantees yield inventory regret. Here $\mathcal{S}_T(u_{1:T})$ bounds the base learner's regret plus movement costs.

Theorem F.5. *Assume Assumption 2.1, the inventory dynamics constraint* (2)*, and windowed clearing (Assumption E.1). Suppose the base learner produces targets $z_t \in \mathcal{Y}$, with $z_1 \in J_1$, such that*

$$\sum_{r=t}^{t+B-1} \|z_{r+1} - z_r\|_2 \le \rho \qquad \textit{for every } t \in [T-B].$$

Suppose also that, for every comparator sequence $u_{1:T} \in \mathcal{Y}^T$,

$$\sum_{t=1}^{T} \langle g_t, z_t - u_t\rangle + \Lambda \sum_{t=1}^{T-1} \|z_{t+1} - z_t\|_2 \le \mathcal{S}_T(u_{1:T}), \tag{32}$$

for some $\Lambda \ge GB$. Then

$$R_T(u_{1:T}) \le \mathcal{S}_T(u_{1:T}) \qquad \textit{for every } u_{1:T} \in \mathcal{Y}^T.$$

Proof. The movement condition allows us to apply Theorem E.3, giving

$$R_T(u_{1:T}) \le \sum_{t=1}^{T} \langle g_t, z_t - u_t\rangle + GB \sum_{t=1}^{T-1} \|z_{t+1} - z_t\|_2.$$

Since $GB \le \Lambda$, (32) bounds the right-hand side by $\mathcal{S}_T(u_{1:T})$. □

We now record the SOGD parameters and the two bounds needed above. Following Algorithm 4 of Zhang et al. [12], set

$$K := \Big\lfloor \log_2\Big(\frac{T}{32\max\{\lambda,1\}\log T}\Big)\Big\rfloor + 1, \qquad n^{(k)} := T2^{1-k}, \qquad \eta^{(k)} := \frac{D}{G\sqrt{(1+2\lambda)n^{(k)}}}.$$

Run the kth OGD expert with stepsize $\eta^{(k)}$ and use the aggregation rule in that algorithm with parameters $M = 2$ and $Z = 1/T$.

Proposition F.6. *Assume $\lambda \geq 1$ and $T \geq \max\{32\lambda \log T, e\}$. Run Algorithm 3 with the SOGD parameters above. Then*

$$\|z_{t+1} - z_t\|_2 \leq \frac{2D}{\lambda} \qquad \textit{for every } t \in [T-1], \tag{33}$$

and, for every comparator sequence $u_{1:T} \in \mathcal{Y}^T$,

$$\sum_{t=1}^{T} \langle g_t, z_t - u_t\rangle + G\lambda \sum_{t=1}^{T-1} \|z_{t+1} - z_t\|_2 \leq \mathcal{R}_T^{\mathrm{SOGD}}(u_{1:T};\lambda), \tag{34}$$

where $\mathcal{R}_T^{\mathrm{SOGD}}$ is defined in (13)*.*

Proof of Proposition F.6. For the analysis, define the shifted losses

$$\widetilde{f}_t(z) := \langle g_t, z\rangle - \min_{w\in\mathcal{Y}} \langle g_t, w\rangle.$$

These losses are convex, take values in $[0, GD]$, and have gradient norm at most G. SOGD produces the same targets for f_t and $\widetilde{f}_t$: the OGD updates depend only on gradients, and the aggregation updates use differences between the experts' losses plus their movement costs, as in Eq. (28) of Zhang et al. [12]. The common additive shift cancels from these differences. Thus Algorithm 3 need not compute the minimum defining $\widetilde{f}_t$.

Our parameters agree with those in their Algorithm 4, and the horizon condition ensures that the expert grid is nonempty and satisfies the conditions of their Theorem 3. Their Theorem 4, applied on $[1, T]$ to $\widetilde{f}_t$, gives (34), since

$$\widetilde{f}_t(z_t) - \widetilde{f}_t(u_t) = \langle g_t, z_t - u_t\rangle.$$

Here we set $u_{T+1} = u_T$ in their definition of path variation and discard the nonnegative terminal movement cost $G\lambda\|z_{T+1} - z_T\|_2$.

Lemma 2 in their supplementary material proves that every expert and aggregate in Algorithm 4 moves by at most MD/λ. With $M = 2$, this gives (33) for the final aggregate, which produces z_t. □

It remains to choose λ so that the movement bound meets the inventory reduction's requirements. The choice below ensures both $B(2D/\lambda) \leq \rho$ and $G\lambda \geq GB$.

Theorem F.7. *Assume Assumption 2.1, the inventory dynamics constraint* (2)*, and windowed clearing (Assumption E.1). Set*

$$\lambda_B := \Big\lceil B\max\Big\{1, \frac{2D}{\rho}\Big\}\Big\rceil,$$

and assume $T \geq \max\{32\lambda_B \log T, e\}$. Run Algorithm 3 from $z_1 \in J_1$ with switching parameter λ_B. Then, for every comparator sequence $u_{1:T} \in \mathcal{Y}^T$,

$$R_T(u_{1:T}) \leq \mathcal{R}_T^{\mathrm{SOGD}}(u_{1:T};\lambda_B). \tag{35}$$

Proof. Proposition F.6 applies because $\lambda_B \geq B \geq 1$. Its movement bound gives

$$\sum_{r=t}^{t+B-1} \|z_{r+1} - z_r\|_2 \leq \frac{2DB}{\lambda_B} \leq \rho \qquad \text{for every } t \in [T-B].$$

Since $G\lambda_B \geq GB$, Theorem F.5 applies with $\Lambda = G\lambda_B$ and $\mathcal{S}_T = \mathcal{R}_T^{\mathrm{SOGD}}(\cdot;\lambda_B)$, proving (35). □

UPPD supplies the clearing windows needed for this theorem.

Proof of Theorem 4.5. By Lemma E.4, windowed clearing holds with $B = B_\delta$ with probability at least $1 - \delta$. On this event, Theorem F.7 applies with $\lambda_B = \lambda_\delta$ and gives (14) for every comparator sequence. For $\lambda \geq 1$ and $T \geq e$, (13) gives

$$\mathcal{R}_T^{\mathrm{SOGD}}(u_{1:T}; \lambda) = O\left(GD\sqrt{\lambda T\left(1 + \frac{P_{T,2}}{D}\right)\log T} \right).$$

Substituting $\lambda_\delta = \Theta\big(\mu^{-1}\log(T/\delta)\max\{1, D/\rho\}\big)$ yields the stated asymptotic bound. □

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: In both the introduction and the abstract, we discuss the new theoretical regret bounds and provide background on our queue-based reduction method.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The introduction explains how dimension can enter through problem parameters. Appendix D.2 and the conclusion discuss UPPD's limitations for large assortments with nearly independent intermittent demand and identify local clearing conditions as an open direction.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The assumptions are stated or referenced with the results, and complete proofs are included in the appendix. The main text explains the queue construction and the principal proof ideas.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: In Section 5 and Appendix B, we describe the full setting for each experiment, how we computed the relevant quantities, and what quantities we're plotting. For experiments involving algorithms not formally described in this paper, we cite the relevant paper for a description of those algorithms.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: The algorithm (HT-OGD) is fully specified in pseudocode in Algorithm 2 and the experimental setup is described in detail in both Section 5 and Appendix B, allowing reproduction. We will release the codebase publicly via Github.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5 summarizes the experimental settings, and Appendix B describes the demand processes, capacity sets, loss parameters, stepsizes and tuning, time horizons, and numbers of runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The stochastic experiments report mean regret with ± 2 standard-error bars across runs. The dynamic-regret experiment in Figure 2c is deterministic, so we do not report error bars for it.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix B reports the hardware and execution times: all experiments ran on an Apple Mac Studio with an M4 Max chip and 48GB of memory, and each experiment finished in under an hour.

   Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: We make a theoretical contribution and perform experiments on synthetic demand sequences whose construction is described in the text, so concerns about human subjects and data related concerns do not apply. We do not anticipate any social harm as a result of our work.

Guidelines:

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

10. **Broader impacts**

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

Answer: [N/A]

Justification: This paper makes theoretical contributions to online convex optimization and multi-product inventory control, with experiments on synthetic demand sequences. The work does not propose a system, model, or dataset for deployment, and we do not identify a direct path from our regret-bound analysis to specific societal applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: The paper does not release any data or models. Our contribution is theoretical with experimental demonstrations, so safeguards for high-risk releases do not apply.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [N/A]

    Justification: The paper does not use existing datasets, pre-trained models, or third-party research code packages. Experiments are run on synthetic demand sequences whose construction is described in the paper, implemented using standard scientific computing libraries.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release any new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: The use of ChatGPT and Claude for editorial assistance, proof simplification, and coding is disclosed in the acknowledgments.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.