Online Inverse Linear Optimization: Efficient Logarithmic-Regret Algorithm, Robustness to Suboptimality, and Lower Bound

Shinsaku Sakaue[*]
CyberAgent
Tokyo, Japan
shinsaku.sakaue@gmail.com

Taira Tsuchiya
The University of Tokyo and RIKEN AIP
Tokyo, Japan
tsuchiya@mist.i.u-tokyo.ac.jp

Han Bao[*]
The Institute of Statistical Mathematics
Tokyo, Japan
bao.han@ism.ac.jp

Taihei Oki
Hokkaido University
Hokkaido, Japan
oki@icredd.hokudai.ac.jp

Abstract

In online inverse linear optimization, a learner observes time-varying sets of feasible actions and an agent's optimal actions, selected by solving linear optimization over the feasible actions. The learner sequentially makes predictions of the agent's true linear objective function, and their quality is measured by the *regret*, the cumulative gap between optimal objective values and those achieved by following the learner's predictions. A seminal work by Bärmann et al. (2017) obtained a regret bound of $O(\sqrt{T})$, where T is the time horizon. Subsequently, the regret bound has been improved to $O(n^4 \ln T)$ by Besbes et al. (2021, 2025) and to $O(n \ln T)$ by Gollapudi et al. (2021), where n is the dimension of the ambient space of objective vectors. However, these logarithmic-regret methods are highly inefficient when T is large, as they need to maintain regions specified by $O(T)$ constraints, which represent possible locations of the true objective vector. In this paper, we present the first logarithmic-regret method whose per-round complexity is independent of T; indeed, it achieves the best-known bound of $O(n \ln T)$. Our method is strikingly simple: it applies the online Newton step (ONS) to appropriate exp-concave loss functions. Moreover, for the case where the agent's actions are possibly suboptimal, we establish a regret bound of $O(n \ln T + \sqrt{\Delta_T n \ln T})$, where Δ_T is the cumulative suboptimality of the agent's actions. This bound is achieved by using MetaGrad, which runs ONS with $\Theta(\ln T)$ different learning rates in parallel. We also present a lower bound of $\Omega(n)$, showing that the $O(n \ln T)$ bound is tight up to an $O(\ln T)$ factor.

1 Introduction

Optimization problems serve as forward models of various processes and systems, ranging from human decision-making to natural phenomena. In real-world applications, the true objective function of such models is rarely known a priori. This motivates the problem of inferring the objective function from observed optimal solutions, or *inverse optimization*. Early work in this area emerged from geophysics, aiming at estimating subsurface structure from seismic wave data [11, 53]. Subsequently, inverse optimization has been extensively studied [2, 13, 14, 28], applied across various domains, such as

[*]This work was primarily conducted during the period when SS was affiliated with the University of Tokyo and RIKEN AIP, and HB with Kyoto University and OIST.

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

transportation [6], power systems [9], and healthcare [12], and have laid the foundation for various machine learning methods, including inverse reinforcement learning [44] and contrastive learning [50].

This study focuses on an elementary yet fundamental case where the objective function of forward optimization is linear. We consider an *agent* who repeatedly selects an action from a set of feasible actions by solving forward linear optimization.[1] Let n be a positive integer and $\mathbb{R}^n$ the ambient space where forward optimization is defined. For $t = 1, \dots, T$, given a set $X_t \subseteq \mathbb{R}^n$ of feasible actions, the agent selects an action $x_t \in X_t$ that maximizes $x \mapsto \langle c^*, x \rangle$ over X_t, where $c^* \in \mathbb{R}^n$ is the agent's internal objective vector and $\langle \cdot, \cdot \rangle$ denotes the standard inner product on $\mathbb{R}^n$. We want to infer c^* from observations consisting of the feasible sets and the agent's optimal actions, i.e., $\{(X_t, x_t)\}_{t=1}^T$.

For this problem, Bärmann et al. [4, 5] have shown that online learning methods are effective for inferring the agent's underlying objective vector c^*. Consider a *learner* who aims to infer c^*. For $t = 1, \dots, T$, the learner makes a prediction $\hat{c}_t$ of c^* based on the past observations $\{(X_i, x_i)\}_{i=1}^{t-1}$ and receives (X_t, x_t) as feedback. Let $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$ represent an optimal action induced by the learner's tth prediction. The *regret* of choosing $\hat{x}_t$ instead of x_t is defined as $\sum_{t=1}^T \langle c^*, x_t - \hat{x}_t \rangle$.[2] Their idea is to regard $\mathbb{R}^n \ni c \mapsto \langle c, \hat{x}_t - x_t \rangle$ as a cost function and apply online learning methods, such as the online gradient descent (OGD). Then, $\sum_{t=1}^T \langle c^*, x_t - \hat{x}_t \rangle \leq \sum_{t=1}^T \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle = O(\sqrt{T})$ follows from the standard guarantee of online learning methods. As such, online learning methods with sublinear regret bounds can make the average regret converge to zero as $T \to \infty$.

While the regret bound of $O(\sqrt{T})$ is optimal in general online linear optimization (e.g., Hazan [26, Section 3.2]), the above online inverse linear optimization has special problem structures that could allow for better regret bounds; intuitively, feedback (X_t, x_t) is more informative about c^*, which defines the regret, due to the optimality of $x_t \in X_t$ for c^*. Besbes et al. [7, 8] indeed showed that a logarithmic regret bound of $O(n^4 \ln T)$ is possible, and Gollapudi et al. [25] further improved the bound to $O(n \ln T)$.[3] While these methods significantly improve the dependence on T in the regret bounds, their per-round computation cost is prohibitively high when T is large. Specifically, these methods iteratively update (appropriately inflated) regions that indicate possible locations of the true objective vector c^* and set prediction $\hat{c}_t$ to the "center" of the regions (the circumcenter in Besbes et al. [7, 8] and the centroid in Gollapudi et al. [25]). Since those regions are represented by $O(T)$ constraints, their per-round complexity grows polynomially in T, at least in a straightforward implementation. Indeed, Besbes et al. [7, 8] and Gollapudi et al. [25] only claim that their methods run in $\mathrm{poly}(n, T)$ time. This is in stark contrast to the earlier online-learning approach of Bärmann et al. [4, 5], whose per-round complexity is independent of T; however, its $O(\sqrt{T})$-regret bound is much worse in terms of T. Is it then possible to design a logarithmic-regret method whose per-round complexity is independent of T?

1.1 Our contributions

In this paper, we first present an $O(n \ln T)$-regret method whose per-round complexity is independent of T (Theorem 3.1), answering the above question affirmatively. Table 1 summarizes the comparisons of our result with prior work. Our method is very simple: we apply the online Newton step (ONS) [27] to appropriately designed exp-concave loss functions. We believe this simplicity is a strength of our method, which makes it accessible to a wider audience and easier to implement.

We then address more realistic situations where the agent's actions can be suboptimal. We establish a regret bound of $O(n \ln T + \sqrt{\Delta_T n \ln T})$, where Δ_T denotes the cumulative suboptimality of the agent's actions over T rounds (Theorem 4.1). We also apply this result to the offline setting via the online-to-batch conversion (Corollary 4.2). This bound is achieved by applying MetaGrad [55, 56], a universal online learning method that runs ONS with $\Theta(\ln T)$ different learning rates in parallel, to the *suboptimality loss* [43], a loss function commonly used in inverse optimization. While universal online learning is originally intended to adapt to unknown types of loss functions, our result shows that it is useful for adapting to unknown suboptimality levels in online inverse linear optimization. At

[1] An "agent" is sometimes called an "expert," which we do not use to avoid confusion with the expert in universal online learning (see Section 2.3). Additionally, our results could potentially be extended to nonlinear settings based on kernel inverse optimization [6, 39], although we focus on the linear setting for simplicity.

[2] In the online setting, the learner's goal subtly deviates from inferring c^* directly. Instead, the learner aims to make predictions $\hat{c}_t$ such that the induced actions $\hat{x}_t$ are good for the true objective c^*.

[3] Gollapudi et al. [25] studied the same problem under the name of *contextual recommendation*.

Table 1: Comparisons of the regret bound under optimal feedback and per-round/total complexity. Here, τ_{solve} is the time for computing $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$, and $\tau_{\text{E-proj}}$/$\tau_{\text{G-proj}}$ is the time for the Euclidean/generalized projection; typically, $\tau_{\text{E-proj}} = O(n)$ and $\tau_{\text{G-proj}} = O(n^3)$ (see Section 3 and Appendix A for details). Regarding the regret bound of Bärmann et al. [4, 5], the dependence on n varies depending on the problem setting, which we discuss in Appendix A (here, set sizes are regarded as constants). Besbes et al. [7, 8] and Gollapudi et al. [25] only claim that the total complexity is $\mathrm{poly}(n, T)$. Our inspection in Appendix A estimates the per-round complexity of Gollapudi et al. [25] as $O(\tau_{\text{solve}} + n^5 T^3)$ or higher.

	Regret bound	Per-round complexity	Total complexity
Bärmann et al. [4, 5]	$O(\sqrt{T})$	$O(\tau_{\text{solve}} + \tau_{\text{E-proj}} + n)$	Per-round $\times\, T$
Besbes et al. [7, 8]	$O(n^4 \ln T)$	Not claimed	$\mathrm{poly}(n, T)$
Gollapudi et al. [25]	$O(n \ln T)$	Not claimed	$\mathrm{poly}(n, T)$
This work (Section 3)	$O(n \ln T)$	$O(\tau_{\text{solve}} + \tau_{\text{G-proj}} + n^2)$	Per-round $\times\, T$

a high level, our important contribution lies in uncovering the deeper connection between inverse optimization and online learning, thereby enabling the former to leverage the powerful toolkit of the latter.

Finally, we present a regret lower bound of $\Omega(n)$ (Theorem 5.1). Thus, the upper bound of $O(n \ln T)$ achieved by the method of Gollapudi et al. [25] and ours is tight up to an $O(\ln T)$ factor. While the proof idea is somewhat straightforward, this lower bound clarifies the optimal dependence on n in the regret bound, thereby resolving a question raised in Besbes et al. [8, Section 7].

1.2 Related work

Classic studies on inverse optimization explored formulations for identifying parameters of forward optimization from a single observation [2, 29]. Recently, data-driven inverse optimization, which is intended to infer parameters of forward optimization from multiple noisy (possibly suboptimal) observations, has drawn significant interest [3, 6, 10, 35, 39, 42, 43, 52, 63]. This body of work has addressed offline settings with other criteria than the regret, which we formally define in (2). The suboptimality loss was introduced by Mohajerin Esfahani et al. [43] in this context.

The line of work by Bärmann et al. [4, 5], Besbes et al. [7, 8], and Gollapudi et al. [25], mentioned in Section 1, is the most relevant to our work; we present the detailed comparisons with them in Appendix A. It is worth mentioning here that Gollapudi et al. [25] obtained an $\exp(O(n \ln n))$-regret bound, in addition to the $O(n \ln T)$ bound in Table 1; therefore, it is possible to achieve a finite regret bound, although the dependence on n is exponential. Recently, Sakaue et al. [49] obtained a finite regret bound by assuming a gap between the optimal and suboptimal objective values. Unlike their work, we do not require such gap assumptions. Online inverse linear optimization can also be viewed as a variant of stochastic linear bandits [1, 18], where noisy objective values are given as feedback, instead of optimal actions. Intuitively, the optimal-action feedback is more informative and allows for the $O(n \ln T)$ regret upper bound, while there is a lower bound of $\Omega(n\sqrt{T})$ in stochastic linear bandits [18, Theorem 3]. Online-learning approaches to other related settings have also been studied [20, 30, 59]; see Besbes et al. [8, Section 1.2] for an extensive discussion on the relation to these studies. Additionally, Chen and Kılınç-Karzan [15] and Sun et al. [51] studied online-learning methods for related settings with different criteria.

ONS [27] is a well-known online convex optimization (OCO) method that achieves a logarithmic regret bound for exp-concave loss functions. While ONS requires the prior knowledge of the exp-concavity, universal online learning methods, including MetaGrad, can automatically adapt to the unknown curvatures of loss functions, such as the strong convexity and exp-concavity [55, 56, 58, 64]. Our strategy for achieving robustness to suboptimal feedback is to combine the regret bound of MetaGrad (Proposition 2.6) with the *self-bounding* technique (see Section 4 for details), which is widely adopted in the online learning literature [23, 60, 66].

Contextual search [38, 48] is a related problem of inferring the value of $\langle c^*, x_t \rangle$ for an underlying vector c^* given context vectors x_t. The method of Gollapudi et al. [25] is based on techniques developed in this context. Robustness to corrupted feedback is also studied in contextual search [36, 46, 47].

However, note that the problem setting is different from ours. Also, the regret in contextual search is defined with optimal choices even under corrupted feedback, and the regret bounds scale linearly with the cumulative corruption level. By contrast, our regret is defined with the agent's possibly suboptimal actions and our regret bound grows only at the rate of $\sqrt{\Delta_T}$ for the cumulative suboptimality Δ_T.

Improving the per-round complexity is crucial. This topic has gathered particular attention in online portfolio selection [17, 34, 57]. There exists a trade-off between the per-round complexity and regret bounds among known methods for this problem, and advancing this frontier is recognized as important research [32, 54, 65]. When it comes to online inverse linear optimization, logarithmic regret bounds had only been achieved by the somewhat inefficient methods of Besbes et al. [7, 8] and Gollapudi et al. [25], while the efficient online-learning approach of Bärmann et al. [4, 5] only enjoys the $O(\sqrt{T})$-regret bound. This background highlights the significance of our efficient $O(n \ln T)$-regret method, which realizes the benefits of both approaches that previously existed in a trade-off relationship.

2 Preliminaries

2.1 Problem setting

We consider an online learning setting with two players, a *learner* and an *agent*. The agent sequentially solves linear optimization problems of the following form for $t = 1, \dots, T$:

$$\text{maximize} \ \ \langle c^*, x \rangle \qquad \text{subject to} \ \ x \in X_t, \tag{1}$$

where c^* is the agent's objective vector, which is unknown to the learner. Every feasible set $X_t \subseteq \mathbb{R}^n$ is non-empty and compact, and the agent's action x_t always belongs to X_t. We assume that the agent's action is optimal for (1), i.e., $x_t \in \arg\max_{x \in X_t} \langle c^*, x \rangle$, except in Section 4, where we discuss the case where x_t can be suboptimal. The set, X_t, is not necessarily convex; we only assume access to an oracle that returns an optimal solution $x \in \arg\max_{x' \in X_t} \langle c, x' \rangle$ for any $c \in \mathbb{R}^n$. If X_t is a polyhedron, any solver for linear programs (LPs) of the form (1) can serve as the oracle. Even if (1) is, for example, an integer LP, we may use empirically efficient solvers, such as Gurobi, to obtain an optimal solution.

The learner sequentially makes a prediction of c^* for $t = 1, \dots, T$. Let $\Theta \subseteq \mathbb{R}^n$ denote a set of linear objective vectors, from which the learner picks predictions. We assume that Θ is a closed convex set and that the true objective vector c^* is contained in Θ. For $t = 1, \dots, T$, the learner outputs a prediction $\hat{c}_t$ of c^* based on past observations $\{(X_i, x_i)\}_{i=1}^{t-1}$ and then receives (X_t, x_t) as feedback from the agent. Let $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$ denote an optimal action induced by the learner's tth prediction.[4] We consider the following two measures of the quality of predictions $\hat{c}_1, \dots, \hat{c}_T \in \Theta$:

$$R_T^{c^*} := \sum_{t=1}^{T} \langle c^*, x_t - \hat{x}_t \rangle \quad \text{and} \quad \tilde{R}_T^{c^*} := R_T^{c^*} + \sum_{t=1}^{T} \langle \hat{c}_t, \hat{x}_t - x_t \rangle = \sum_{t=1}^{T} \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle. \tag{2}$$

Following prior work [7, 8, 25], we call $R_T^{c^*}$ the *regret*, which is the cumulative gap between the optimal objective values and the objective values achieved by following the learner's predictions. Note that we have $\langle c^*, x_t - \hat{x}_t \rangle \geq 0$ as long as x_t is optimal for c^*. While the regret is a natural performance measure, the second one, $\tilde{R}_T^{c^*}$, in (2) is convenient when considering the online-learning approach [4, 5]. We always have $R_T^{c^*} \leq \tilde{R}_T^{c^*}$ since the additional term consisting of $\langle \hat{c}_t, \hat{x}_t - x_t \rangle$ is non-negative due to the optimality of $\hat{x}_t$ for $\hat{c}_t$; intuitively, this term quantifies how well $\hat{c}_t$ explains the agent's choice x_t. Our upper bounds in Theorems 3.1 and 4.1 apply to $\tilde{R}_T^{c^*}$, and our lower bound in Theorem 5.1 applies to $R_T^{c^*}$.

Remark 2.1. The problem setting of Besbes et al. [7, 8] involves *context functions* and *initial knowledge sets*, which might make their setting appear more general than ours. However, it is not difficult to confirm that our methods are applicable to their setting. See Appendix A for details.

2.2 Boundedness assumptions and suboptimality loss

We introduce the following bounds on the sizes of X_t and Θ.

[4] We may break ties, if any, arbitrarily. Our results remain true as long as $\hat{x}_t$ is optimal for $\hat{c}_t$.

Assumption 2.2. *The ℓ_2-diameter of Θ is bounded by $D > 0$, i.e., $\max\{ \|c - c'\|_2 : c, c' \in \Theta \} \leq D$. Similarly, the ℓ_2-diameter of X_t is bounded by $K > 0$ for $t = 1, \dots, T$. Furthermore, there exists $B > 0$ satisfying the following condition:*

$$\max\{ \langle c - c', x - x' \rangle \,:\, c, c' \in \Theta, x, x' \in X_t \} \leq B \quad \text{for } t = 1, \dots, T.$$

Assuming bounds on the diameters is common in the previous studies [4, 5, 7, 8, 25]. We additionally introduce $B > 0$ to measure the sizes of X_t and Θ taking their mutual relationship into account. Note that the choice of $B = DK$ is always valid due to the Cauchy–Schwarz inequality. This quantity is inspired by a semi-norm of gradients used in Van Erven et al. [56] and enables sharper analysis than that conducted by simply setting $B = DK$.

We also define the *suboptimality loss* for later use.

Definition 2.3. For $t = 1, \dots, T$, for any action set X_t and the agent's possibly suboptimal action x_t, the suboptimality loss is defined by $\ell_t(c) \coloneqq \max_{x \in X_t} \langle c, x \rangle - \langle c, x_t \rangle$ for all $c \in \Theta$.

That is, $\ell_t(c)$ is the suboptimality of $x_t \in X_t$ for c. Mohajerin Esfahani et al. [43] introduced this as a loss function that enjoys desirable computational properties in the context of inverse optimization. Specifically, the suboptimality loss is convex, and there is a convenient expression of a subgradient.

Proposition 2.4 (cf. Bärmann et al. [4, Proposition 3.1])**.** *The suboptimality loss, $\ell_t \colon \Theta \to \mathbb{R}$, is convex. Moreover, for any $\hat{c}_t \in \Theta$ and $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$, it holds that $\hat{x}_t - x_t \in \partial \ell_t(\hat{c}_t)$.*

Confirming these properties is not difficult: the convexity is due to the fact that ℓ_t is the pointwise maximum of linear functions $c \mapsto \langle c, x \rangle - \langle c, x_t \rangle$, and the subgradient expression is a consequence of Danskin's theorem [19] (or one can directly prove this as in Bärmann et al. [4, Proposition 3.1]). It is worth mentioning that, as pointed out by Sakaue et al. [49], $\tilde{R}_T^{c^*}$ appears as the linearized upper bound on the regret with respect to the suboptimality loss, i.e., $\sum_{t=1}^T (\ell_t(\hat{c}_t) - \ell_t(c^*)) \leq \sum_{t=1}^T \langle \hat{c}_t - c^*, g_t \rangle = \tilde{R}_T^{c^*}$, where $g_t = \hat{x}_t - x_t \in \partial \ell_t(\hat{c}_t)$. This enables the online-to-batch conversion for the suboptimality loss, as discussed in Section 4.1. Additionally, we have $\tilde{R}_T^{c^*} = R_T^{c^*} + \sum_{t=1}^T \ell_t(\hat{c}_t)$ in (2).

2.3 ONS and MetaGrad

We briefly describe ONS and MetaGrad, based on Hazan [26, Section 4.4] and Van Erven et al. [56], to aid understanding of our methods. Appendix B shows the details for completeness. Readers who wish to proceed directly to our results may skip this subsection, taking Propositions 2.5 and 2.6 as given.

For convenience, we first state a specific form of ONS's $O(n \ln T)$ regret bound, which is later used in MetaGrad and in our analysis. See Algorithm 1 in Appendix B.1 for the pseudocode of ONS.

Proposition 2.5. *Let $\mathcal{W} \subseteq \mathbb{R}^n$ be a closed convex set whose ℓ_2-diameter is at most $W > 0$. Let $w_1, \dots, w_T$ and $g_1, \dots, g_T$ be vectors in $\mathbb{R}^n$ satisfying the following conditions for some $G, H > 0$:*

$$w_t \in \mathcal{W}, \quad \|g_t\|_2 \leq G, \quad \text{and} \quad \max\{ \langle w' - w, g_t \rangle \,:\, w, w' \in \mathcal{W} \} \leq H \quad \text{for } t = 1, \dots, T. \tag{3}$$

Take any $\eta \in \left(0, \frac{1}{5H}\right]$ and define loss functions $f_t^\eta \colon \mathcal{W} \to \mathbb{R}$ for $t = 1, \dots, T$ as follows:

$$f_t^\eta(w) \coloneqq -\eta \langle w_t - w, g_t \rangle + \eta^2 \langle w_t - w, g_t \rangle^2 \quad \text{for any } w \in \mathcal{W}. \tag{4}$$

Let $w_1^\eta, \dots, w_T^\eta \in \mathcal{W}$ be the outputs of ONS applied to $f_1^\eta, \dots, f_T^\eta$. Then, for any $u \in \mathcal{W}$, it holds that

$$\sum_{t=1}^T (f_t^\eta(w_t^\eta) - f_t^\eta(u)) = O\left(n \ln\left(\frac{WGT}{Hn} \right) \right).$$

Next, we describe MetaGrad (see Algorithm 2 in Appendix B.3), which we apply to the following general OCO problem on a closed convex set, $\mathcal{W} \subseteq \mathbb{R}^n$. For $t = 1, \dots, T$, we select $w_t \in \mathcal{W}$ based on information obtained up to the end of round $t - 1$; then, we incur $f_t(w_t)$ and observe a subgradient, $g_t \in \partial f_t(w_t)$, where $f_t \colon \mathcal{W} \to \mathbb{R}$ denotes the tth convex loss function. We assume that $\mathcal{W}$ and g_t for $t = 1, \dots, T$ satisfy the conditions in (3). Our goal is to make the regret with respect to f_t, i.e., $\sum_{t=1}^T (f_t(w_t) - f_t(u))$, as small as possible for any comparator $u \in \mathcal{W}$.

MetaGrad maintains *η-experts*, each of whom is associated with one of $\Theta(\ln T)$ different learning rates $\eta \in \left(0, \frac{1}{5H}\right]$. Each η-expert applies ONS to loss functions f_t^η of the form (4), where $w_t \in \mathcal{W}$

is the tth output of MetaGrad and $g_t \in \partial f_t(w_t)$ is given as feedback. In each round t, given the outputs w_t^η of η-experts (which are computed based on information up to round $t-1$), MetaGrad computes $w_t \in \mathcal{W}$ by aggregating them via the exponentially weighted average (EWA).

For any comparator $u \in \mathcal{W}$, define $\tilde{R}_T^u := \sum_{t=1}^T \langle w_t - u, g_t \rangle$ and $V_T^u := \sum_{t=1}^T \langle w_t - u, g_t \rangle^2$. Since all functions f_t are convex, the regret with respect to f_t, or $\sum_{t=1}^T (f_t(w_t) - f_t(u))$, is bounded by $\tilde{R}_T^u$ from above. Furthermore, from the definition of f_t^η, we can decompose $\tilde{R}_T^u$ as follows:

$$\tilde{R}_T^u = -\frac{\sum_{t=1}^T f_t^\eta(u)}{\eta} + \eta V_T^u = \frac{1}{\eta}\left(\sum_{t=1}^T \big(\overbrace{f_t^\eta(w_t)}^{\text{Zero by (4)}} - f_t^\eta(w_t^\eta)\big) + \sum_{t=1}^T (f_t^\eta(w_t^\eta) - f_t^\eta(u))\right) + \eta V_T^u,$$

which simultaneously holds for all $\eta > 0$. The first summation on the right-hand side, i.e., the regret of EWA compared to w_t^η, is indeed as small as $O(\ln \ln T)$, while Proposition 2.5 ensures that the second summation is $O(n \ln T)$. Thus, the right-hand side is $O\left(\frac{n \ln T}{\eta} + \eta V_T^u\right)$. If we knew the true V_T^u value, we could choose $\eta \simeq \sqrt{n \ln T / V_T^u}$ to achieve $O\big(\sqrt{n \ln T \cdot V_T^u}\big)$. This might seem impossible as we do not know any of u, g_t, and w_t beforehand. However, we can show that at least one of $\Theta(\ln T)$ values of η leads to almost the same regret, eschewing the need for knowing V_T^u. Formally, MetaGrad achieves the following regret bound (cf. Van Erven et al. [56, Corollary 8]).[5]

Proposition 2.6. *Let $\mathcal{W} \subseteq \mathbb{R}^n$ be given as in Proposition 2.5. Let $w_1, \dots, w_T \in \mathcal{W}$ be the outputs of MetaGrad applied to convex loss functions $f_1, \dots, f_T \colon \mathcal{W} \to \mathbb{R}$. Assume that for every $t = 1, \dots, T$, subgradient $g_t \in \partial f_t(w_t)$ satisfies the conditions (3) in Proposition 2.5. Then, it holds that*

$$\tilde{R}_T^u = O\left(\sqrt{n \ln\left(\frac{WGT}{Hn}\right) \cdot V_T^u} + Hn \ln\left(\frac{WGT}{Hn}\right)\right).$$

We outline how this result applies to exp-concave losses. Taking W, G, and H to be constants and ignoring the additive term of $O(n \ln(T/n))$ for simplicity, we have $\tilde{R}_T^u = O(\sqrt{n \ln T \cdot V_T^u})$. If all f_t are α-exp-concave for some $\alpha \le 1/(GW)$, then $f_t(w_t) - f_t(u) \le \langle w_t - u, g_t \rangle - \frac{\alpha}{2}\langle w_t - u, g_t \rangle^2$ holds (e.g., Hazan [26, Lemma 4.3]). Summing this over t and using Proposition 2.6 yield

$$\sum_{t=1}^T (f_t(w_t) - f_t(u)) \le \tilde{R}_T^u - \frac{\alpha}{2} V_T^u = O\left(\sqrt{n \ln T \cdot V_T^u} - \alpha V_T^u\right) \lesssim O\left(\frac{n}{\alpha} \ln T\right),$$

where the last inequality is due to $\sqrt{ax} - bx \le \frac{a}{4b}$ for any $a \ge 0$, $b > 0$, and $x \ge 0$. Remarkably, MetaGrad achieves the $O\big(\frac{n}{\alpha} \ln T\big)$ regret bound without prior knowledge of α, whereas ONS achieves this regret bound by using the α value. Furthermore, even when some f_t are not exp-concave, MetaGrad still enjoys a regret bound of $O(\sqrt{T} \ln \ln T)$ [56, Corollary 8]. As such, MetaGrad can automatically adapt to the unknown curvature of loss functions (at the cost of the negligible $\ln \ln T$ factor), which is the key feature of universal online learning methods.

3 An efficient $O(n \ln T)$-regret method based on ONS

This section presents an efficient logarithmic-regret method for online inverse linear optimization. Our method is remarkably simple: we apply ONS to exp-concave loss functions defined similarly to the η-experts' losses (4) used in MetaGrad. The proof is very short given the ONS's regret bound in Proposition 2.5. Despite this simplicity, we can achieve the regret bound of $O(n \ln T)$, which matches the best-known regret upper bound of Gollapudi et al. [25], with far lower per-round complexity.

Theorem 3.1. *Assume that for every $t = 1, \dots, T$, action $x_t \in X_t$ is optimal for $c^* \in \Theta$. Let $\hat{c}_1, \dots, \hat{c}_T \in \Theta$ be the outputs of ONS applied to loss functions defined as follows for $t = 1, \dots, T$:*

$$\ell_t^\eta(c) := -\eta \langle \hat{c}_t - c, \hat{x}_t - x_t \rangle + \eta^2 \langle \hat{c}_t - c, \hat{x}_t - x_t \rangle^2 \quad \text{for all } c \in \Theta, \tag{5}$$

where $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$ and we set $\eta = \frac{1}{5B}$.[6] Then, for $R_T^{c^*}$ and $\tilde{R}_T^{c^*}$ in (2), it holds that

$$R_T^{c^*} \le \tilde{R}_T^{c^*} = O\left(Bn \ln\left(\frac{DKT}{Bn}\right)\right).$$

[5] In Van Erven et al. [56, Corollary 8], the multiplicative factor of H in the second term and the denominators of Hn in ln are replaced with WG and n, respectively. We modify it to obtain the above bound; see Appendix B.

[6] This is equivalent to MetaGrad with a single $\frac{1}{5B}$-expert applied to the suboptimality losses, $\ell_1, \dots, \ell_T$.

Proof. Consider using Proposition 2.5 in the current setting with $\mathcal{W} = \Theta$, $w_t^\eta = w_t = \hat{c}_t$, $g_t = \hat{x}_t - x_t$, $u = c^*$, $W = D$, $G = K$, and $H = B$. Since the optimality of x_t and $\hat{x}_t$ for c^* and $\hat{c}_t$, respectively, ensures $\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle \geq 0$, we have $\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle^2 \leq B \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle$ due to Assumption 2.2. Therefore, $\tilde{R}_T^{c^*} = \sum_{t=1}^T \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle$ and $V_T^{c^*} \coloneqq \sum_{t=1}^T \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle^2$ satisfy $V_T^{c^*} \leq B \tilde{R}_T^{c^*}$. By using this and Proposition 2.5 with $\eta = \frac{1}{5B}$, for some constant $C_{\mathrm{ONS}} > 0$, it holds that

$$\tilde{R}_T^{c^*} = -\sum_{t=1}^T \frac{\ell_t^\eta(c^*)}{\eta} + \eta V_T^{c^*} \leq \sum_{t=1}^T \frac{\overbrace{\ell_t^\eta(\hat{c}_t)}^{\text{Zero by (5)}} - \ell_t^\eta(c^*)}{\eta} + \eta B \tilde{R}_T^{c^*} \leq 5BC_{\mathrm{ONS}} n \ln\left(\frac{DKT}{Bn}\right) + \frac{\tilde{R}_T^{c^*}}{5},$$

and rearranging the terms yields $\tilde{R}_T^{c^*} = O\big(Bn \ln\big(\frac{DKT}{Bn}\big)\big)$.[7] This also applies to $R_T^{c^*} \leq \tilde{R}_T^{c^*}$. □

Time complexity. We discuss the time complexity of the method. Let τ_{solve} be the time for solving linear optimization to find $\hat{x}_t$ and $\tau_{\text{G-proj}}$ the time for the generalized projection onto Θ used in ONS (see Appendix B.1). In each round t, we compute $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$ in τ_{solve} time; after that, the ONS update takes $O(n^2 + \tau_{\text{G-proj}})$ time. Therefore, it runs in $O\big(\tau_{\text{solve}} + n^2 + \tau_{\text{G-proj}}\big)$ time per round, which is independent of T. If problem (1) is an LP, τ_{solve} equals the time for solving the LP (cf. Cohen et al. [16] and Jiang et al. [33]). Also, $\tau_{\text{G-proj}}$ is often affordable as Θ is usually specified by the learner and hence has a simple structure. For example, if Θ is the unit Euclidean ball, the generalized projection can be computed in $O(n^3)$ time by singular value decomposition (e.g., Mhammedi et al. [41, Section 4.1]). We may also use the quasi-Newton-type method for further efficiency [40].

4 Robustness to suboptimal feedback with MetaGrad

In practice, assuming that the agent's actions are always optimal is unrealistic. This section discusses how to handle suboptimal feedback effectively. Here, we let $x_t \in X_t$ denote an arbitrary action taken by the agent, which the learner observes. Now that x_t may have nothing to do with c^*, we can no longer ensure meaningful bounds on the regret that compares $\hat{x}_t$ with optimal actions. For example, if revealed actions x_t remain all zeros for $t = 1, \dots, T$, we can learn nothing about c^*, and hence the regret that compares $\hat{x}_t$ with optimal actions grows linearly in T in the worst case. Considering this issue, we highlight that the regret, $R_T^{c^*} = \sum_{t=1}^T \langle c^*, x_t - \hat{x}_t \rangle$, used here is defined with the agent's possibly suboptimal actions x_t, not with those optimal for c^*. Small upper bounds on this regret ensure that, if the agent's actions x_t are nearly optimal for c^*, so are $\hat{x}_t$. This regret still satisfies $R_T^{c^*} \leq \tilde{R}_T^{c^*} = \sum_{t=1}^T \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle$ since $\hat{x}_t$ is optimal for $\hat{c}_t$. Additionally, recall that the suboptimality loss, ℓ_t, in Definition 2.3 can be defined for any action $x_t \in X_t$ and that $\ell_t(c^*) = \max_{x \in X_t} \langle c^*, x \rangle - \langle c^*, x_t \rangle \geq 0$ indicates the suboptimality of x_t for c^*. Below, we use $\Delta_T \coloneqq \sum_{t=1}^T \ell_t(c^*)$ to denote the cumulative suboptimality of the agent's actions x_t.

In this setting, it is not difficult to show that ONS used in Theorem 3.1 enjoys a regret bound that scales linearly with Δ_T. However, the linear dependence on Δ_T is not satisfactory, as it results in a regret bound of $O(T)$ even for small suboptimality that persists across all rounds. The following theorem ensures that by applying MetaGrad to the suboptimality losses, we can obtain a regret bound that scales with $\sqrt{\Delta_T}$.

Theorem 4.1. *Let $\hat{c}_1, \dots, \hat{c}_T \in \Theta$ be the outputs of MetaGrad applied to the suboptimality losses, $\ell_1, \dots, \ell_T$, given in Definition 2.3. Let $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$ for $t = 1, \dots, T$. Then, it holds that*

$$R_T^{c^*} \leq \tilde{R}_T^{c^*} = O\left(Bn \ln\left(\frac{DKT}{Bn}\right) + \sqrt{\Delta_T Bn \ln\left(\frac{DKT}{Bn}\right)}\right).$$

Proof. Similar to the proof of Theorem 3.1, we apply Proposition 2.6 with $\mathcal{W} = \Theta$, $w_t = \hat{c}_t$, $g_t = \hat{x}_t - x_t$, $u = c^*$, $W = D$, $G = K$, and $H = B$; in addition, $g_t = \hat{x}_t - x_t \in \partial \ell_t(\hat{c}_t)$ holds due

[7] We may use any η as long as $\eta B < 1$ holds; $\eta = \frac{1}{5B}$ is for consistency with MetaGrad in Appendix B.

to Proposition 2.4. Thus, Proposition 2.6 ensures the following bound for some constant $C_{\mathrm{MG}} > 0$:[8]

$$\tilde{R}_T^{c^*} \leq C_{\mathrm{MG}} \left(\sqrt{n \ln\left(\frac{DKT}{Bn}\right) \cdot V_T^{c^*}} + Bn \ln\left(\frac{DKT}{Bn}\right) \right), \tag{6}$$

where $\tilde{R}_T^{c^*} = \sum_{t=1}^T \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle$ and $V_T^{c^*} = \sum_{t=1}^T \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle^2$. In contrast to the case of Theorem 3.1, $\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle^2 \leq B \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle$ is not ensured since $\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle$ can be negative due to the suboptimality of x_t. Instead, we will show that the following inequality holds:

$$\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle^2 \leq B \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle + 2B\ell_t(c^*). \tag{7}$$

If $\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle \geq 0$, (7) is immediate from $\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle^2 \leq B \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle$ and $\ell_t(c^*) \geq 0$. If $\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle < 0$, $\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle^2 \leq B(-\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle)$ holds. In addition, we have

$$\ell_t(c^*) = \max_{x \in X_t} \langle c^*, x \rangle - \langle c^*, x_t \rangle \geq \langle c^*, \hat{x}_t - x_t \rangle \geq \langle c^*, \hat{x}_t - x_t \rangle - \langle \hat{c}_t, \hat{x}_t - x_t \rangle = -\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle,$$

where the second inequality follows from $\langle \hat{c}_t, \hat{x}_t - x_t \rangle \geq 0$. Multiplying both sides by 2 yields

$$-2\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle \leq 2\ell_t(c^*) \iff -\langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle \leq \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle + 2\ell_t(c^*).$$

Thus, (7) holds in any case, and hence $V_T^{c^*} \leq B\tilde{R}_T^{c^*} + 2B\Delta_T$. Substituting this into (6), we obtain

$$\tilde{R}_T^{c^*} \leq C_{\mathrm{MG}} \left(\sqrt{Bn \ln\left(\frac{DKT}{Bn}\right) \left(\tilde{R}_T^{c^*} + 2\Delta_T\right)} + Bn \ln\left(\frac{DKT}{Bn}\right) \right).$$

We assume $\tilde{R}_T^{c^*} > 0$; otherwise, the trivial bound of $\tilde{R}_T^{c^*} \leq 0$ holds. By the subadditivity of $x \mapsto \sqrt{x}$ for $x \geq 0$, we have $\tilde{R}_T^{c^*} \leq \sqrt{a\tilde{R}_T^{c^*}} + b$, where $a = C_{\mathrm{MG}}^2 Bn \ln\left(\frac{DKT}{Bn}\right)$ and $b = \sqrt{2a\Delta_T} + \frac{a}{C_{\mathrm{MG}}}$. Since $x \leq \sqrt{ax} + b$ implies $x = \frac{4}{3}x - \frac{x}{3} \leq \frac{4}{3}(\sqrt{ax} + b) - \frac{x}{3} = -\frac{1}{3}(\sqrt{x} - 2\sqrt{a})^2 + \frac{4}{3}(a+b) \leq \frac{4}{3}(a+b)$ for any $a, b, x \geq 0$, we obtain $\tilde{R}_T^{c^*} \leq \frac{4}{3}(a+b) = O\left(Bn \ln\left(\frac{DKT}{Bn}\right) + \sqrt{\Delta_T Bn \ln\left(\frac{DKT}{Bn}\right)}\right)$. □

If every x_t is optimal, i.e., $\Delta_T = 0$, the bound recovers that in Theorem 3.1. Note that MetaGrad requires no prior knowledge of Δ_T; it automatically achieves the bound that scales with $\sqrt{\Delta_T}$, analogous to the original bound in Proposition 2.6 that scales with $\sqrt{V_T^u}$. Moreover, a refined version of MetaGrad [56] enables us to achieve a similar bound without prior knowledge of K, B, or T (see Appendix B.4). Universal online learning methods shine in such scenarios where adaptivity to unknown quantities is desired. Another noteworthy point is that the last part of the proof uses the self-bounding technique [23, 60, 66]. Specifically, we derived $\tilde{R}_T^{c^*} \lesssim a + b$ from $\tilde{R}_T^{c^*} \leq \sqrt{a\tilde{R}_T^{c^*}} + b$, where the latter means that $\tilde{R}_T^{c^*}$ is upper bounded by a term of lower order in $\tilde{R}_T^{c^*}$ itself, hence the name self-bounding. We expect that the combination of universal online learning methods and self-bounding, through relations like $V_T^{c^*} \lesssim \tilde{R}_T^{c^*} + \Delta_T$ used above, will be a useful technique for deriving meaningful guarantees in online inverse linear optimization.

Time complexity. The use of MetaGrad comes with a slight increase in time complexity. First, as with the case of ONS, $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$ is computed in each round, taking τ_{solve} time. Then, each η-expert performs the ONS update, taking $O(n^2 + \tau_{\text{G-proj}})$ time. Since MetaGrad maintains $\Theta(\ln T)$ distinct η values, the total per-round time complexity is $O\left(\tau_{\text{solve}} + (n^2 + \tau_{\text{G-proj}}) \ln T\right)$. If the $O(\tau_{\text{G-proj}} \ln T)$ factor is a bottleneck, we can use more efficient universal algorithms [41, 61] to reduce the number of projections from $\Theta(\ln T)$ to 1. Moreover, the $O(n^2)$ factor can also be reduced by sketching techniques (see Van Erven et al. [56, Section 5]).

4.1 Online-to-batch conversion

We briefly discuss the implication of Theorem 4.1 in the offline setting, where feedback follows some underlying distribution. As noted in Section 2.2, the bound in Theorem 4.1 applies to the regret with respect to the suboptimality loss, $\sum_{t=1}^T (\ell_t(\hat{c}_t) - \ell_t(c^*))$, since it is bounded by $\tilde{R}_T^{c^*}$ from above. Therefore, the standard online-to-batch conversion (e.g., Orabona [45, Theorem 3.1]) implies the following convergence of the average prediction in terms of the suboptimality loss.

[8] Here, we can simultaneously achieve $\tilde{R}_T^{c^*} = O(DK\sqrt{T \ln \ln T})$ thanks to MetaGrad's guarantee [56, Corollary 8], which can yield a stronger bound when n is huge.

Corollary 4.2. *For any non-empty and compact $X \subseteq \mathbb{R}^n$, $x \in X$, and $c \in \Theta$, define the corresponding suboptimality loss as $\ell_{X,x}(c) \coloneqq \max_{x' \in X} \langle c, x' \rangle - \langle c, x \rangle$. Let $\Delta > 0$ and define $\mathcal{X}_\Delta$ as the set of observations (X, x) with bounded suboptimality, $\ell_{X,x}(c^*) \leq \Delta$. Assume that $\{(X_t, x_t)\}_{t=1}^T$ are drawn i.i.d. from some distribution on $\mathcal{X}_\Delta$ (hence $\Delta_T \leq \Delta T$). Let $\hat{c}_1, \dots, \hat{c}_T \in \Theta$ be the outputs of MetaGrad applied to the suboptimality losses $\ell_t = \ell_{X_t, x_t}$ for $t = 1, \dots, T$. Then, it holds that*

$$\mathbb{E}\left[\ell_{X,x}\left(\frac{1}{T}\sum_{t=1}^T \hat{c}_t\right) - \ell_{X,x}(c^*)\right] = O\left(\frac{Bn}{T}\ln\left(\frac{DKT}{Bn}\right) + \sqrt{\frac{\Delta Bn}{T}\ln\left(\frac{DKT}{Bn}\right)}\right).$$

Bärmann et al. [4, Theorem 3.14] also obtained a similar offline guarantee via the online-to-batch conversion. Their convergence rate is $O\big(\frac{1}{\sqrt{T}}\big)$ even when $\Delta = 0$, whereas our Corollary 4.2 offers the faster rate of $O\big(\frac{\ln T}{T}\big)$ if $\Delta = 0$. It also applies to the case of $\Delta > 0$, which is important in practice because stochastic feedback is rarely optimal at all times. We emphasize that if regret bounds scale linearly with Δ_T, the above online-to-batch conversion cannot ensure that the excess suboptimality loss (the left-hand side) converge to zero as $T \to 0$. This observation lends support to the importance of the $\sqrt{\Delta_T}$-dependent regret bound we established in Theorem 4.1.

5 $\Omega(n)$ lower bound

We construct an instance where any online learner incurs an $\Omega(n)$ regret, implying that the $O(n \ln T)$ upper bound is tight up to an $O(\ln T)$ factor. More strongly, the following Theorem 5.1 shows that, for any $B > 0$ that gives the tight upper bound in Assumption 2.2, no learner can achieve a regret smaller than $\frac{Bn}{4}$, which means that the Bn factor in our Theorem 3.1 is inevitable.

Theorem 5.1. *Let n be a positive integer and $\Theta = \left[-\frac{1}{\sqrt{n}}, +\frac{1}{\sqrt{n}}\right]^n$. For any $T \geq n$, $B > 0$, and the learner's outputs $\hat{c}_1, \dots, \hat{c}_T \in \Theta$, there exist $c^* \in \Theta$ and $X_1, \dots, X_T \subseteq \mathbb{R}^n$ such that*

$$\max_{t=1,\dots,T} \max\{\, \langle c - c', x - x' \rangle \,:\, c, c' \in \Theta, x, x' \in X_t \,\} = B \quad \text{and} \quad \mathbb{E}\Big[R_T^{c^*}\Big] \geq \frac{Bn}{4}$$

hold, where $R_T^{c^*} = \sum_{t=1}^T \langle c^*, x_t - \hat{x}_t \rangle$, $x_t \in \arg\max_{x \in X_t} \langle c^*, x \rangle$, $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$, and the expectation is taken over the learner's possible randomness.

Proof. We focus on the first n rounds and show that any learner must incur $\frac{Bn}{4}$ in these rounds; in the remaining rounds, we may use any instance since the optimality of x_t for c^* ensures $\langle c^*, x_t - \hat{x}_t \rangle \geq 0$. For $t = 1, \dots, n$, let $X_t = \left\{ x \in \mathbb{R}^n \,:\, -\frac{B}{4}\sqrt{n} \leq x(t) \leq \frac{B}{4}\sqrt{n},\ x(i) = 0 \text{ for } i \neq t \right\}$, where $x(i)$ denotes the ith element of x. That is, X_t is the line segment on the tth axis from $-\frac{B}{4}\sqrt{n}$ to $\frac{B}{4}\sqrt{n}$. Then, $\max\{\, \langle c - c', x - x' \rangle \,:\, c, c' \in \Theta, x, x' \in X_t \,\} = B$ holds for each $t = 1, \dots, n$. Let $c^* \in \Theta$ be a random vector such that each entry is $-\frac{1}{\sqrt{n}}$ or $\frac{1}{\sqrt{n}}$ with probability $\frac{1}{2}$, which is drawn independently of any other randomness. Then, the optimal action, $x_t \in X_t$, which is zero everywhere except that its tth coordinate equals $\frac{c^*(t)}{|c^*(t)|} \cdot \frac{B}{4}\sqrt{n}$, achieves $\langle c^*, x_t \rangle = \frac{B}{4}$. Note that the learner's tth prediction $\hat{c}_t$ is independent of $c^*(t)$ since it depends only on past observations, $\{(X_i, x_i)\}_{i=1}^{t-1}$, which have no information about $c^*(t)$. Thus, $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$ is also independent of $c^*(t)$, and hence

$$\mathbb{E}[\langle c^*, x_t - \hat{x}_t \rangle] = \mathbb{E}[\langle c^*, x_t \rangle] - \mathbb{E}[\langle c^*, \hat{x}_t \rangle] = \frac{B}{4} - \frac{1}{2}\left(-\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{n}}\right)\hat{x}_t(t) = \frac{B}{4},$$

where the expectation is taken over the randomness of c^*. This implies that any deterministic learner incurs $\frac{Bn}{4}$ in the first n rounds in expectation. Thanks to Yao's minimax principle [62], we can conclude that for any randomized learner, there exists $c^* \in \Theta$ such that $\mathbb{E}\big[R_T^{c^*}\big] \geq \frac{Bn}{4}$ holds. □

In the above proof, we restricted $X_1, \dots, X_T$ to line segments so that each $x_t \in \arg\max_{x \in X_t} \langle c^*, x \rangle$ reveals nothing about $c^*(t+1), \dots, c^*(n)$. Whether a similar lower bound holds when all X_t are full-dimensional remains an open question. Another side note is that the $\Omega(n)$ lower bound does not contradict the $O(\sqrt{T})$ upper bound of Bärmann et al. [4]. Their OGD-based method indeed achieves a regret bound of $O(DK\sqrt{T})$, where D and K are upper bounds on the ℓ_2-diameters of Θ and X_t, respectively. In the above proof, $T \geq n$, $D \geq 1$, and $K \geq \frac{B}{2}\sqrt{n}$ hold, implying that their regret upper bound is $DK\sqrt{T} \gtrsim Bn$. Hence, the $\Omega(n)$-lower bound and their $O(DK\sqrt{T})$-upper bound are compatible.

6 Conclusion and discussion

We have presented an efficient ONS-based method that achieves an $O(n \ln T)$-regret bound for online inverse linear optimization. Then, we have extended the method to deal with suboptimal feedback based on MetaGrad, achieving an $O(n \ln T + \sqrt{\Delta_T n \ln T})$-regret bound, where Δ_T is the cumulative suboptimality of the agent's actions. Finally, we have presented a lower bound of $\Omega(n)$, which shows that the $O(n \ln T)$ upper bound is tight up to an $O(\ln T)$ factor. Regarding limitations, our work is restricted to the case where the agent's optimization problem is linear, as mentioned in Footnote 1; how to deal with non-linearity is an important direction for future work. In online portfolio selection, ONS is efficient but inferior to the universal portfolio algorithm regarding the dependence on the gradient norm [57]. Exploring possible similar relationships in online inverse linear optimization is left for future work. Last but not least, closing the $O(\ln T)$ gap between the upper and lower bounds is an important open problem. Interestingly, if all X_t are line segments as in Section 5 and the learner can observe X_t in the beginning of round t, the algorithm of Gollapudi et al. [25, Theorem 5.2] offers a regret upper bound of $O(n^5 \log^2 n)$, which is finite and polynomial in n. We also provide an additional discussion on a finite regret bound for the case of $n = 2$ in Appendix C.

Acknowledgments and Disclosure of Funding

SS was supported by JST ERATO Grant Number JPMJER1903. TT was supported by JST ACT-X Grant Number JPMJAX210E and JSPS KAKENHI Grant Number JP24K23852. HB was supported by JST PRESTO Grant Number JPMJPR24K6. TO was supported by JST FOREST Grant Number JPMJFR232L and JSPS KAKENHI Grant Number JP22K17853.

References

[1] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 24, pages 2312–2320. Curran Associates, Inc., 2011 (cited on page 3).

[2] R. K. Ahuja and J. B. Orlin. Inverse optimization. *Operations Research*, 49(5):771–783, 2001 (cited on pages 1, 3).

[3] A. Aswani, Z.-J. (Max) Shen, and A. Siddiq. Inverse optimization with noisy data. *Operations Research*, 66(3):870–892, 2018 (cited on page 3).

[4] A. Bärmann, A. Martin, S. Pokutta, and O. Schneider. An online-learning approach to inverse optimization. *arXiv:1810.12997*, 2020 (cited on pages 2–5, 9, 21, 28).

[5] A. Bärmann, S. Pokutta, and O. Schneider. Emulating the expert: Inverse optimization through online learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 400–410. PMLR, 2017 (cited on pages 2–5, 21).

[6] D. Bertsimas, V. Gupta, and I. C. Paschalidis. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153(2):595–633, 2015 (cited on pages 2, 3).

[7] O. Besbes, Y. Fonseca, and I. Lobel. Online learning from optimal actions. In *Proceedings of the 34th Conference on Learning Theory*, volume 134, pages 586–586. PMLR, 2021 (cited on pages 2–5, 21, 26).

[8] O. Besbes, Y. Fonseca, and I. Lobel. Contextual inverse optimization: Offline and online learning. *Operations Research*, 73(1):424–443, 2025 (cited on pages 2–5, 21, 26–28).

[9] J. R. Birge, A. Hortaçsu, and J. M. Pavlin. Inverse optimization for the recovery of market structure from market outcomes: An application to the MISO electricity market. *Operations Research*, 65(4):837–855, 2017 (cited on page 2).

[10] J. R. Birge, X. Li, and C. Sun. Learning from stochastically revealed preference. In *Advances in Neural Information Processing Systems*, volume 35, pages 35061–35071. Curran Associates, Inc., 2022 (cited on page 3).

[11] D. Burton and P. L. Toint. On an instance of the inverse shortest paths problem. *Mathematical Programming*, 53(1):45–61, 1992 (cited on page 1).

[12] T. C. Y. Chan, M. Eberg, K. Forster, C. Holloway, L. Ieraci, Y. Shalaby, and N. Yousefi. An inverse optimization approach to measuring clinical pathway concordance. *Management Science*, 68(3):1882–1903, 2022 (cited on page 2).

[13] T. C. Y. Chan, T. Lee, and D. Terekhov. Inverse optimization: Closed-form solutions, geometry, and goodness of fit. *Management Science*, 65(3):1115–1135, 2019 (cited on page 1).

[14] T. C. Y. Chan, R. Mahmood, and I. Y. Zhu. Inverse optimization: Theory and applications. *Operations Research*, 73(2):1046–1074, 2025 (cited on page 1).

[15] V. X. Chen and F. Kılınç-Karzan. Online convex optimization perspective for learning from dynamically revealed preferences. *arXiv:2008.10460*, 2020 (cited on page 3).

[16] M. B. Cohen, Y. T. Lee, and Z. Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM*, 68(1):1–39, 2021 (cited on page 7).

[17] T. M. Cover and E. Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363, 1996 (cited on page 4).

[18] V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Conference on Learning Theory*, pages 355–366. PMLR, 2008 (cited on page 3).

[19] J. M. Danskin. The theory of max-min, with applications. *SIAM Journal on Applied Mathematics*, 14(4):641–664, 1966 (cited on page 5).

[20] C. Dong, Y. Chen, and B. Zeng. Generalized inverse optimization through online learning. In *Advances in Neural Information Processing Systems*, volume 31, pages 86–95. Curran Associates, Inc., 2018 (cited on page 3).

[21] V. Feldman, C. Guzman, and S. Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. *arXiv:1512.09170*, 2015 (cited on page 22).

[22] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956 (cited on page 22).

[23] P. Gaillard, G. Stoltz, and T. van Erven. A second-order bound with excess losses. In *Proceedings of the 27th Conference on Learning Theory*, volume 35, pages 176–196. PMLR, 2014 (cited on pages 3, 8).

[24] D. Garber and N. Wolf. Frank–Wolfe with a nearest extreme point oracle. In *Proceedings of the 34th Conference on Learning Theory*, volume 134, pages 2103–2132. PMLR, 2021 (cited on page 22).

[25] S. Gollapudi, G. Guruganesh, K. Kollias, P. Manurangsi, R. Paes Leme, and J. Schneider. Contextual recommendations and low-regret cutting-plane algorithms. In *Advances in Neural Information Processing Systems*, volume 34, pages 22498–22508. Curran Associates, Inc., 2021 (cited on pages 2–6, 10, 21, 22, 26, 28).

[26] E. Hazan. Introduction to online convex optimization. *arXiv:1909.05207*, 2023. https://arxiv.org/abs/1909.05207v3 (cited on pages 2, 5, 6, 22–24).

[27] E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2):169–192, 2007 (cited on pages 2, 3).

[28] C. Heuberger. Inverse combinatorial optimization: A survey on problems, methods, and results. *Journal of Combinatorial Optimization*, 8(3):329–361, 2004 (cited on page 1).

[29] G. Iyengar and W. Kang. Inverse conic programming with applications. *Operations Research Letters*, 33(3):319–330, 2005 (cited on page 3).

[30] S. Jabbari, R. M. Rogers, A. Roth, and S. Z. Wu. Learning from rational behavior: Predicting solutions to unknown linear programs. In *Advances in Neural Information Processing Systems*, volume 29, pages 1570–1578. Curran Associates, Inc., 2016 (cited on page 3).

[31] M. Jaggi. Revisiting Frank–Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 427–435. PMLR, 2013 (cited on page 22).

[32] R. Jézéquel, D. M. Ostrovskii, and P. Gaillard. Efficient and near-optimal online portfolio selection. *arXiv:2209.13932*, 2022 (cited on page 4).

[33] S. Jiang, Z. Song, O. Weinstein, and H. Zhang. A faster algorithm for solving general LPs. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 823–832. ACM, 2021 (cited on page 7).

[34] A. Kalai and S. Vempala. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, 3(Nov):423–440, 2002 (cited on page 4).

[35] A. Keshavarz, Y. Wang, and S. Boyd. Imputing a convex objective function. In *Proceedings of the 2011 IEEE International Symposium on Intelligent Control*, pages 613–619. IEEE, 2011 (cited on page 3).

[36] A. Krishnamurthy, T. Lykouris, C. Podimata, and R. Schapire. Contextual search in the presence of irrational agents. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 910–918. ACM, 2021 (cited on page 3).

[37] S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank–Wolfe optimization variants. In *Advances in Neural Information Processing Systems*, volume 28, pages 496–504. Curran Associates, Inc., 2015 (cited on page 22).

[38] A. Liu, R. Paes Leme, and J. Schneider. Optimal contextual pricing and extensions. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms*, pages 1059–1078. SIAM, 2021 (cited on page 3).

[39] Y. Long, T. Ok, P. Zattoni Scroccaro, and P. Mohajerin Esfahani. Scalable kernel inverse optimization. In *Advances in Neural Information Processing Systems*, volume 37, pages 99464–99487. Curran Associates, Inc., 2024 (cited on pages 2, 3).

[40] Z. Mhammedi and K. Gatmiry. Quasi-Newton steps for efficient online exp-concave optimization. In *Proceedings of the 36th Conference on Learning Theory*, volume 195, pages 4473–4503. PMLR, 2023 (cited on page 7).

[41] Z. Mhammedi, W. M. Koolen, and T. van Erven. Lipschitz adaptivity with multiple learning rates in online learning. In *Proceedings of the 32nd Conference on Learning Theory*, volume 99, pages 2490–2511. PMLR, 2019 (cited on pages 7, 8, 26).

[42] S. K. Mishra, A. Raj, and S. Vaswani. From inverse optimization to feasibility to ERM. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 35805–35828. PMLR, 2024 (cited on page 3).

[43] P. Mohajerin Esfahani, S. Shafieezadeh-Abadeh, G. A. Hanasusanto, and D. Kuhn. Data-driven inverse optimization with imperfect information. *Mathematical Programming*, 167:191–234, 2018 (cited on pages 2, 3, 5).

[44] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 663–670. Morgan Kaufmann Publishers Inc., 2000 (cited on page 2).

[45] F. Orabona. A modern introduction to online learning. *arXiv:1912.13213*, 2023. https://arxiv.org/abs/1912.13213v6 (cited on page 8).

[46] R. Paes Leme, C. Podimata, and J. Schneider. Corruption-robust contextual search through density updates. In *Proceedings of the 35th Conference on Learning Theory*, volume 178, pages 3504–3505. PMLR, 2022 (cited on page 3).

[47] R. Paes Leme, C. Podimata, and J. Schneider. Density-based algorithms for corruption-robust contextual search and convex optimization. *arXiv:2206.07528*, 2025 (cited on page 3).

[48] R. Paes Leme and J. Schneider. Contextual search via intrinsic volumes. *SIAM Journal on Computing*, 51(4):1096–1125, 2022 (cited on page 3).

[49] S. Sakaue, H. Bao, and T. Tsuchiya. Revisiting online learning approach to inverse linear optimization: A Fenchel–Young loss perspective and gap-dependent regret analysis. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258, pages 46–54. PMLR, 2025 (cited on pages 3, 5).

[50] L. Shi, G. Zhang, H. Zhen, J. Fan, and J. Yan. Understanding and generalizing contrastive learning from the inverse optimal transport perspective. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 31408–31421. PMLR, 2023 (cited on page 2).

[51] C. Sun, S. Liu, and X. Li. Maximum optimality margin: A unified approach for contextual linear programming and inverse linear programming. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 32886–32912. PMLR, 2023 (cited on page 3).

[52] Y. Tan, D. Terekhov, and A. Delong. Learning linear programs from optimal decisions. In *Advances in Neural Information Processing Systems*, volume 33, pages 19738–19749. Curran Associates, Inc., 2020 (cited on page 3).

[53] A. Tarantola. Inverse problem theory: Methods for data fitting and model parameter estimation. *Geophysical Journal International*, 94(1):167–167, 1988 (cited on page 1).

[54] C.-E. Tsai, H.-C. Cheng, and Y.-H. Li. Online self-concordant and relatively smooth minimization, with applications to online portfolio selection and learning quantum states. In *Proceedings of the 34th International Conference on Algorithmic Learning Theory*, volume 201, pages 1481–1483. PMLR, 2023 (cited on page 4).

[55] T. van Erven and W. M. Koolen. MetaGrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems*, volume 29, pages 3666–3674. Curran Associates, Inc., 2016 (cited on pages 2, 3, 25).

[56] T. van Erven, W. M. Koolen, and D. van der Hoeven. MetaGrad: Adaptation using multiple learning rates in online learning. *Journal of Machine Learning Research*, 22(161):1–61, 2021 (cited on pages 2, 3, 5, 6, 8, 22, 26).

[57] T. van Erven, D. van der Hoeven, W. Kotłowski, and W. M. Koolen. Open problem: Fast and optimal online portfolio selection. In *Proceedings of the 33rd Conference on Learning Theory*, volume 125, pages 3864–3869. PMLR, 2020 (cited on pages 4, 10).

[58] G. Wang, S. Lu, and L. Zhang. Adaptivity and optimality: A universal algorithm for online convex optimization. In *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference*, volume 115, pages 659–668. PMLR, 2020 (cited on pages 3, 25).

[59] A. Ward, N. Master, and N. Bambos. Learning to emulate an expert projective cone scheduler. In *Proceedings of the 2019 American Control Conference*, pages 292–297. IEEE, 2019 (cited on page 3).

[60] C.-Y. Wei and H. Luo. More adaptive algorithms for adversarial bandits. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pages 1263–1291. PMLR, 2018 (cited on pages 3, 8).

[61] W. Yang, Y. Wang, P. Zhao, and L. Zhang. Universal online convex optimization with 1 projection per round. In *Advances in Neural Information Processing Systems*, volume 37, pages 31438–31472. Curran Associates, Inc., 2024 (cited on page 8).

[62] A. C.-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 222–227. IEEE, 1977 (cited on page 9).

[63] P. Zattoni Scroccaro, B. Atasoy, and P. Mohajerin Esfahani. Learning in inverse optimization: Incenter cost, augmented suboptimality loss, and algorithms. *Operations Research*, 0(0):1–19, 2024 (cited on page 3).

[64] L. Zhang, G. Wang, J. Yi, and T. Yang. A simple yet universal strategy for online convex optimization. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 26605–26623. PMLR, 2022 (cited on page 3).

[65] J. Zimmert, N. Agarwal, and S. Kale. Pushing the efficiency-regret Pareto frontier for online learning of portfolios and quantum states. In *Proceedings of the 35th Conference on Learning Theory*, volume 178, pages 182–226. PMLR, 2022 (cited on page 4).

[66] J. Zimmert and Y. Seldin. Tsallis-INF: An optimal algorithm for stochastic and adversarial bandits. *Journal of Machine Learning Research*, 22(28):1–49, 2021 (cited on pages 3, 8).

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: See Abstract, Section 1, and Section 1.1.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Section 2.2 for assumptions, Sections 3 and 4 for the computational complexity, Section 5 for limitations regarding the lower bound, and Section 6 for additional discussions.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: See Section 2.2 for assumptions. All theorems are followed by proofs.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Appendix D provides a detailed description of our preliminary experiments.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The source code and its readme file are available at https://github.com/ssakaue/online-inverse-linear-optimization-code.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: See Appendix D.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Table 2 reports the results with the mean and standard deviation.

Guidelines:

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

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Appendix D.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification:

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [NA]

    Justification: This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [NA]

    Justification:

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification:

    Guidelines:

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

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

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

    Answer: [NA]

    Justification:

    Guidelines:

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

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

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

    Answer: [NA]

    Justification:

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification:

Guidelines:

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

A Detailed comparisons with previous results

Below we compare our results with Bärmann et al. [4, 5], Besbes et al. [7, 8], and Gollapudi et al. [25].

Bärmann et al. [4, 5] used $\tilde{R}_T^{c^*}$ as the performance measure, as with our Theorems 3.1 and 4.1, and provided two specific methods. The first one, based on the multiplicative weights update (MWU), is tailored for the case where Θ is the probability simplex, i.e., $\Theta = \{c \in \mathbb{R}^n \mid c \geq 0, \|c\|_1 = 1\}$. The authors assumed a bound of $K_\infty > 0$ on the ℓ_∞-diameters of X_t and obtained a regret bound of $O(K_\infty \sqrt{T \ln n})$. The second one is based on the online gradient descent (OGD) and applies to general convex sets Θ. The authors assumed that the ℓ_2-diameters of Θ and X_t are bounded by $D > 0$ and $K > 0$, respectively, and obtained a regret bound of $O(DK\sqrt{T})$. In the first case, our Theorem 3.1 with $B = K_\infty$, $D = \sqrt{2}$, and $K \leq 2\sqrt{n}K_\infty$ offers a bound of $O(K_\infty n \ln(T/\sqrt{n}))$; in the second case, we obtain a bound of $O(DKn \ln(T/n))$ by setting $B = DK$. In both cases, our bounds improve the dependence on T from $\sqrt{T}$ to $\ln T$, while scaled up by a factor of n, up to logarithmic terms. Regarding the computation time, their MWU and OGD methods run in $O(\tau_{\text{solve}} + \tau_{\text{E-proj}} + n)$ time per round, where $\tau_{\text{E-proj}}$ is the time for the Euclidean projection onto Θ, hence faster than our method. Also, suboptimal feedback is discussed in Bärmann et al. [4, Sections 3.1]. However, their bound does not achieve the logarithmic dependence on T even when $\Delta_T = 0$, unlike our Theorem 4.1.

Besbes et al. [7, 8] used $R_T^{c^*}$ as the performance measure, which is upper bounded by $\tilde{R}_T^{c^*}$. They assumed that c^* lies in the unit Euclidean sphere and that the ℓ_2-diameters of X_t are at most one. Under these conditions, they obtained the first logarithmic regret bound of $O(n^4 \ln T)$. By applying Theorem 3.1 to this case, we obtain a bound of $O(n \ln(T/n))$, which is better than their bound by a factor of n^3. As discussed in Section 1, their method relies on the idea of narrowing down regions represented with $O(T)$ constraints, and hence it seems inefficient for large T; indeed Besbes et al. [8, Theorem 4] only claims that the total time complexity is polynomial in n and T. Considering this, our ONS-based method is arguably much faster while achieving the better regret bound.

On the problem setting of Besbes et al. [7, 8]. As mentioned in Remark 2.1, the problem setting of Besbes et al. [7, 8] is seemingly different from ours. In their setting, in each round t, the learner first observes (X_t, f_t), where $f_t \colon X_t \to \mathbb{R}^n$ is called a *context function*. Then, the learner chooses $\hat{x}_t \in X_t$ and receives an optimal action $x_t \in \arg\max_{x \in X_t} \langle c^*, f_t(x) \rangle$ as feedback. It is assumed that the learner can solve $\max_{x \in X_t} \langle c, f_t(x) \rangle$ for any $c \in \mathbb{R}^n$ and that all f_t are 1-Lipschitz, i.e., $\|f_t(x) - f_t(x')\|_2 \leq \|x - x'\|_2$ for all $x, x' \in X_t$. We note that our methods work in this setting, while the presence of f_t might make their setting appear more general. Specifically, we redefine X_t as the image of f_t, i.e., $\{ f_t(x) : x \in X_t \}$. Then, their assumption ensures that we can find $f_t(\hat{x}_t) \in X_t$ that maximizes $X_t \ni \xi \mapsto \langle \hat{c}_t, \xi \rangle$, and the ℓ_2-diameter of the newly defined X_t is bounded by 1 due to the 1-Lipschitzness of f_t. Therefore, by defining $g_t = f_t(\hat{x}_t) - f_t(x_t)$ and applying it in Theorems 3.1 and 4.1, we recover the bounds therein on $\sum_{t=1}^T \langle \hat{c}_t - c^*, f_t(\hat{x}_t) - f_t(x_t) \rangle$, with D, K, and B being constants. The bounds also apply to the regret, $\sum_{t=1}^T \langle c^*, f_t(x_t) - f_t(\hat{x}_t) \rangle$, used in Besbes et al. [7, 8]. Additionally, Besbes et al. [7, 8] consider a (possibly non-convex) initial knowledge set $C_0 \subseteq \mathbb{R}^n$ that contains c^*. We note, however, that they do not care about whether predictions $\hat{c}_t$ lie in C_0 or not since the regret, their performance measure, does not explicitly involve $\hat{c}_t$. Indeed, predictions $\hat{c}_t$ that appear in their method are chosen from ellipsoidal cones that properly contain C_0 in general. Therefore, our methods carried out on a convex set $\Theta \supseteq C_0$ work similarly in their setting.

Gollapudi et al. [25] studied essentially the same problem as online inverse linear optimization under the name of contextual recommendation (where they and Besbes et al. [7, 8] appear to have been unaware of each other's work). As with Besbes et al. [7, 8], Gollapudi et al. [25] assumed that c^* and $X_1, \dots, X_T$ lie in the unit Euclidean ball, denoted by $\mathbb{B}^n$. Similar to Besbes et al. [7, 8], their method maintains the region K_t, which is the intersection of hyperplanes $\left\{ c \in \mathbb{R}^d : \langle c - \hat{c}_s, x_s - \hat{x}_s \rangle \geq 0 \right\}$ for $s = 1, \dots, t-1$, and sets $\hat{c}_t$ to the centroid of $K_t + \frac{1}{T}\mathbb{B}^n$, where $+$ is the Minkowski sum. As regards the regret analysis, their key idea is to use the approximate Grünbaum theorem: whenever the learner incurs $\langle c^*, x_t - \hat{x}_t \rangle \geq \frac{1}{T}$, $\mathsf{Vol}\left(K_t + \frac{1}{T}\mathbb{B}^n\right)$ decreases by a constant factor, where Vol denotes the volume. Consequently, $\mathsf{Vol}\left(K_1 + \frac{1}{T}\mathbb{B}^n\right) / \mathsf{Vol}\left(K_T + \frac{1}{T}\mathbb{B}^n\right) \lesssim T^n$ implies the regret bound of $R_T^{c^*} = \sum_t \langle c^*, x_t - \hat{x}_t \rangle = O(n \ln T)$. As such, the per-round complexity of their method also inherently depends on T, and Gollapudi et al. [25, Section 1.2] only claims the total time complexity of $\mathrm{poly}(n, T)$. In this setting, our ONS-based method achieves a regret bound of $O(n \ln(T/n))$ and is arguably more efficient since the per-round complexity is independent of T.

Algorithm 1 Online Newton Step

Set $\gamma = \frac{1}{2}\min\{\frac{1}{\beta}, \alpha\}$, $\varepsilon = \frac{n}{W^2\gamma^2}$, $A_0 = \varepsilon I_n$, and $w_1 \in \mathcal{W}$.
for $t = 1, \dots, T$ **:**
  Play w_t and observe q_t.
  $A_t \leftarrow A_{t-1} + \nabla q_t(w_t)\nabla q_t(w_t)^\top$.
  $w_{t+1} \leftarrow \arg\min\left\{ \left\| w_t - \frac{1}{\gamma}A_t^{-1}\nabla q_t(w_t) - w \right\|_{A_t}^2 : w \in \mathcal{W} \right\}$. ▷ Generalized projection.

Estimating the per-round complexity of Gollapudi et al. [25]. As described above, the method of Gollapudi et al. [25] requires $\hat{x}_t$ for each t, and hence the per-round complexity involves τ_{solve}, the time to solve $\max_{x \in X_t}\langle \hat{c}_t, x\rangle$. Aside from this, its per-round complexity is dominated by the cost for computing the centroid of $K_t + \frac{1}{T}\mathbb{B}^n$, where K_t is represented by $O(T)$ hyperplanes. It is known that the problem of exactly computing the centroid is #P-hard in general, but we can approximate it via sampling with a membership oracle of $K_t + \frac{1}{T}\mathbb{B}^n$. To the best of our knowledge, computing a point that is ε-close to the centroid takes $O(n^4/\varepsilon^2)$ membership queries, up to logarithmic factors [21, Theorem 5.7], and it is natural to set $\varepsilon = 1/T$ to make the approximation error negligible. Thus, it takes $O(n^4T^2)$ membership queries. Regarding the complexity of the membership oracle, naively checking whether a given point satisfies all the $O(T)$ linear constraints of K_t takes $O(nT)$ time. Handling the Minkowski sum with $\frac{1}{T}\mathbb{B}$ would complicates the procedure, though it can be done in $\mathrm{poly}(n, T)$ time by using, for example, Frank–Wolfe-type algorithms [22, 24, 31, 37]. For now, $O(nT)$ would be a reasonable (optimistic) estimate of the complexity of the membership oracle. Consequently, the total per-round complexity of their method is estimated to be $O\big(\tau_{\text{solve}} + n^5T^3\big)$ (or higher).

B Details of ONS and MetaGrad

We present the details of ONS and MetaGrad. The main purpose of this section is to provide simple descriptions and analyses of those algorithms, thereby assisting readers who are not familiar with them. As in Appendix B.4, we can also derive a regret bound of MetaGrad that yields a similar result to Theorem 4.1 directly from the results of Van Erven et al. [56].

First, we discuss the regret bound of ONS used by η-experts in MetaGrad, proving Proposition 2.5. Then, we establish the regret bound of MetaGrad in Proposition 2.6.

B.1 Regret bound of ONS

Let $I_n \in \mathbb{R}^{n\times n}$ denote the identity matrix. For any $A, B \in \mathbb{R}^{n\times n}$, $A \succeq B$ means that $A - B$ is positive semidefinite. For positive semidefinite $A \in \mathbb{R}^{n\times n}$, let $\|x\|_A = \sqrt{x^\top Ax}$ for $x \in \mathbb{R}^n$. Let $\mathcal{W} \subseteq \mathbb{R}^n$ be a closed convex set. A function $q : \mathcal{W} \to \mathbb{R}$ is *α-exp-concave* for some $\alpha > 0$ if $\mathcal{W} \ni w \mapsto \mathrm{e}^{-\alpha q(w)}$ is concave. For twice differentiable q, this is equivalent to $\nabla^2 q(w) \succeq \alpha\nabla q(w)\nabla q(w)^\top$. The following regret bound of ONS mostly comes from the standard analysis [26, Section 4.4], and hence readers familiar with it can skip the subsequent proof. The only modification lies in the use of β instead of $W\lambda$ (defined below), where $\beta \le W\lambda$ always holds and hence slightly tighter. This leads to the multiplicative factor of B, rather than DK, in Theorems 3.1 and 4.1.

Proposition B.1. *Let $\mathcal{W} \subseteq \mathbb{R}^n$ be a closed convex set with the ℓ_2-diameter of at most $W > 0$. Assume that $q_1, \dots, q_T : \mathcal{W} \to \mathbb{R}$ are twice differentiable and α-exp-concave for some $\alpha > 0$. Additionally, assume that there exist $\beta, \lambda > 0$ such that $\max_{w\in\mathcal{W}}\left|\nabla q_t(w_t)^\top(w - w_t)\right| \le \beta$ and $\|\nabla q_t(w_t)\|_2 \le \lambda$ hold. Let $w_1, \dots, w_T \in \mathcal{W}$ be the outputs of ONS (Algorithm 1). Then, for any $u \in \mathcal{W}$, it holds that*

$$\sum_{t=1}^{T}(q_t(w_t) - q_t(u)) \le \frac{n}{2\gamma}\left(\ln\left(\frac{W^2\gamma^2\lambda^2 T}{n} + 1\right) + 1\right),$$

where $\gamma = \frac{1}{2}\min\{\frac{1}{\beta}, \alpha\}$ is the parameter used in ONS.

Proof. We first give a useful inequality that follows from the α-exp-concavity. By the same analysis as the proof of Hazan [26, Lemma 4.3], for $\gamma \leq \frac{\alpha}{2}$, we have

$$q_t(w_t) - q_t(u) \leq \frac{1}{2\gamma} \ln\bigl(1 - 2\gamma \nabla q_t(w_t)^\top (u - w_t)\bigr).$$

Note that we also have $\bigl|2\gamma \nabla q_t(w_t)^\top (u - w_t)\bigr| \leq 2\gamma\beta \leq 1$. Since $\ln(1-x) \leq -x - x^2/4$ holds for $x \geq -1$, applying this with $x = 2\gamma \nabla q_t(w_t)^\top (u - w_t)$ yields

$$q_t(w_t) - q_t(u) \leq \nabla q_t(w_t)^\top (w_t - u) - \frac{\gamma}{2} (w_t - u)^\top \nabla q_t(w_t) \nabla q_t(w_t)^\top (w_t - u). \tag{8}$$

We turn to the iterates of ONS. Since w_{t+1} is the projection of $w_t - \frac{1}{\gamma} A_t^{-1} \nabla q_t(w_t)$ onto $\mathcal{W}$ with respect to the norm $\|\cdot\|_{A_t}$, we have $\|w_{t+1} - u\|_{A_t}^2 \leq \left\|w_t - \frac{1}{\gamma} A_t^{-1} \nabla q_t(w_t) - u\right\|_{A_t}^2$ for $u \in \mathcal{W}$ due to the Pythagorean theorem, hence

$$\begin{aligned}
&(w_{t+1} - u)^\top A_t (w_{t+1} - u) \\
&\leq \left(w_t - \frac{1}{\gamma} A_t^{-1} \nabla q_t(w_t) - u\right)^\top A_t \left(w_t - \frac{1}{\gamma} A_t^{-1} \nabla q_t(w_t) - u\right) \\
&= (w_t - u)^\top A_t (w_t - u) - \frac{2}{\gamma} \nabla q_t(w_t)^\top (w_t - u) + \frac{1}{\gamma^2} \nabla q_t(w_t)^\top A_t^{-1} \nabla q_t(w_t).
\end{aligned}$$

Rearranging the terms, we obtain

$$\begin{aligned}
&\nabla q_t(w_t)^\top (w_t - u) \\
&\leq \frac{1}{2\gamma} \nabla q_t(w_t)^\top A_t^{-1} \nabla q_t(w_t) + \frac{\gamma}{2} (w_t - u)^\top A_t (w_t - u) - \frac{\gamma}{2} (w_{t+1} - u)^\top A_t (w_{t+1} - u).
\end{aligned}$$

From $A_t = A_{t-1} + \nabla q_t(w_t) \nabla q_t(w_t)^\top$, summing over t and ignoring $\frac{\gamma}{2}(w_{T+1} - u)^\top A_T (w_{T+1} - u) \geq 0$, we obtain

$$\begin{aligned}
&\sum_{t=1}^T \nabla q_t(w_t)^\top (w_t - u) \\
&\leq \frac{1}{2\gamma} \sum_{t=1}^T \nabla q_t(w_t)^\top A_t^{-1} \nabla q_t(w_t) + \frac{\gamma}{2} (w_1 - u)^\top A_1 (w_1 - u) \\
&\quad + \frac{\gamma}{2} \sum_{t=2}^T (w_t - u)^\top (A_t - A_{t-1})(w_t - u) \\
&= \frac{1}{2\gamma} \sum_{t=1}^T \nabla q_t(w_t)^\top A_t^{-1} \nabla q_t(w_t) + \frac{\gamma}{2} (w_1 - u)^\top (A_1 - \nabla q_1(w_1) \nabla q_1(w_1)^\top)(w_1 - u) \\
&\quad + \frac{\gamma}{2} \sum_{t=1}^T (w_t - u)^\top \nabla q_t(w_t) \nabla q_t(w_t)^\top (w_t - u).
\end{aligned}$$

Since we have $A_1 - \nabla q_1(w_1) \nabla q_1(w_1)^\top = A_0 = \varepsilon I_n$ and $\varepsilon = \frac{n}{W^2 \gamma^2}$, the above inequality implies

$$\begin{aligned}
&\sum_{t=1}^T \nabla q_t(w_t)^\top (w_t - u) - \frac{\gamma}{2} \sum_{t=1}^T (w_t - u)^\top \nabla q_t(w_t) \nabla q_t(w_t)^\top (w_t - u) \\
&\leq \frac{1}{2\gamma} \sum_{t=1}^T \nabla q_t(w_t)^\top A_t^{-1} \nabla q_t(w_t) + \frac{\gamma}{2} (w_1 - u)^\top A_0 (w_1 - u) \\
&\leq \frac{1}{2\gamma} \sum_{t=1}^T \nabla q_t(w_t)^\top A_t^{-1} \nabla q_t(w_t) + \frac{\gamma\varepsilon}{2} \|w_1 - u\|_2^2 \\
&\leq \frac{1}{2\gamma} \sum_{t=1}^T \nabla q_t(w_t)^\top A_t^{-1} \nabla q_t(w_t) + \frac{n}{2\gamma}.
\end{aligned} \tag{9}$$

The first term in the right-hand side is bounded as follows due to the celebrated elliptical potential lemma (e.g., Hazan [26, proof of Theorem 4.5]):

$$\sum_{t=1}^{T} \nabla q_t(w_t)^\top A_t^{-1} \nabla q_t(w_t) \le \ln \frac{\det A_T}{\det A_0} \le n \ln\left(\frac{T\lambda^2}{\varepsilon} + 1\right) = n \ln\left(\frac{W^2\gamma^2\lambda^2 T}{n} + 1\right), \quad (10)$$

where we used $\det A_0 = \varepsilon^n$ and $\det A_T = \det\left(\sum_{t=1}^{T} \nabla q_t(w_t) \nabla q_t(w_t)^\top + \varepsilon I_n\right) \le \left(T\lambda^2 + \varepsilon\right)^n$, which follows from the fact that eigenvalues of $\sum_{t=1}^{T} \nabla q_t(w_t) \nabla q_t(w_t)^\top$ are at most $T\lambda^2$. Combining (8), (9), and (10), we obtain

$$\begin{aligned}\sum_{t=1}^{T} (q_t(w_t) - q_t(u)) &\le \sum_{t=1}^{T} \nabla q_t(w_t)^\top (w_t - u) - \frac{\gamma}{2} \sum_{t=1}^{T} (w_t - u)^\top \nabla q_t(w_t) \nabla q_t(w_t)^\top (w_t - u) \\ &\le \frac{n}{2\gamma}\left(\ln\left(\frac{W^2\gamma^2\lambda^2 T}{n} + 1\right) + 1\right)\end{aligned}$$

as desired. □

B.2 Regret bound of η-expert

We now establish the regret bound of ONS in Proposition 2.5, which is used by η-experts in MetaGrad. Let $\eta \in \left(0, \frac{1}{5H}\right]$ and consider applying ONS to the following loss functions, which are defined in (4):

$$f_t^\eta(w) = -\eta\langle w_t - w, g_t\rangle + \eta^2 \langle w_t - w, g_t\rangle^2 \quad \text{for } t = 1, \dots, T.$$

As in Proposition 2.5, the ℓ_2-diameter of $\mathcal{W}$ is at most $W > 0$, and the following conditions hold:

$$w_t \in \mathcal{W}, \quad \|g_t\|_2 \le G, \quad \text{and} \quad \max_{w, w' \in \mathcal{W}} \langle w - w', g_t\rangle \le H \quad \text{for } t = 1, \dots, T.$$

From $\nabla f_t^\eta(w) = \eta\left(1 - 2\eta g_t^\top (w_t - w)\right) g_t$ and $\nabla^2 f_t^\eta(w) = 2\eta^2 g_t g_t^\top$, we have

$$\begin{aligned}\nabla f_t^\eta(w) \nabla f_t^\eta(w)^\top &= \eta^2 \left(1 - 2\eta g_t^\top (w_t - w)\right)^2 g_t g_t^\top \\ &\preceq \eta^2 (1 + 2\eta H)^2 g_t g_t^\top = \frac{(1 + 2\eta H)^2}{2} \nabla^2 f_t^\eta(w) \quad \text{for all } w \in \mathcal{W}, \\ \max_{w \in \mathcal{W}} \left|\nabla f_t^\eta(w_t^\eta)^\top (w - w_t^\eta)\right| &= \max_{w \in \mathcal{W}} \left|\eta g_t^\top (w - w_t^\eta) - 2\eta^2 \left(g_t^\top (w_t^\eta - w_t)\right)^2\right| \\ &\le \eta H + 2\eta^2 H^2, \\ \|\nabla f_t^\eta(w)\|_2 &= \left\|\eta\left(1 - 2\eta g_t^\top (w_t - w)\right) g_t\right\|_2 \le \eta(1 + 2\eta H) G.\end{aligned}$$

Therefore, f_t^η satisfies the conditions in Proposition B.1 with $\alpha = \frac{2}{(1+2\eta H)^2}$, $\beta = \eta H + 2\eta^2 H^2$, and $\lambda = \eta(1 + 2\eta H) G$. Since $\frac{1}{\alpha} = \frac{1}{2} + 2\eta H + 2\eta^2 H^2 \ge \beta$ holds, we have $\gamma = \frac{1}{2} \min\left\{\frac{1}{\beta}, \alpha\right\} = \frac{\alpha}{2}$. Thus, for any $\eta \in \left(0, \frac{1}{5H}\right]$, we have $\gamma \in \left[\frac{25}{49}, 1\right) \subseteq \left[\frac{1}{2}, 1\right]$ and $\gamma\lambda = \frac{\eta G}{1 + 2\eta H} \le \frac{G}{7H}$. Consequently, Proposition B.1 implies that for any $u \in \mathcal{W}$, the regret of the η-expert's ONS is bounded as follows:

$$\sum_{t=1}^{T} (f_t^\eta(w_t^\eta) - f_t^\eta(u)) \le n\left(\ln\left(\frac{W^2 G^2 T}{49 n H^2} + 1\right) + 1\right) = O\left(n \ln\left(\frac{WGT}{Hn}\right)\right). \quad (11)$$

B.3 Regret bound of MetaGrad

We turn to MetaGrad applied to convex loss functions $f_1, \dots, f_T \colon \mathcal{W} \to \mathbb{R}$. We here use $w_t \in \mathcal{W}$ and $g_t \in \partial f_t(w_t)$ to denote the tth output of MetaGrad and a subgradient of f_t at w_t, respectively, for $t = 1, \dots, T$. We assume that these satisfy the conditions in (3), as stated in Proposition 2.6.

Algorithm 2 describes the procedure of MetaGrad. Define $\eta_i = \frac{2^{-i}}{5H}$ for $i = 0, 1, \dots, \left\lceil \frac{1}{2} \log_2 T \right\rceil$, called *grid points*, and let $\mathcal{G} \subseteq \left(0, \frac{1}{5H}\right]$ denote the set of all grid points. For each $\eta \in \mathcal{G}$, η-expert runs ONS with loss functions $f_1^\eta, \dots, f_T^\eta$ to compute $w_1^\eta, \dots, w_T^\eta$. In each round t, we obtain w_t by

Algorithm 2 MetaGrad

$p_1^{\eta_i} \leftarrow \frac{C}{(i+1)(i+2)}$ for all $\eta_i \in \mathcal{G} = \left\{ \frac{2^{-i}}{5H} \,:\, i = 0, 1, \ldots, \left\lceil \frac{1}{2}\log_2 T \right\rceil \right\}$.
for $t = 1, \ldots, T$ **:**
  Fetch w_t^η from η-experts for all $\eta \in \mathcal{G}$.
  Play $w_t = \frac{\sum_{\eta\in\mathcal{G}} \eta p_t^\eta w_t^\eta}{\sum_{\eta\in\mathcal{G}} \eta p_t^\eta}$.
  Observe $g_t \in \partial f_t(w_t)$ and send (w_t, g_t) to η-experts for all $\eta \in \mathcal{G}$.
  $p_{t+1}^\eta \leftarrow p_t^\eta \exp(-f_t^\eta(w_t^\eta))/Z_t$ for all $\eta \in \mathcal{G}$, where $Z_t = \sum_{\eta\in\mathcal{G}} p_t^\eta \exp(-f_t^\eta(w_t^\eta))$.

aggregating the η-experts' outputs w_t^η based on the exponentially weighted average method (EWA). We set the prior as $p_1^{\eta_i} = \frac{C}{(i+1)(i+2)}$ for all $\eta_i \in \mathcal{G}$, where $C = 1 + \frac{1}{1+\left\lceil \frac{1}{2}\log_2 T \right\rceil}$. Then, it is known that for every $\eta \in \mathcal{G}$, the regret of EWA relative to the η-expert's choice w_t^η is bounded as follows:

$$\begin{aligned}\sum_{t=1}^T (f_t^\eta(w_t) - f_t^\eta(w_t^\eta)) &\le \ln\frac{1}{p_1^\eta} \le \ln\left(\left(\left\lceil\frac{1}{2}\log_2 T\right\rceil + 1\right)\left(\left\lceil\frac{1}{2}\log_2 T\right\rceil + 2\right)\right) \\ &\le 2\ln\left(\frac{1}{2}\log_2 T + 3\right),\end{aligned} \tag{12}$$

where we used $C \ge 1$ in the second inequality. We here omit the proof as it is completely the same as that of Van Erven and Koolen [55, Lemma 4] (see also Wang et al. [58, Lemma 1]).

We are ready to prove Proposition 2.6. Let $V_T^u = \sum_{t=1}^T \langle w_t - u, g_t\rangle^2$. By using $f_t^\eta(w_t) = 0$, (11), and (12), it holds that

$$\begin{aligned}\sum_{t=1}^T \langle w_t - u, g_t\rangle &= -\frac{\sum_{t=1}^T f_t^\eta(u)}{\eta} + \eta V_T^u \\ &= \frac{1}{\eta}\left(\underbrace{\sum_{t=1}^T (f_t^\eta(w_t) - f_t^\eta(w_t^\eta))}_{\text{Regret of EWA w.r.t. } w_t^\eta} + \underbrace{\sum_{t=1}^T (f_t^\eta(w_t^\eta) - f_t^\eta(u))}_{\text{Regret of } \eta\text{-expert w.r.t. } u}\right) + \eta V_T^u \\ &\le \frac{1}{\eta}\left(2\ln\left(\frac{1}{2}\log_2 T + 3\right) + n\left(\ln\left(\frac{W^2G^2T}{49nH^2} + 1\right) + 1\right)\right) + \eta V_T^u\end{aligned}$$

for all $\eta \in \mathcal{G}$. For brevity, let

$$A = 2\ln\left(\frac{1}{2}\log_2 T + 3\right) + n\left(\ln\left(\frac{W^2G^2T}{49nH^2} + 1\right) + 1\right) \ge 1.$$

If we knew V_T^u, we could set η to $\eta^* := \sqrt{\frac{A}{V_T^u}} \ge \frac{1}{5H\sqrt{T}}$ to minimize the above regret bound, $\frac{A}{\eta} + \eta V_T^u$. Actually, we can do almost the same without knowing V_T^u thanks to the fact that the regret bound holds for all $\eta \in \mathcal{G}$. If $\eta^* \le \frac{1}{5H}$, by construction we have a grid point $\eta \in \mathcal{G}$ such that $\eta^* \in \left[\frac{\eta}{2}, \eta\right]$, hence

$$\sum_{t=1}^T \langle w_t - u, g_t\rangle \le \eta V_T^u + \frac{A}{\eta} \le 2\eta^* V_T^u + \frac{A}{\eta^*} \le 3\sqrt{AV_T^u}.$$

Otherwise, $\eta^* = \sqrt{\frac{A}{V_T^u}} \ge \frac{1}{5H}$ holds, which implies $V_T^u \le 25H^2A$. Thus, for $\eta_0 = \frac{1}{5H} \in \mathcal{G}$, we have

$$\sum_{t=1}^T \langle w_t - u, g_t\rangle \le \eta_0 V_T^u + \frac{A}{\eta_0} \le 10HA.$$

Therefore, in any case, we have

$$\sum_{t=1}^T \langle w_t - u, g_t\rangle \le 3\sqrt{AV_T^u} + 10HA = O\left(\sqrt{n\ln\left(\frac{WGT}{Hn}\right)\cdot V_T^u} + Hn\ln\left(\frac{WGT}{Hn}\right)\right),$$

obtaining the regret bound in Proposition 2.6.

Algorithm 3 $O(1)$-Regret Algorithm for $n = 2$.

Set $\mathcal{C}_1$ to $\mathbb{S}^1$.
for $t = 1, \dots, T$ **:**
  Draw $\hat{c}_t$ uniformly at random from $\mathcal{C}_t$.
  Observe (X_t, x_t).
  $\mathcal{C}_{t+1} \leftarrow \mathcal{C}_t \cap \mathcal{N}_t$. $\triangleright$ $\mathcal{N}_t$ is the normal cone.

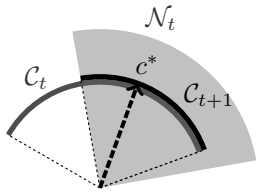

Figure 1: Illustration of c^*, $\mathcal{C}_t$, $\mathcal{N}_t$, and $\mathcal{C}_{t+1}$.

B.4 Lipschitz adaptivity and anytime guarantee

Recent studies [41, 56] have shown that MetaGrad can be further made Lipschitz adaptive and agnostic to the number of rounds. Specifically, MetaGrad given in Van Erven et al. [56, Algorithms 1 and 2] works without knowing G, H, or T in advance, while using (a guess of) W. By expanding the proofs of Van Erven et al. [56, Theorem 7 and Corollary 8], we can confirm that the refined version of MetaGrad enjoys the following regret bound:

$$\sum_{t=1}^{T} \langle w_t - u, g_t \rangle = O\left(\sqrt{n \ln\left(\frac{WGT}{n}\right) \cdot V_T^u} + Hn \ln\left(\frac{WGT}{n}\right) \right).$$

By using this in the proof of Theorem 4.1, we obtain

$$\sum_{t=1}^{T} \langle c^*, x_t - \hat{x}_t \rangle \le \sum_{t=1}^{T} \langle \hat{c}_t - c^*, \hat{x}_t - x_t \rangle = O\left(Bn \ln\left(\frac{DKT}{n}\right) + \sqrt{\Delta_T Bn \ln\left(\frac{DKT}{n}\right)} \right),$$

and the algorithm does not require knowing K, B, T, or Δ_T in advance.

C On removing the $\ln T$ factor: the case of $n = 2$

This section provides an additional discussion on closing the $\ln T$ gap in the upper and lower bounds on the regret. Specifically, focusing on the case of $n = 2$, we provide a simple algorithm that achieves a regret bound of $O(1)$ in expectation, removing the $\ln T$ factor. We also observe that extending the algorithm to general $n \ge 2$ might be challenging. Note that Gollapudi et al. [25, Theorem 4.1] has already established a regret bound of $\exp(O(n \ln n))$ as mentioned in Section 1.2, which implies an $O(1)$-regret bound for $n = 2$. The purpose of this section is simply to stimulate discussions on closing the $\ln T$ gap by presenting another simple analysis. Below, let $\mathbb{B}^n$ and $\mathbb{S}^{n-1}$ denote the unit Euclidean ball and sphere in $\mathbb{R}^n$, respectively, for any integer $n > 1$.

C.1 An $O(1)$-regret method for $n = 2$

We focus on the case of $n = 2$ and present an algorithm that achieves a regret bound of $O(1)$ in expectation. We assume that all $x_t \in X_t$ are optimal for c^* for $t = 1, \dots, T$. For simplicity, we additionally assume that all X_t are contained in $\frac{1}{2}\mathbb{B}^2$ and that c^* lies in $\mathbb{S}^1$. For any non-zero vectors $c, c' \in \mathbb{R}^n$, let $\theta(c, c')$ denote the angle between the two vectors. The following lemma from Besbes et al. [8], which holds for general $n \ge 2$, is useful in the subsequent analysis.

Lemma C.1 (Besbes et al. [8, Lemma 1])**.** *Let $c^*, \hat{c}_t \in \mathbb{S}^{n-1}$, $X_t \subseteq \frac{1}{2}\mathbb{B}^n$, $x_t \in \arg\max_{x \in X_t} \langle c^*, x \rangle$, and $\hat{x}_t \in \arg\max_{x \in X_t} \langle \hat{c}_t, x \rangle$. If $\theta(c^*, \hat{c}_t) < \pi/2$, it holds that $\langle c^*, x_t - \hat{x}_t \rangle \le \sin \theta(c^*, \hat{c}_t)$.*

Our algorithm, given in Algorithm 3, is a randomized variant of the one investigated by Besbes et al. [7, 8]. The procedure is intuitive: we maintain a set $\mathcal{C}_t \subseteq \mathbb{S}^1$ that contains c^*, from which we draw $\hat{c}_t$ uniformly at random, and update $\mathcal{C}_t$ by excluding the area that is ensured not to contain c^* based on the tth feedback (X_t, x_t). Formally, the last step takes the intersection of $\mathcal{C}_t$ and the *normal cone* $\mathcal{N}_t = \{ c \in \mathbb{R}^n : \langle c, x_t - x \rangle \ge 0, \forall x \in X_t \}$ of X_t at x_t, which is a convex cone containing c^*. Therefore, every $\mathcal{C}_t$ is a connected arc on $\mathbb{S}^1$ and is non-empty due to $c^* \in \mathcal{C}_t$ (see Figure 1).

Theorem C.2. *For the above setting of $n = 2$, Algorithm 3 achieves $\mathbb{E}\left[R_T^{c^*}\right] \le 2\pi$.*

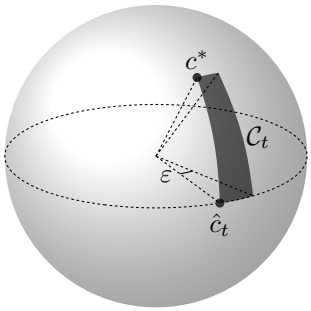

Figure 2: An example of $\mathcal{C}_t$ on $\mathbb{S}^2$. The darker area, $A(\mathcal{C}_t)$, becomes arbitrarily small as $\varepsilon \to 0$, while $\theta(c^*, \hat{c}_t)$ does not.

Proof. For any connected arc $\mathcal{C} \subseteq \mathbb{S}^1$, let $A(\mathcal{C}) \in [0, 2\pi]$ denote its central angle, which equals its length. Fix $\mathcal{C}_t$. If $\hat{c}_t \in \mathcal{C}_t \cap \mathrm{int}(\mathcal{N}_t)$, where $\mathrm{int}(\cdot)$ denotes the interior, $\hat{x}_t = x_t$ is the unique optimal solution for $\hat{c}_t$, hence $\langle c^*, x_t - \hat{x}_t \rangle = 0$. Taking the expectation about the randomness of $\hat{c}_t$, we have

$$\begin{aligned}\mathbb{E}[\langle c^*, x_t - \hat{x}_t \rangle] &= \Pr[\hat{c}_t \in \mathcal{C}_t \setminus \mathrm{int}(\mathcal{N}_t)]\, \mathbb{E}[\, \langle c^*, x_t - \hat{x}_t \rangle \,|\, \hat{c}_t \in \mathcal{C}_t \setminus \mathrm{int}(\mathcal{N}_t)\,] \\ &= \frac{A(\mathcal{C}_t \setminus \mathcal{N}_t)}{A(\mathcal{C}_t)}\, \mathbb{E}[\, \langle c^*, x_t - \hat{x}_t \rangle \,|\, \hat{c}_t \in \mathcal{C}_t \setminus \mathrm{int}(\mathcal{N}_t)\,],\end{aligned}$$

where we used $\Pr[\hat{c}_t \in \mathcal{C}_t \setminus \mathrm{int}(\mathcal{N}_t)] = \Pr[\hat{c}_t \in \mathcal{C}_t \setminus \mathcal{N}_t] = A(\mathcal{C}_t \setminus \mathcal{N}_t)/A(\mathcal{C}_t)$ (since the boundary of $\mathcal{N}_t$ has zero measure). If $A(\mathcal{C}_t) \geq \pi/2$, from $\langle c^*, x_t - \hat{x}_t \rangle \leq \|c^*\|_2 \|x_t - \hat{x}_t\|_2 \leq 1$, we have

$$\mathbb{E}[\langle c^*, x_t - \hat{x}_t \rangle] \leq \frac{2}{\pi} A(\mathcal{C}_t \setminus \mathcal{N}_t) \leq A(\mathcal{C}_t \setminus \mathcal{N}_t).$$

If $A(\mathcal{C}_t) < \pi/2$, Lemma C.1 and $\hat{c}_t, c^* \in \mathcal{C}_t$ imply $\langle c^*, x_t - \hat{x}_t \rangle \leq \sin \theta(c^*, \hat{c}_t) \leq \sin A(\mathcal{C}_t)$. Thus, by using $\frac{1}{x} \sin x \leq 1$ ($x \in \mathbb{R}$), we obtain

$$\mathbb{E}[\langle c^*, x_t - \hat{x}_t \rangle] \leq \frac{A(\mathcal{C}_t \setminus \mathcal{N}_t)}{A(\mathcal{C}_t)} \sin A(\mathcal{C}_t) \leq A(\mathcal{C}_t \setminus \mathcal{N}_t).$$

Therefore, we have $\mathbb{E}[\langle c^*, x_t - \hat{x}_t \rangle] \leq A(\mathcal{C}_t \setminus \mathcal{N}_t)$ in any case. Consequently, we obtain

$$\mathbb{E}\left[R_T^{c^*}\right] = \sum_{t=1}^{T} \mathbb{E}[\langle c^*, x_t - \hat{x}_t \rangle] \leq \sum_{t=1}^{T} A(\mathcal{C}_t \setminus \mathcal{N}_t) \leq 2\pi,$$

where the last inequality is due to $\mathcal{C}_{t+1} = \mathcal{C}_t \cap \mathcal{N}_t$, which implies $\mathcal{C}_s \subseteq \mathcal{C}_t$ and $\mathcal{C}_s \cap (\mathcal{C}_t \setminus \mathcal{N}_t) = \emptyset$ for any $s > t$, and hence no double counting occurs in the above summation. □

C.2 Discussion on higher-dimensional cases

Algorithm 3 might appear applicable to general $n \geq 2$ by replacing $\mathbb{S}^1$ with $\mathbb{S}^{n-1}$ and defining $A(\mathcal{C}_t)$ as the area of $\mathcal{C}_t \subseteq \mathbb{S}^{n-1}$. However, this idea faces a challenge in bounding the regret when extending the above proof to general $n \geq 2$.[9]

As suggested in the proof of Theorem C.2, bounding $\mathbb{E}[\langle c^*, x_t - \hat{x}_t \rangle]$ is trickier when $A(\mathcal{C}_t)$ is small (cf. the case of $A(\mathcal{C}_t) < \pi/2$). Luckily, when $n = 2$, we can bound it thanks to Lemma C.1 and $\sin \theta(c^*, \hat{c}_t) \leq \sin A(\mathcal{C}_t)$, where the latter roughly means the angle, $\theta(c^*, \hat{c}_t)$, is bounded by the area, $A(\mathcal{C}_t)$, from above. Importantly, when $n = 2$, both the central angle and the area of an arc are identified with the length of the arc, which is the key to establishing $\sin \theta(c^*, \hat{c}_t) \leq \sin A(\mathcal{C}_t)$. This is no longer true for $n \geq 3$. As in Figure 2, the area, $A(\mathcal{C}_t)$, can be arbitrarily small even if the angle

[9] We note that a hardness result given in Besbes et al. [8, Theorem 2] is different from what we encounter here. They showed that their *greedy circumcenter policy* fails to achieve a sublinear regret, which stems from the shape of the initial knowledge set and the behavior of the greedy rule for selecting $\hat{c}_t$; this differs from the issue discussed above.

within there, or the maximum $\theta(c^*, \hat{c}_t)$ for $c^*, \hat{c}_t \in \mathcal{C}_t$, is large.[10] This is why the proof for the case of $n = 2$ does not directly extend to higher dimensions. We leave closing the $O(\ln T)$ gap for $n \geq 3$ as an important open problem for future research.

D Numerical experiments

We conducted numerical experiments to complement our theoretical results. Experiments were conducted on Google Colab equipped with an Intel® Xeon® CPU @ 2.20GHz, 12 GB RAM, running Ubuntu 22.04.4 LTS with Python 3.12.11. The code is available at https://github.com/ssakaue/online-inverse-linear-optimization-code.

We use a setup based on the hard instance considered in our lower bound analysis (Section 5). Let c^* be a random vector with $\|c^*\|_2 = 1$. The learner's prediction set Θ is the n-dimensional Euclidean unit ball. At each round t, we sample an endpoint v uniformly at random from the unit sphere in $\mathbb{R}^n$, and set $X_t = \{-v, +v\}$. We report results for $T = 10{,}000$ rounds in dimensions $n = 2$, 20, and 200. To mitigate randomness, we repeat each experiment 10 times independently and report mean and standard deviation.

Compared methods. We compare the following three methods:

- **ONS**: our proposed method based on ONS,
- **OGD**: the OGD-based method of Bärmann et al. [4],
- **CP**: a cutting-plane style method inspired by Gollapudi et al. [25].

For CP, computing the centroid of the feasible region is #P-hard. Therefore, we adopt a randomized heuristic: we pre-sample $n \times 10^4$ candidate points from the unit ball, eliminate those violating accumulated cuts, and approximate the centroid by averaging the remaining points.

Results. Table 2 summarizes the cumulative regret and runtime over $T = 10{,}000$ rounds (mean $\pm$ standard deviation across 10 trials).

Table 2: Experimental results over $T = 10{,}000$ rounds (mean $\pm$ standard deviation across 10 independent runs).

(a) Cumulative Regret

Method	$n = 2$	$n = 20$	$n = 200$
ONS	7.81 ± 2.52	31.55 ± 0.43	46.19 ± 0.88
OGD	8.46 ± 4.07	33.80 ± 0.50	67.41 ± 1.36
CP	2.77 ± 0.68	829.91 ± 287.00	515.50 ± 43.75

(b) Cumulative Runtime (s)

Method	$n = 2$	$n = 20$	$n = 200$
ONS	0.648 ± 0.115	0.705 ± 0.086	9.073 ± 0.727
OGD	0.150 ± 0.024	0.150 ± 0.017	0.233 ± 0.019
CP	4.670 ± 3.205	5.700 ± 1.748	71.252 ± 12.152

When the dimension is small ($n = 2$), CP achieves the lowest cumulative regret. However, its cumulative regret deteriorates significantly for $n = 20$ and $n = 200$. This degradation stems from the limited number of pre-sampled candidate points: in principle, about T^n samples would be required for an accurate approximation, which is infeasible even for moderate n. Moreover, although the above randomized centroid approximation substantially reduces computation by averaging over the

[10] A similar issue, though leading to different challenges, is noted in Besbes et al. [8, Section 4.4], where their method encounters ill-conditioned (or elongated) ellipsoids. They addressed this by appropriately determining when to update the ellipsoidal cone. The $\ln T$ factor arises as a result of balancing being ill-conditioned with the instantaneous regret.

surviving candidates, it remains less scalable than OGD and ONS. These results highlight practical limitations of CP in moderate to high-dimensional settings.

In contrast, ONS consistently achieves low regret values across all dimensions while remaining computationally feasible. It outperforms OGD in terms of the regret and scales reasonably well with increasing n. Note that our ONS implementation uses a straightforward projection subroutine that repeatedly solves similar linear systems—an overhead that could be reduced by more sophisticated implementation techniques. Further speedups could also be achieved via quasi-Newton-type updates or sketching-based techniques, as discussed in Sections 3 and 4.

Taken together, these findings affirm that ONS provides a strong and scalable alternative to existing methods in online inverse linear optimization, especially when CP is computationally infeasible and OGD's regret performance is unsatisfactory.