Extracting Dual Solutions via Primal Optimizers

Yair Carmon
Tel Aviv University, Israel

Arun Jambulapati
University of Michigan, Ann Arbor, MI, USA

Liam O'Carroll
Stanford University, CA, USA

Aaron Sidford
Stanford University, CA, USA

Abstract

We provide a general method to convert a "primal" black-box algorithm for solving regularized convex-concave minimax optimization problems into an algorithm for solving the associated dual maximin optimization problem. Our method adds recursive regularization over a logarithmic number of rounds where each round consists of an approximate regularized primal optimization followed by the computation of a dual best response. We apply this result to obtain new state-of-the-art runtimes for solving matrix games in specific parameter regimes, obtain improved query complexity for solving the dual of the CVaR distributionally robust optimization (DRO) problem, and recover the optimal query complexity for finding a stationary point of a convex function.

2012 ACM Subject Classification Theory of computation → Mathematical optimization

Keywords and phrases Minimax optimization, black-box optimization, matrix games, distributionally robust optimization

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.29

Related Version *Full Version*: http://arxiv.org/abs/2412.02949

Funding *Yair Carmon*: YC acknowledges support from the Israeli Science Foundation (ISF) grant no. 2486/21, and the Alon Fellowship.
Liam O'Carroll: LO acknowledges support from NSF Grant CCF-1955039.
Aaron Sidford: AS acknowledges support from a Microsoft Research Faculty Fellowship, NSF CAREER Grant CCF-1844855, NSF Grant CCF-1955039, and a PayPal research award.

1 Introduction

We consider the foundational problem of efficiently solving convex-concave games. For nonempty, closed, convex constraint sets $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ and differentiable convex-concave objective function $\psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ (namely, $\psi(\cdot, y)$ is convex for any fixed y and $\psi(x, \cdot)$ is concave for any fixed x), we consider the following *primal*, minimax optimization problem (P) and its associated *dual*, maximin optimization problem (D):

$$\underset{x\in\mathcal{X}}{\text{minimize}}\, f(x) \text{ for } f(x) \coloneqq \max_{y\in\mathcal{Y}} \psi(x, y), \text{ and} \tag{P}$$

$$\underset{y\in\mathcal{Y}}{\text{maximize}}\, \phi(y) \text{ for } \phi(y) \coloneqq \min_{x\in\mathcal{X}} \psi(x, y). \tag{D}$$

If additionally $\mathcal{X}$ and $\mathcal{Y}$ are bounded (which we assume for simplicity in the introduction but generalize later), every pair of primal and dual optimizers $x^\star \in \operatorname{argmin}_{x\in\mathcal{X}} f(x)$ and $y^\star \in \operatorname{argmax}_{y\in\mathcal{Y}} \phi(y)$ satisfies the *minimax principle*: $f(x^\star) = \phi(y^\star) = \psi(x^\star, y^\star)$.

© Yair Carmon, Arun Jambulapati, Liam O'Carroll, and Aaron Sidford;
licensed under Creative Commons License CC-BY 4.0

Convex-concave games are pervasive in algorithm design, machine learning, data analysis, and optimization. For example, the games induced by bilinear objectives, i.e., $\psi(x,y) = x^\top Ay + b^\top x + c^\top y$, where $\mathcal{X}$ and $\mathcal{Y}$ are either the simplex, $\Delta^k \coloneqq \{x \in \mathbb{R}^k_{\geq 0} : \|x\|_1 = 1\}$, or the Euclidean ball, $B^k \coloneqq \{x \in \mathbb{R}^k : \|x\|_2 \leq 1\}$, encompass zero-sum games, linear programming, hard-margin support vector machines (SVMs), and minimum enclosing/maximum inscribed ball [14, 2, 31, 10]. Additionally, the case when $\psi(x,y) = \sum_{i=1}^n y_i f_i(x)$ for some functions $f_i : \mathbb{R}^d \to \mathbb{R}$ and $\mathcal{Y}$ is a subset of the simplex encompasses a variety of distributionally robust optimization (DRO) problems [29, 5] and (for $\mathcal{Y} = \Delta^n$) the problem of minimizing the maximum loss [6, 8, 4].

In this paper, we study the following question:

> *Given only a black-box oracle which solves (regularized versions of)* (P) *to ϵ accuracy, and a black-box oracle for computing an exact dual best response $y_x \coloneqq \operatorname{argmax}_{y\in\mathcal{Y}} \psi(x,y)$ to any primal point $x \in \mathcal{X}$, can we extract an ϵ-optimal solution to* (D)*?*

We develop a general *dual-extraction framework* which answers this question in the affirmative. We show that as long as these oracles can be implemented as cheaply as obtaining an ϵ-optimal point of (P), then our framework can obtain an ϵ-optimal point of (D) at the same cost as that of obtaining an ϵ-optimal point of (P), up to logarithmic factors. We then instantiate our framework to obtain new state-of-the-art results in the settings of bilinear matrix games and DRO. Finally, as evidence of its broader applicability, we show that our framework can be used to recover the optimal complexity for computing a stationary point of a smooth convex function.

In the remainder of the introduction we describe our results in greater detail (Section 1.1), give an overview of the dual extraction framework and its analysis (Section 1.2), discuss related work (Section 1.3), and provide a roadmap for the remainder of the paper (Section 1.4).

1.1 Our results

From primal algorithms to dual optimization

We give a general framework which obtains an ϵ-optimal solution to (D) via a sequence of calls to two black-box oracles: (i) an oracle for obtaining an ϵ-optimal point of a regularized version of (P), and (ii) an oracle for obtaining a dual best response $y_x \coloneqq \operatorname{argmax}_{y\in\mathcal{Y}} \psi(x,y)$ for a given $x \in \mathcal{X}$. In particular, we show it is always possible to obtain an ϵ-optimal point to (D) with at most a logarithmic number of calls to each of these oracles, where the regularized primal optimization oracle is always called to an accuracy of ϵ over a logarithmic factor. We also provide an alternate scheme (or more specifically choice of parameters) for applications where the cost of obtaining an ϵ-optimal point of the regularized primal problem decreases sufficiently as the regularization level increases. In such cases, e.g., in our stationary point application, it is possible to avoid even logarithmic factor increases in computational complexity for approximately solving (D) relative to the complexity of approximately solving (P).

Application 1: Bilinear matrix games

In this application, $\psi(x,y) \coloneqq x^\top Ay$ for a matrix $A \in \mathbb{R}^{d\times n}$, $\mathcal{Y}$ is the simplex Δ^n, and $\mathcal{X}$ is either the simplex Δ^d or the unit Euclidean ball B^d. Recently, [8] gave a new state-of-the-art runtime in certain parameter regimes of $\widetilde{O}(nd + n(d/\epsilon)^{2/3} + d\epsilon^{-2})$ for obtaining an expected ϵ-optimal point for the primal problem (P) for this setup. However, unlike previous algorithms

for bilinear matrix games (see Section 1.3 for details), their algorithm does not return an ϵ-optimal solution for the dual (D), and their runtime is not symmetric in the dimensions n and d. As a result, it was unclear whether the same runtime is achievable for obtaining an ϵ-optimal solution of the dual (D). We resolve this question by applying our general framework to achieve an expected ϵ-optimal point of (D) with runtime $\widetilde{O}(nd + n(d/\epsilon)^{2/3} + d\epsilon^{-2})$. We then observe (see Corollary 22) that in the setting where $\mathcal{X} = \Delta^d$, our result can equivalently be viewed as a new state-of-the-art runtime of $\widetilde{O}(nd + d(n/\epsilon)^{2/3} + n\epsilon^{-2})$ for obtaining an ϵ-optimal point of the primal (P) due to the symmetry of ψ and the constraint sets.

Application 2: CVaR at level α DRO

In this application, $\psi(x, y) \coloneqq \sum_{i=1}^n y_i f_i(x)$ for convex, bounded, and Lipschitz loss functions $f_i : \mathbb{R}^d \to \mathbb{R}$, $\mathcal{X}$ is a convex, compact set, and $\mathcal{Y} \coloneqq \left\{y \in \Delta^n : \|y\|_\infty \le \frac{1}{\alpha n}\right\}$ is the CVaR at level α uncertainty set for $\alpha \in [1/n, 1]$. The primal (P) is a canonical and well-studied DRO problem, and corresponds to the average of the top α fraction of the losses. We consider this problem given access to a first-order oracle that, when queried at $x \in \mathbb{R}^d$ and $i \in [n]$, outputs $(f_i(x), \nabla f_i(x))$. Ignoring dependencies other than α, the target accuracy $\epsilon > 0$, and the number of losses n for brevity, [29] gave a matching upper and lower bound (up to logarithmic factors) of $\widetilde{O}(\alpha^{-1}\epsilon^{-2})$ queries to obtain an expected ϵ-optimal point of the primal (P). However, the best known query complexity for obtaining an expected ϵ-optimal point of the dual (D) was $\widetilde{O}(n\epsilon^{-2})$ prior to this paper (see Section 1.3 for details). Applying our general framework to this setting, we obtain an algorithm with a new state-of-the-art query complexity of $\widetilde{O}(\alpha^{-1}\epsilon^{-2} + n)$ for obtaining an expected ϵ-optimal point of the dual (D). In particular, note that this complexity is nearly linear in n when $\epsilon \ge (\alpha n)^{-2}$.

Application 3: Obtaining stationary points of convex functions

In this application, we show that our framework yields an alternative optimal approach for computing an approximate critical point of a smooth convex function given a gradient oracle. Specifically, for $\gamma > 0$ and convex and β-smooth $h : \mathbb{R}^n \to \mathbb{R}$, in Section 5, we give an algorithm which computes $x \in \mathbb{R}^n$ such that $\|\nabla h(x)\|_2 \le \gamma$ using $O\left(\sqrt{\beta\Delta}/\gamma\right)$ gradient queries, where $\Delta \coloneqq h(x_0) - \inf_{x \in \mathbb{R}^n} h(x)$ is the initial suboptimalityf. While this optimal complexity has been achieved before [24, 37, 15, 28, 27], that we achieve it is a consequence of our general framework illustrates its broad applicability.

For this application, we instantiate our framework with $\psi(x, y) \coloneqq \langle x, y\rangle - h^*(y)$, where $h^* : \mathbb{R}^n \to \mathbb{R}$ denotes the convex conjugate of h. (For reasons discussed in Section 5, we actually first substitute h for an appropriately regularized version of h, call it f, before applying the framework, but the following discussion still holds with respect to f.) This objective function ψ is known as the *Fenchel game* and has been used in the past to recover classic convex optimization algorithms (e.g., the Frank-Wolfe algorithm and Nesterov's accelerated methods) via a minimax framework [1, 43, 12, 23]. In the Fenchel game, a dual best response corresponds to a gradient evaluation:

$$\operatorname*{argmax}_{y \in \mathbb{R}^n} \{\langle x, y\rangle - h^*(y)\} = \nabla h(x),$$

and we show that approximately optimal points for the dual objective (D) must have small norm. As a result, obtaining an approximately optimal dual point y as a best response to a primal point x yields a bound on the norm of $y = \nabla f(x)$. Furthermore, we note that in this setting, adding regularization to ψ with respect to an appropriate choice of

distance-generating function (namely h^*) is equivalent to rescaling and recentering the primal function f, as well as the point at which a gradient is taken in the dual best response computation (cf. Lemma 14 in the full version). Thus, the properties of the Fenchel game extend naturally to appropriately regularized versions of ψ.

1.2 Overview of the framework and analysis

We now give an overview of the dual-extraction framework. Our framework applies generally to a set of assumptions given in Section 3.1 (cf. Definition 9), but for now we specialize to the assumptions given above, namely: (i) the constraint sets $\mathcal{X}$ and $\mathcal{Y}$ are nonempty, compact, and convex; and (ii) ψ is differentiable and convex-concave. Throughout this section, let $\|\cdot\|$ denote any norm on $\mathbb{R}^n$ and assume that the dual function, ϕ, is L-Lipschitz with respect to $\|\cdot\|$.[1] Let $r : \mathbb{R}^n \to \mathbb{R}$ denote a differentiable distance-generating function (dgf) which is μ_r-strongly convex with respect to $\|\cdot\|$ for $\mu_r > 0$,[2] and let $V_u(v) \coloneqq r(v) - r(u) - \langle \nabla r(u), v - u \rangle$ denote the associated Bregman divergence. For the sake of illustration, it may be helpful to consider the choices $\|\cdot\| \coloneqq \|\cdot\|_2$, $r(u) \coloneqq \frac{1}{2}\|u\|_2^2$, $\mu_r = 1$, and $V_u(v) = \frac{1}{2}\|u - v\|_2^2$ in the following, in which case relative strong convexity with respect to r is equivalent to the standard notion of strong convexity with respect to $\|\cdot\|_2$.

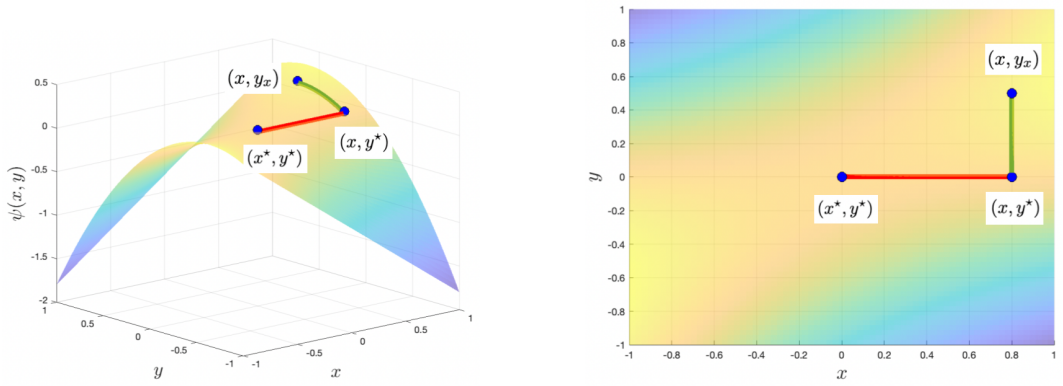
How should we obtain an ϵ-optimal point for (D) using the two oracles discussed previously, namely: (i) an oracle for approximately solving a regularized primal objective, and (ii) an oracle for computing a dual best response? We call (i) a dual-regularized primal optimization (DRPO) oracle and (ii) a dual-regularized best response (DRBR) oracle; their formal definitions are given in Section 3.1. Note that to solve (D), one cannot simply solve the primal problem (P) to high accuracy and then compute a dual best response. Consider $\psi(x, y) = xy$ with $\mathcal{X} = \mathcal{Y} = [-1, 1]$; clearly $x^\star = y^\star = 0$, but for any x arbitrarily close to $x^\star$, the dual best response is either -1 or 1.

The key observation underlying our framework is that if $\psi(x, \cdot)$ is strongly concave for a given $x \in \mathcal{X}$, it is possible to upper bound the distance between the best response $y_x \coloneqq \operatorname{argmax}_{y \in \mathcal{Y}} \psi(x, y)$ and the dual optimum $y^\star$ in terms of the primal suboptimality of x. Figure 1 illustrates why this should be the case when subtracting a quadratic regularizer in y (so that $\psi(x, \cdot)$ is strongly concave) to the preceding example of $\psi(x, y) = xy$. We generalize this intuition in the following lemma (replacing strong concavity with relative strong concavity and a distance bound with a divergence bound), which is itself generalized further and proven in Section 3:

▶ **Lemma 1** (Lemma 3 from the full version specialized). *For a given $x \in \mathcal{X}$, suppose $-\psi(x, \cdot)$ is μ-strongly convex relative to the dgf r for some $\mu > 0$. Then $y_x \coloneqq \operatorname{argmax}_{y \in \mathcal{Y}} \psi(x, y)$ satisfies*

$$V_{y_x}(y^\star) \leq \frac{f(x) - f(x^\star)}{\mu}.$$

[1] This is a weak assumption since we ensure at most a logarithmic dependence on L; see Remark 5.
[2] Section 2 gives the general setup for a distance-generating function which also covers the case where $\operatorname{dom} r \neq \mathbb{R}^n$.



(a) **(b)**

Figure 1 An example to give intuition behind Lemma 1. Here, $\psi(x, y) = xy - 0.8y^2$, $(x^\star, y^\star) = (0, 0)$, $x = 0.8$, and $y_x = 0.5$. To see why it is possible to bound $|y^\star - y_x|$ in terms of the primal suboptimality $f(x) - f(x^\star)$, note that by the strong concavity of $\psi(x, \cdot)$ and the fact that y_x is the maximizer of $\psi(x, \cdot)$ over $\mathcal{Y}$, we can upper bound $|y^\star - y_x|$ in terms of $\psi(x, y_x) - \psi(x, y^\star)$ (the vertical drop over the green line) via a standard strong-concavity inequality. In turn, $\psi(x, y_x) - \psi(x, y^\star)$ can be upper bounded by $\psi(x, y_x) - \psi(x^\star, y^\star) = f(x) - f(x^\star)$ (the vertical drop over the green line plus the vertical drop over the red line) due to the fact that $\psi(x^\star, y^\star) \le \psi(x, y^\star)$ by the optimality of $x^\star$.

A first try

In particular, Lemma 1 suggests the following approach: Define "dual-regularized" versions of ψ, ϕ, f as follows for $\lambda > 0$ and $y_0 \in \mathcal{Y}$:

$$\begin{aligned} \psi_1(x, y) &\coloneqq \psi(x, y) - \lambda V_{y_0}(y), \\ f_1(x) &\coloneqq \max_{y \in \mathcal{Y}} \psi_1(x, y), \\ \phi_1(y) &\coloneqq \min_{x \in \mathcal{X}} \psi_1(x, y)\,. \end{aligned}$$

(Here, the subscript 1 denotes one level of regularization and will be extended later.) For any $x \in \mathcal{X}$, note that $-\psi_1(x, \cdot)$ is λ-strongly convex relative to r, in which case Lemma 1 applied to ψ_1 yields

$$V_{y_x}(y_1^\star) \le \frac{f_1(x) - f_1(x_1^\star)}{\lambda}, \tag{1}$$

for $y_1^\star \coloneqq \operatorname{argmax}_{y \in \mathcal{Y}} \phi_1(y)$, $x_1^\star \in \operatorname{argmin}_{x \in \mathcal{X}} f_1(x)$, and $y_x \coloneqq \operatorname{argmax}_{y \in \mathcal{Y}} \psi_1(x, y)$. Then note

$$\phi(y_1^\star) \ge \phi_1(y_1^\star) \ge \phi_1(y^\star) = \min_{x \in \mathcal{X}} \left\{\psi(x, y^\star) - \lambda V_{y_0}(y^\star)\right\} = \phi(y^\star) - \lambda V_{y_0}(y^\star)\,, \tag{2}$$

where the first inequality follows since $\phi \ge \phi_1$ pointwise. Then by the L-Lipschitzness of ϕ and μ_r-strong convexity of r, it is straightforward to bound the suboptimality of y_x as

$$\phi(y^\star) - \phi(y_x) \le \lambda V_{y_0}(y^\star) + L\sqrt{\frac{2(f_1(x) - f_1(x_1^\star))}{\mu_r \lambda}}. \tag{3}$$

Consequently, an ϵ-optimal point for (D) can be obtained via our oracles as follows: Set $\lambda \leftarrow \frac{\epsilon}{2V_{y_0}(y^\star)}$, and use the DRPO oracle on the regularized primal problem to obtain $x \in \mathcal{X}$ such that

$$f_1(x) - f_1(x_1^\star) \le \frac{\epsilon^3 \mu_r}{16L^2 \cdot V_{y_0}(y^\star)}\,. \tag{4}$$

Then the best response to x with respect to ψ_1, namely $y_x \coloneqq \operatorname{argmax}_{y \in \mathcal{Y}} \psi_1(x, y)$, is ϵ-optimal by (3). However, a typical setting in our applications is $V_{y_0}(y^\star) = \Omega(1)$, $\mu_r = 1$, and $L \geq 1$, in which case ensuring (4) requires solving the regularized primal problem to $O(\epsilon^3)$ error.

Recursive regularization and the dual-extraction framework

To lower the accuracy requirements, we apply dual regularization recursively. A key issue with the preceding argument is that it required a nontrivial bound on $V_{y_0}(y^\star)$. However, it provided us with a nontrivial bound (1) on $V_{y_x}(y_1^\star)$, the "level-one equivalent" of $V_{y_0}(y^\star)$. This suggests solving f_1 to lower accuracy while still obtaining a bound on $V_{y_x}(y_1^\star)$ due to (1), and then adding regularization centered at y_x with a larger value of λ. Indeed, our framework recursively repeats this process until the total regularization is large enough so that (a term similar to) the right-hand side of (3) can be bounded by ϵ, despite never needing to solve a regularized primal problem to high accuracy.

To more precisely describe our approach, let $\psi_0 \coloneqq \psi, f_0 \coloneqq f, \phi_0 \coloneqq \phi$. Over iterations $k = 1, 2, \ldots, K$, our framework implicitly constructs a sequence of convex-concave games $\psi_k : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$, along with corresponding primal and dual functions $f_k : \mathcal{X} \to \mathbb{R}$ and $\phi_k : \mathcal{Y} \to \mathbb{R}$ respectively, as follows:

$$\begin{aligned} \psi_k(x, y) &\coloneqq \psi_{k-1}(x, y) - \lambda_{k-1} V_{y_{k-1}}(y), \\ f_k(x) &\coloneqq \max_{y \in \mathcal{Y}} \psi_k(x, y), \\ \phi_k(y) &\coloneqq \min_{x \in \mathcal{X}} \psi_k(x, y). \end{aligned} \tag{5}$$

Here, $(\lambda_k \in \mathbb{R}_{>0})_{k=0}^{K-1}$ is a dual-regularization schedule given as input to the framework, and $(y_k \in \mathcal{Y})_{i=0}^{K}$ is a sequence of dual-regularization "centers" generated by the algorithm, with y_0 given as input. For $k \in \{0\} \cup [K]$, it will be useful to let $y_k^\star$ denote a maximizer of ϕ_k over $\mathcal{Y}$ and $x_k^\star$ denote a minimizer of f_k over $\mathcal{X}$, with $y_0^\star \coloneqq y^\star$ and $x_0^\star \coloneqq x^\star$ in particular.

Over the K rounds of recursive dual regularization, we aim to balance two goals:

- On the one hand, we want λ_k to increase quickly so that $-\psi_k(x, \cdot)$ is very strongly convex relative to r, thereby allowing us to apply Lemma 1 with a larger strong convexity constant.
- On the other hand, we want to maintain the invariant that, roughly speaking, $y_k^\star$ is always $\epsilon/2$-optimal for the original dual ϕ. Indeed, we were constrained in choosing λ in (2) to be on the order of $\epsilon / V_{y_0}(y^\star)$ to ensure $y_1^\star$ is $\epsilon/2$-optimal for ϕ. A similar "constraint" on the dual-regularization schedule $(\lambda_k)_{k=0}^{K-1}$ appears when (2) is extended to additional levels of regularization. This prevents us from increasing λ_k too quickly.

In all the applications in this paper we choose $\lambda_k \approx 2\lambda_{k-1}$. λ_0 typically must remain on the order of $\epsilon / V_{y_0}(y^\star)$ due to the second point.

Pseudocode of the framework is given in Algorithm 1. Each successive dual-regularization center y_k is computed via the DRBR oracle (Line 5) as a best response to a primal point x_k obtained via the DRPO oracle (Line 4). In Section 3, we generalize Algorithm 1 (cf. Algorithm 2) in several ways: (i) we allow for stochasticity in the DRPO oracle; (ii) we allow for distance-generating functions r such that $\operatorname{dom} r \neq \mathbb{R}^n$; (iii) we give different but equivalent characterizations of x_k and y_k which facilitate the derivation of explicit expressions for the DRPO and DRBR oracles in applications.

Algorithm 1 Dual-extraction framework (Algorithm 2 specialized).

Input: Initial dual-regularization center $y_0 \in \mathcal{Y}$, iteration count $K \in \mathbb{N}$, dual-regularization schedule $(\lambda_k \in \mathbb{R}_{>0})_{k=0}^{K-1}$, primal-accuracy schedule $(\epsilon_k \in \mathbb{R}_{>0})_{k=1}^{K}$, DRPO and DRBR oracles

$\psi_0 \coloneqq \psi$, $f_0 \coloneqq f$, and $\phi_0 \coloneqq \phi$

for $k = 1, 2, \ldots, K$ **do**

  Define ψ_k, f_k, and ϕ_k as in (5)

  Let $x_k \in \mathcal{X}$ be such that $f_k(x_k) - f_k(x_k^\star) \le \epsilon_k$ // Computed via the DRPO oracle

  $y_k = \operatorname{argmax}_{y \in \mathcal{Y}} \psi_k(x_k, y)$ // Computed via the DRBR oracle

return y_K

Analysis of Algorithm 1

Theorem 2 is our main result for Algorithm 1. We then instantiate Theorem 2 with two illustrative choices of parameters in Corollaries 5 and 7, and defer the proofs of the latter to their general versions in Section 3. All of the remarks below (Remarks 3, 5, 7) are stated with reference to the specialized results in this section (Theorem 2 and Corollaries 4, 6 resp.), but extend immediately to the corresponding general versions (Theorem 15 and Corollaries 16, 17 resp.).

▶ **Theorem 2** (Theorem 15 specialized). *Algorithm 1 returns y_K satisfying*

$$V_{y_K}(u) \le \frac{\epsilon_K}{\Lambda_K} \text{ where } \Lambda_k \coloneqq \sum_{j=0}^{k-1} \lambda_j \text{ for } k \in [K] \tag{6}$$

and $u \in \mathcal{Y}$ is a point with dual suboptimality bounded as

$$\phi(y^\star) - \phi(u) \le \lambda_0 V_{y_0}(y^\star) + \sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k} \epsilon_k. \tag{7}$$

If we additionally assume that ϕ is L-Lipschitz with respect to $\|\cdot\|$, we can directly bound the suboptimality of y_K as

$$\phi(y^\star) - \phi(y_K) \le \lambda_0 V_{y_0}(y^\star) + \sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k} \epsilon_k + L\sqrt{\frac{2}{\mu_r} \frac{\epsilon_K}{\Lambda_K}}. \tag{8}$$

Proof. We claim the first half of Theorem 2 holds with $u \leftarrow y_K^\star$. To see this, note that we can bound the suboptimality of $y_K^\star$ as

$$\begin{aligned}
\phi(y_K^\star) \overset{(i)}{\ge} \phi_K(y_K^\star) \ge \phi_K(y_{K-1}^\star) &= \max_{x \in \mathcal{X}} \left\{ \psi_{K-1}(x, y_{K-1}^\star) - \lambda_{K-1} V_{y_{K-1}}\left(y_{K-1}^\star\right) \right\} \\
&= \phi_{K-1}(y_{K-1}^\star) - \lambda_{K-1} V_{y_{K-1}}\left(y_{K-1}^\star\right) \\
&\overset{(ii)}{\ge} \phi_0(y_0^\star) - \lambda_0 V_{y_0}(y_0^\star) - \sum_{k=1}^{K-1} \lambda_k V_{y_k}(y_k^\star) \\
&\overset{(iii)}{\ge} \phi(y^\star) - \lambda_0 V_{y_0}(y^\star) - \sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k} \epsilon_k,
\end{aligned}$$

where (i) follows since $\phi \geq \phi_K$ pointwise, (ii) follows from repeating the argument in the previous lines recursively (starting by lower bounding $\phi_{K-1}(y^\star_{K-1})$, etc.), and (iii) uses Lemma 1 applied to ψ_k, which yields by Lines 4 and 5 in Algorithm 1:

$$V_{y_k}(y^\star_k) \leq \frac{f_k(x_k) - f_k(x^\star_k)}{\Lambda_k} \leq \frac{\epsilon_k}{\Lambda_k},$$

since $\psi_k(x, \cdot) = \psi(x, \cdot) + \sum_{j=0}^{k-1} \lambda_j V_{y_j}(\cdot)$ is Λ_k-strongly concave relative to $-r$. Thus, we have proven Equation 7, and Equation 6 follows again from Lemma 1 applied to ψ_K. Equation 8 then follows since the fact that r is μ_r-strongly convex with respect to $\|\cdot\|$ and Equation 6 imply

$$\|y_K - y^\star_K\| \leq \sqrt{\frac{2}{\mu_r} V_{y_K}(y^\star_K)} \leq \sqrt{\frac{2}{\mu_r} \frac{\epsilon_K}{\Lambda_K}}. \qquad \blacktriangleleft$$

We give a remark regarding how to pick the parameters $(\lambda_k)_{k=0}^{K-1}$ and $(\epsilon_k)_{k=1}^{K}$ when applying Theorem 2:

▶ Remark 3 (Picking the parameters for Theorem 2). Equation 8 can be interpreted as follows: To ensure y_K is ϵ-optimal for ϕ, it suffices to choose the sequences $(\lambda_k)_{k=0}^{K-1}$ and $(\epsilon_k)_{k=1}^{K}$ so that the right side of (8) is at most ϵ. Then the first term, $\lambda_0 V_{y_0}(y^\star)$, constrains λ_0 to be on the order of $\epsilon / V_{y_0}(y^\star)$. Skipping ahead, the third term, $L\sqrt{\frac{2}{\mu_r}\frac{\epsilon_K}{\Lambda_K}}$, is the reason we always choose $\lambda_k \approx 2\lambda_{k-1}$ in our applications, as this ensures Λ_K is large enough to handle this term with K only needing to be logarithmic in the problem parameters. Then the second term, $\sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k}\epsilon_k$, effectively constrains roughly $\sum_{k=1}^{K-1} \epsilon_k \leq \epsilon$, as $\lambda_k / \Lambda_k \approx 1$.

▶ **Corollary 4** (Corollary 16 specialized). *Suppose ϕ is L-Lipschitz with respect to $\|\cdot\|$, and let $B > 0$ be such that $V_{y_0}(y^\star) \leq B$. Then for any $\epsilon > 0$, and $K \geq \max\left\{\log_2 \frac{L^2 B}{\mu_r \epsilon^2}, 1\right\} + 10$, the output of Algorithm 1 with dual-regularization and primal-accuracy schedules of*

$$\lambda_k = 2^k \frac{\epsilon}{4B} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\epsilon}{4K} \text{ for } k \in [K]$$

satisfies $\phi(y^\star) - \phi(y_K) \leq \epsilon$.

▶ Remark 5. Corollary 4 achieves the stated goal of obtaining an ϵ-optimal point for (D) by running for a number of iterations which depends logarithmically on the problem parameters, and solving each dual-regularized primal subproblem to an accuracy of ϵ divided by a logarithmic factor. Note in particular the logarithmic dependence on the dual divergence bound B and dual Lipschitz constant L, meaning these are weak assumptions. Furthermore, it is clear from the proof of Theorem 2 that ϕ only need be L-Lipschitz on a set containing y_K and $y^\star_K$.

▶ **Corollary 6** (Corollary 17 specialized). *Let $B > 0$ be such that $V_{y_0}(y^\star) \leq B$. Then for any $\epsilon > 0$ and $K \in \mathbb{N}$, the output of Algorithm 1 with dual-regularization and primal-accuracy schedules of*

$$\lambda_k = 2^k \frac{\epsilon}{4B} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\epsilon}{8 \cdot 1.5^k} \text{ for } k \in [K]$$

satisfies

$$\|y_K - u\| \leq \frac{1}{1.5^K} \sqrt{\frac{2B}{\mu_r}},$$

where $u \in \mathcal{Y}$ is a point whose suboptimality is at most ϵ, i.e., $\phi(y^\star) - \phi(u) \leq \epsilon$.

▶ Remark 7. Later calls to the DRPO oracle during the run of Algorithm 1 may be cheaper since there will be a significant amount of dual regularization at that point (namely, $\Lambda_k = \sum_{j=0}^{k-1} \lambda_j$ is large). One can sometimes take advantage of this (in particular, if the cost of a DRPO oracle call scales inverse polynomially with the regularization) to design schedules that avoid even the typically additional multiplicative logarithmic cost of Corollary 4 over the cost of a single DRPO oracle call. In such cases, a choice of schedules similar to those of Corollary 6 is often appropriate. With this choice of schedules, later rounds require very high accuracy. However, if one can argue that the increasing dual regularization Λ_k makes the DRPO oracle call cheaper at a faster rate than the decreasing error ϵ_k makes it more expensive (as we do in Section 5), the total cost of applying the framework may collapse geometrically to the cost of a single DRPO oracle call made with target error approximately ϵ.

We purposely state Corollary 6 without the assumption that ϕ is Lipschitz because that is the form we will use in Section 5. However, it is straightforward to reformulate a version of Corollary 6 with the Lipschitz assumption. Here the focus was to illustrate different possible choices of schedules.

1.3 Related work

Black-box reductions

Our main contribution can be viewed as a black-box reduction from (regularized) primal optimization to dual optimization. Similar black-box reductions exist in the optimization literature. For example, [3] develops reductions between various fundamental classes of optimization problems, e.g., strongly convex optimization and smooth optimization. In a similar vein, the line of work [30, 18, 7] reduces convex optimization to approximate proximal point computation (i.e., regularized minimization).

Bilinear matrix games

Consider the bilinear objective $\psi(x, y) = x^\top A y$ where $\mathcal{X}$ and $\mathcal{Y}$ are either the simplex, $\Delta^k := \{x \in \mathbb{R}^k_{\geq 0} : \|x\|_1 = 1\}$, or the Euclidean ball, $B^k := \{x \in \mathbb{R}^k : \|x\|_2 \leq 1\}$. State-of-the-art methods in regard to runtime for obtaining an approximately optimal primal and/or dual solution can be divided into second-order interior point methods [11, 42] and stochastic first-order methods [22, 10, 9, 8]; see Table 2 in [8] for a summary of the best known runtimes as well as other references. Of importance to this paper, all state-of-the-art algorithms other than that of [8] are either (i) primal-dual algorithms which return both an ϵ-optimal primal and dual solution simultaneously, and/or (ii) achieve runtimes which are symmetric in the primal dimension d and dual dimension n, meaning the cost of obtaining an ϵ-optimal dual solution is the same as that of obtaining an ϵ-optimal primal solution. The algorithm of [8], on the other hand, only returns an ϵ-optimal primal point and further has a runtime which is not symmetric in n and d (see the footnote on the first page of that paper). As a result, solving the dual by simply swapping the roles of the primal and dual variables may be more expensive than solving the primal. (In fact, swapping the variables in this way may not even always be possible without further modifications due to restrictions on the constraint sets.)

CVaR at level α distributionally robust optimization (DRO)

The DRO objectives we study are of the form $\psi(x, y) = \sum_{i=1}^n y_i f_i(x)$, where the functions $f_i : \mathbb{R}^d \to \mathbb{R}$ are convex, bounded, and Lipschitz, and $\mathcal{Y}$, known as the uncertainty set, is a subset of the simplex. This objective corresponds to a robust version of the empirical risk

minimization (ERM) objective where instead of taking an average over the losses (namely, y_i is fixed at $1/n$), larger losses may be given more weight. In particular, in this paper we focus on a canonical DRO setting, *CVaR at level α,* where the uncertainty set is given by $\mathcal{Y} \coloneqq \left\{y \in \Delta^n : \|y\|_\infty \leq \frac{1}{\alpha n}\right\}$ for a choice of $\alpha \in [1/n, 1]$. CVaR DRO, along with its generalization f-divergence DRO, has been of significant interest over the past decade; see [29, 5, 13, 32, 16] and the references therein. [29] is the most relevant to this paper – omitting parameters other than α, the number of losses n, and the target accuracy $\epsilon > 0$, they give a matching upper and lower bound (up to logarithmic factors) of $\widetilde{O}(\alpha^{-1}\epsilon^{-2})$ first-order queries of the form $(\nabla f_i(x), f_i(x))$ to obtain an expected ϵ-optimal point of the primal objective. Their upper bound is achieved by a stochastic gradient method where the gradient estimator is based on a multilevel Monte Carlo (MLMC) scheme [19, 20]. However, the best known complexity for obtaining an expected ϵ-optimal point of the dual of CVaR at level α is $O(n\epsilon^{-2})$ via a primal-dual method based on [33]; see also [13, 32] as well as [5, Appendix A.1], the last of which obtains complexity $\widetilde{O}(n\epsilon^{-2})$ in the more general setting of the uncertainty set being an f-divergence ball.

Stationary point computation

For $\gamma > 0$, convex and β-smooth $h : \mathbb{R}^n \to \mathbb{R}$ with global minimum $z^\star$, and initialization point z_0, consider the problem of computing a point z such that $\|\nabla h(z)\|_2 \leq \gamma$. Two worst-case optimal gradient query complexities for this problem exist in the literature: $O\left(\sqrt{\beta(h(z_0) - h(z^\star))}/\gamma\right)$ and $O\left(\sqrt{\beta\|z_0 - z^\star\|_2}/\gamma\right)$. An algorithm (the OGM-G method) which achieves the former complexity was given in [24], and [37] pointed out that any algorithm which achieves the former complexity can achieve the latter complexity. This is obtainable by running N iterations of any optimal gradient method for reducing the function value, followed by N iterations of a method which achieves the former complexity for reducing the gradient magnitude. In what may be of independent interest, we observe in Section 5.1 that a reduction in the opposite direction is also possible. More broadly, algorithms and frameworks for reducing the gradient magnitude of convex functions have been of much recent interest, and further algorithms and related work for this problem include [25, 27, 26, 28, 15, 37, 21], with lower bounds given in [34, 35].

1.4 Paper organization

In Section 2, we go over notation and conventions for the rest of the paper. We give our general dual-extraction framework and its guarantees in Section 3. In Section 4, we apply our framework to bilinear matrix games and the CVaR at level α DRO problem. Finally, in Section 5 we give an optimal algorithm (in terms of query complexity) for computing an approximate stationary point of a convex and β-smooth function.

2 Notation and conventions

We defer standard notation and conventions to the full version, and only include paper-specific notation here.

For $\psi : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$, we use the notation $\psi(\cdot, y) : \mathbb{R}^d \to \mathbb{R}$ for a fixed $y \in \mathbb{R}^n$ to denote the map $x \mapsto \psi(x, y)$ (and define $\psi(x, \cdot)$ analogously). When we say $\psi(\cdot, y)$ satisfies a property, we mean it satisfies that property for any fixed $y \in \mathbb{R}^n$ (and analogously for $\psi(x, \cdot)$). We let $[K] \coloneqq \{1, 2, \ldots, K\}$, $\Delta^n \coloneqq \left\{x \in \mathbb{R}^n_{\geq 0} : \|x\|_1 = 1\right\}$, and $B^n_r(x) \coloneqq \{x \in \mathbb{R}^n : \|x\|_2 \leq r\}$. In the latter two definitions, we may drop the superscript n if it is clear from context, the

argument x if it is 0, and the subscript r if it is 1. For $y \in \mathbb{R}^n$, we may use either the notation y_i or $[y]_i$ to denote its i-th entry. $\mathbf{1}$ denotes the all-ones vector. For a function f which depends on some inputs $x_1, \ldots, x_k \in \mathbb{R}$, we write $f \leq \mathrm{poly}(x_1, \ldots, x_k)$ to denote the fact that f is uniformly bounded above by a polynomial in $x_1, \ldots, x_k$ as $x_1, \ldots, x_k$ vary. We use the notation f^* for the convex or Fenchel conjugate of f. For $S \subseteq \mathbb{R}^n$, we let $\mathbb{I}_S$ denote the infinite indicator of S, namely $\mathbb{I}_S(x) = 0$ if $x \in S$ and $\mathbb{I}_S(x) = \infty$ if $x \notin S$. For a function $f : S \to [-\infty, \infty]$ initially defined on a strict subset $S \subset \mathbb{R}^n$, we may implicitly extend the domain of f to all of $\mathbb{R}^n$ via its indicator as $f + \mathbb{I}_S$ without additional comment. For a function $f : U \to [-\infty, \infty]$ with $S \subseteq U \subseteq \mathbb{R}^n$, we let $f_S \coloneqq f + \mathbb{I}_S$ denote the restriction of f to S. We note that f_S^* denotes the convex conjugate of f_S (and not f^* restricted to S).

Following [38, Sec. 6.4], we encapsulate the setup for a dgf as follows. See the full version for additional discussion of this definition.

▶ **Definition 8** (dgf setup). *We say $(\mathcal{U}, \mathcal{P}, \|\cdot\|, r)$ is a* dgf setup *over $\mathbb{R}^n$ for closed and convex sets $\mathcal{U} \subseteq \mathcal{P} \subseteq \mathbb{R}^n$ with $\mathcal{U} \cap \operatorname{int} \mathcal{P} \neq \emptyset$ if: (i) the distance-generating function (dgf) $r : \mathcal{P} \to \mathbb{R}$ is convex and continuous over $\mathcal{P}$, differentiable on $\operatorname{int} \mathcal{P}$, and μ_r-strongly convex with respect to the chosen norm $\|\cdot\|$ on $\mathcal{U} \cap \operatorname{int} \mathcal{P}$ for some $\mu_r > 0$; and (ii) either $\lim_{u \to \operatorname{bd} \mathcal{P}} \|\nabla r(u)\|_2 = \infty$ or $\mathcal{U} \subseteq \operatorname{int} \mathcal{P}$.*

For a given dgf setup, we define its induced Bregman divergence $V_u^r(v) \coloneqq r(v) - r(u) - \langle \nabla r(u), v - u \rangle$ for $u \in \operatorname{int} \mathcal{P}, v \in \mathcal{P}$, and drop the superscript r when it is clear from context.

3 Dual-extraction framework

In this section, we provide our general dual-extraction framework and its guarantees. In Section 3.1, we give the general setup, oracle definitions, and assumptions with which we apply and analyze the framework. Section 3.2 contains the statement and guarantees of the framework and Section 3.3 in the full version contains the associated proofs.

3.1 Preliminaries

We bundle all of the inputs to our framework into what we call a *dual-extraction setup*, defined below. Recall that when we say $\psi(x, \cdot)$ satisfies a property, we mean it satisfies that property for any fixed $x \in \mathbb{R}^d$ (and analogously for $\psi(\cdot, y)$).

▶ **Definition 9** (Dual-extraction setup). *A* dual-extraction setup *is a tuple $(\psi, \mathcal{X}, \mathcal{Y}, \mathcal{U}, \mathcal{P}, \|\cdot\|, r)$ where:*

1. *$\psi(x, \cdot)$ is differentiable;*
2. *$\psi(\cdot, y)$ and $\psi(x, \cdot)$ are convex and concave respectively;*
3. *$(\mathcal{U}, \mathcal{P}, \|\cdot\|, r)$ is a dgf setup over $\mathbb{R}^n$ per Definition 8;*
4. *the constraint sets $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}^n$ are nonempty, closed, and convex with $\mathcal{Y} \subseteq \mathcal{U}$ and $\mathcal{Y} \cap \operatorname{int} \mathcal{P} \neq \emptyset$;*
5. *$\mathcal{X}$ is bounded or $\psi(\cdot, y)$ is strongly convex;*
6. *$\mathcal{Y}$ is bounded or $\psi(x, \cdot)$ is strongly concave;*
7. *over all $p \in \mathcal{U} \cap \operatorname{int} \mathcal{P}$ and $w \in \partial \mathbb{I}_{\mathcal{U}}(p)$, the map $y \mapsto \langle w, y \rangle$ is constant over $\mathcal{Y}$.*[3]

[3] In all of our applications, this map will in fact be constant over $\mathcal{U}$.

Assumption 1 is only used in the proofs of Lemma 3 in the full version (the general version of Lemma 1 from Section 1.2) and Corollary 13 in the full version (used to show the framework is well-defined when $\operatorname{dom} r \neq \mathbb{R}^n$). Assumptions 2, 5, and 6 ensure that the minimax optimization problem with objective ψ and constraint sets $\mathcal{X}$ and $\mathcal{Y}$ satisfies the *minimax principle*; see below. Regarding Assumptions 3, 4, and 7, the fact that $\mathcal{Y}$ is potentially a strict subset of $\mathcal{U}$ as well as the necessity of the technical assumption 7 is discussed in Remark 4 in the full version. In particular, Assumption 7 is only used to derive an equivalent formulation of the framework to Algorithm 1 which often allows for easier instantiations in applications, but is not strictly necessary to obtain our guarantees.

While our main results are stated in the full generality of Definition 9, in our applications we only particularize to Definition 10 and Definition 11 introduced below.

▶ **Definition 10** (Unbounded setup). *A $(\psi, \mathcal{X}, \mathcal{Y}, r)$-unbounded setup is a $(\psi, \mathcal{X}, \mathcal{Y}, \mathbb{R}^n, \mathbb{R}^n, \|\cdot\|_2, r)$-dual-extraction setup.*

In other words, in an unbounded setup we choose $\mathcal{U} = \mathcal{P} = \mathbb{R}^n$ and the Euclidean norm, in which case the dgf r can be any differentiable and strongly convex function with respect to $\|\cdot\|_2$. Note that Assumption 7 is trivial as $\partial \mathbb{I}_{\mathcal{U}}(p) = \{0\}$ for all $p \in \mathbb{R}^n$.

▶ **Definition 11** (Simplex setup). *A $(\psi, \mathcal{X}, \mathcal{Y})$-simplex setup is a $(\psi, \mathcal{X}, \mathcal{Y}, \Delta^n, \mathbb{R}^n_{\geq 0}, \|\cdot\|_1, r)$-dual-extraction setup where $r(u) \coloneqq \sum_{i=1}^n u_i \ln u_i$ (with $0 \ln 0 \coloneqq 0$).*

In other words, in a simplex setup we choose $\mathcal{U} = \Delta^n$, $\mathcal{P} = \mathbb{R}^n_{\geq 0}$, we are using the ℓ_1-norm, and the dgf is negative entropy when restricted to the simplex. It is a standard result known as Pinsker's inequality that r is 1-strongly convex over $\Delta^n_{>0}$ with respect to $\|\cdot\|_1$, and the associated Bregman divergence is given by the Kullback-Leibler (KL) divergence $V_u(w) = \sum_{i=1}^n w_i \ln \frac{w_i}{u_i}$ for $u \in \Delta^n_{>0}$ and $w \in \Delta^n$. We verify that Assumption 7 holds in Appendix A.1 in the full version.

Notation associated with a setup

Whenever we instantiate a dual-extraction setup (Definition 9), we use the following notation and oracles associated with that setup without additional comment. We define the associated primal $f : \mathcal{X} \to \mathbb{R}$ and dual $\phi : \mathcal{Y} \to \mathbb{R}$ functions, along with their corresponding primal and dual optimization problems, as they were introduced above in (P) and (D). We let $x^\star \in \operatorname{argmin}_{x \in \mathcal{X}} f(x)$ and $y^\star \in \operatorname{argmax}_{y \in \mathcal{Y}} \phi(y)$ denote arbitrary primal and dual optima. To facilitate the discussion of dual-regularized problems, we define $f_{\lambda,q}(x) : \mathcal{X} \to \mathbb{R}$ as follows

$$f_{\lambda,q}(x) \coloneqq \max_{y \in \mathcal{Y}} \{\psi(x, y) - \lambda V_q(y)\} \text{ for } \lambda > 0 \text{ and } q \in \mathcal{U} \cap \operatorname{int} \mathcal{P}\,.$$

The minimax principle

Assumptions 2, 5, and 6 in Definition 9 guarantee $f(x^\star) = \psi(x^\star, y^\star) = \phi(y^\star)$, which we refer to as the *minimax principle*. See, e.g., [39, 41] as well as Propositions 1.2 and 2.4 in [17, Ch. VI].

Oracle definitions

Our framework assumes black-box access to ψ, $\mathcal{X}$, and $\mathcal{Y}$ via a dual-regularized primal optimization (DRPO) oracle and a dual-regularized dual best response (DRBR) oracle defined below. Note that we generalize the setting of Section 1.2 by allowing the DRPO oracle to return an expected ϵ-optimal point; this is used in our applications in Section 4.

▶ **Definition 12** (DRPO oracle). *A* $(q \in \mathcal{U} \cap \operatorname{int} \mathcal{P}, \lambda > 0, \epsilon_{\mathrm{p}} > 0)$*-dual-regularized primal optimization oracle,* $DRPO(q, \lambda, \epsilon_{\mathrm{p}})$*, returns an expected* ϵ_{p}*-minimizer of* $f_{\lambda,q}$*, i.e., a point* $x \in \mathcal{X}$ *such that* $\mathbb{E} f_{\lambda,q}(x) \le \inf_{x' \in \mathcal{X}} f_{\lambda,q}(x') + \epsilon_{\mathrm{p}}$*, where the expectation is over the internal randomness of the oracle.*

▶ **Definition 13** (DRBR oracle). *A* $(q \in \mathcal{U} \cap \operatorname{int} \mathcal{P}, \lambda > 0, x \in \mathcal{X})$*-dual-regularized best response oracle,* $DRBR(q, \lambda, x)$*, returns* $\operatorname{argmax}_{y \in \mathcal{Y}} \left\{ \psi(x, y) - \lambda V_q(y) \right\}$.

We also define a version of the DRPO oracle, called the DRPOSP oracle, which allows for a failure probability. We include this definition here due to its generality and broad applicability, but it is only used in Section 4.1 since the external result we cite to obtain an expected ϵ_{p}-minimizer of $f_{\lambda,q}$ in that application has a failure probability. We also show in Appendix A.4 in the full version how to boost the success probability of a DRPOSP oracle.

▶ **Definition 14** (DRPOSP oracle). *A* $(q \in \mathcal{U} \cap \operatorname{int} \mathcal{P}, \lambda > 0, \epsilon_{\mathrm{p}} > 0, \delta \in [0, 1))$*-dual-regularized primal optimization oracle with success probability,* $DRPOSP(q, \lambda, \epsilon_{\mathrm{p}}, \delta)$*, returns an expected* ϵ_{p}*-minimizer of* $f_{\lambda,q}$ *with success probability at least* $1 - \delta$*, where the expectation and success probability are over the internal randomness of the oracle.*

3.2 The framework and its guarantees

Algorithm 2 Dual-extraction framework.

Input: $(\psi, \mathcal{X}, \mathcal{Y}, \mathcal{U}, \mathcal{P}, \|\cdot\|, r)$-dual extraction setup (Definition 9), initial dual-regularization center $y_0 \in \mathcal{Y} \cap \operatorname{int} \mathcal{P}$, iteration count $K \in \mathbb{N}$, dual-regularization schedule $(\lambda_k \in \mathbb{R}_{>0})_{k=0}^{K-1}$, primal-accuracy schedule $(\epsilon_k \in \mathbb{R}_{>0})_{k=1}^{K}$, DRPO and DRBR oracles (Definitions 12 and 13)

1 **for** $k = 1, 2, \ldots, K$ **do**
2 $\quad \Lambda_k = \sum_{j=0}^{k-1} \lambda_j$
3 $\quad q_k = \operatorname{argmin}_{q \in \mathcal{U}} \frac{1}{\Lambda_k} \sum_{j=0}^{k-1} \lambda_j V_{y_j}(q)$ // Or, $q_k = \nabla r^*_{\mathcal{U}} \left(\frac{1}{\Lambda_k} \sum_{j=0}^{k-1} \lambda_j \nabla r(y_j) \right)$; see Appendix A.2 in the full version
4 $\quad x_k = \mathrm{DRPO}(q_k, \Lambda_k, \epsilon_k)$ // $\mathbb{E}[f_{\Lambda_k, q_k}(x_k) \mid x_{k-1}] \le \inf_{x \in \mathcal{X}} f_{\Lambda_k, q_k}(x) + \epsilon_k$
5 $\quad y_k = \mathrm{DRBR}(q_k, \Lambda_k, x_k)$ // $y_k = \operatorname*{argmax}_{y \in \mathcal{Y}} \left\{ \psi(x_k, y) - \Lambda_k V_{q_k}(y) \right\}$
6 **return** y_K

We now state the general dual-extraction framework, Algorithm 2, and its guarantees, with proofs in the next section. As mentioned in Section 1.2, Algorithm 2 generalizes Algorithm 1 in three major ways: (i) we allow for stochasticity in the DRPO oracle; (ii) we allow for distance-generating functions r where $\operatorname{dom} r \neq \mathbb{R}^n$; and (iii) we give different but equivalent characterizations of x_k and y_k which often allow for easier instantiations of the framework.

Regarding (iii), consider the case where the DRPO oracle is deterministic and $\operatorname{dom} r = \mathbb{R}^n$ for the sake of discussion. Note that in this case, the definitions of x_k and y_k in Lines 4 and 5 of Algorithm 2 may seem different than those in Lines 4 and 5 of Algorithm 1 at first glance. In particular, x_k in Line 4 of Algorithm 2 is an ϵ_k-minimizer of $x \mapsto \max_{y \in \mathcal{Y}} \{\psi(x, y) - \Lambda_k V_{q_k}(y)\}$ over $\mathcal{X}$, whereas x_k in Line 4 of Algorithm 1 is an ϵ_k-minimizer of $x \mapsto \max_{y \in \mathcal{Y}} \{\psi(x, y) - \sum_{j=0}^{k-1} \lambda_j V_{y_j}(y)\}$ over $\mathcal{X}$. Similarly, $y_k = \operatorname{argmax}_{y \in \mathcal{Y}} \{\psi(x_k, y) - \Lambda_k V_{q_k}(y)\}$ in Line 5 of Algorithm 2, whereas $y_k = \operatorname{argmax}_{y \in \mathcal{Y}} \{\psi(x, y) - \sum_{j=0}^{k-1} \lambda_j V_{y_j}(y)\}$ in Line 5 of Algorithm 1.

In fact, we show in Section 3.3 in the full version that these are equivalent; see Lemma 2 and Remark 4 in the full version. The potential advantage of the expressions in Algorithm 2 compared to those in Algorithm 1 is that they involve only a single regularization term.

Note also that Line 3 of Algorithm 2 gives two equivalent expressions for the iterate q_k; their equivalence is proven in Appendix A.2 in the full version. Also, note that Line 4 is the only potential source of randomness in Algorithm 2; in particular, y_k and q_{k+1} are deterministic upon conditioning on x_k. Finally, we show that Algorithm 2 is well-defined in Appendix A.3 in the full version; in particular, whenever a Bregman divergence $V_u(w)$ is written in Algorithm 2, it is the case that $u \in \mathcal{U} \cap \operatorname{int} \mathcal{P}$. For example, in the context of a simplex setup per Definition 11, this corresponds to $u \in \Delta^n_{>0}$.

We now give the main guarantee for Algorithm 2. See Remark 3 for additional explanation.

▶ **Theorem 15** (Algorithm 2 guarantee)**.** *With K calls to a DRPO oracle and K calls to a DRBR oracle, Algorithm 2 returns y_K satisfying*

$$\mathbb{E}V_{y_K}(u) \le \frac{\epsilon_K}{\Lambda_K},$$

where $u \in \mathcal{Y}$ is a point with expected suboptimality bounded as

$$\phi(y^\star) - \mathbb{E}\phi(u) \le \lambda_0 V_{y_0}(y^\star) + \sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k}\epsilon_k.$$

If we additionally assume that ϕ is L-Lipschitz with respect to $\|\cdot\|$, the expected suboptimality of y_K can be directly bounded as

$$\phi(y^\star) - \mathbb{E}\phi(y_K) \le \lambda_0 V_{y_0}(y^\star) + \sum_{k=1}^{K-1} \frac{\lambda_k}{\Lambda_k}\epsilon_k + L\sqrt{\frac{2}{\mu_r}\frac{\epsilon_K}{\Lambda_K}}. \tag{9}$$

We now particularize Theorem 15 using two exemplary choices of the dual-regularization and primal-accuracy schedules. See Remarks 5 and 7 for additional comments.

▶ **Corollary 16.** *Suppose ϕ is L-Lipschitz with respect to $\|\cdot\|$, and let $B > 0$ be such that $V_{y_0}(y^\star) \le B$. Then for any $\epsilon > 0$, and $K \ge \max\left\{\log_2 \frac{L^2 B}{\mu_r \epsilon^2}, 1\right\} + 10$, the output of Algorithm 2 with dual-regularization and primal-accuracy schedules given by*

$$\lambda_k = 2^k \frac{\epsilon}{4B} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\epsilon}{4K} \text{ for } k \in [K]$$

satisfies $\phi(y^\star) - \mathbb{E}\phi(y_K) \le \epsilon$.

▶ **Corollary 17.** *Let $B > 0$ be such that $V_{y_0}(y^\star) \le B$. Then for any $\epsilon > 0$ and $K \in \mathbb{N}$, the output of Algorithm 2 with dual-regularization and primal-accuracy schedules given by*

$$\lambda_k = 2^k \frac{\epsilon}{4B} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\epsilon}{8 \cdot 1.5^k} \text{ for } k \in [K] \tag{10}$$

satisfies

$$\mathbb{E}\|y_K - u\| \le \frac{1}{1.5^K}\sqrt{\frac{2B}{\mu_r}},$$

where $u \in \mathcal{Y}$ is a point whose expected suboptimality is at most ϵ, i.e., $\phi(y^\star) - \mathbb{E}\phi(u) \le \epsilon$.

4 Efficient maximin algorithms

In this section, we obtain new state-of-the-art runtimes for solving bilinear matrix games in certain parameter regimes (Section 4.1), as well as an improved query complexity for solving the dual of the CVaR at level α distributionally robust optimization (DRO) problem (Section 4.2). In each application, we apply Corollary 16 to compute an ϵ-optimal point for the dual problem at approximately the same cost as computing an ϵ-optimal point for the primal problem (up to logarithmic factors and the cost of representing a dual vector when it comes to CVaR at level α).

4.1 Bilinear matrix games

In this section, we instantiate $\psi(x, y) \coloneqq x^\top A y$ for a matrix $A \in \mathbb{R}^{d\times n}$. Given $p, q \geq 1$, we write $\|A\|_{p\to q} \coloneqq \max_{v\in\mathbb{R}^d, v\neq 0} \frac{\|Av\|_q}{\|v\|_p}$, and use the notation A_{ij}, $A_{i:}$, and $A_{:j}$ for the (i, j) entry, i-th row as a row vector, and j-th column as a column vector. We consider two setups:

▶ **Definition 18** (Matrix games ball setup)**.** *In the* matrix games ball setup, *we set* $\mathcal{X} \coloneqq B^d$ *(the unit Euclidean ball in* $\mathbb{R}^d$*),* $\mathcal{Y} \coloneqq \Delta^n$*, and fix a* $(\psi, \mathcal{X}, \mathcal{Y})$*-simplex setup (Definition 11). We assume* $\|A^\top\|_{2\to\infty} = \max_{i\in[n]} \|A_{:i}\|_2 \leq 1$.

▶ **Definition 19** (Matrix games simplex setup)**.** *In the* matrix games simplex setup, *we set* $\mathcal{X} \coloneqq \Delta^d$, $\mathcal{Y} \coloneqq \Delta^n$, *and fix a* $(\psi, \mathcal{X}, \mathcal{Y})$*-simplex setup (Definition 11). We assume* $\|A^\top\|_{1\to\infty} = \max_{i,j} |A_{ij}| \leq 1$.

Throughout Section 4.1, any theorem, statement, or equation which does not make reference to a specific choice of Definition 18 or 19 applies to both setups. Specializing the primal (P) and dual (D) to this application gives

$$\underset{x\in\mathcal{X}}{\text{minimize}}\, f(x) \text{ for } f(x) \coloneqq \max_{y\in\Delta^n} x^\top A y, \text{ and} \tag{P-MG}$$

$$\underset{y\in\Delta^n}{\text{maximize}}\, \phi(y) \text{ for } \phi(y) \coloneqq \min_{x\in\mathcal{X}} x^\top A y. \tag{D-MG}$$

Regarding the assumptions on the norm of the matrix A in Definitions 18 and 19, note that we can equivalently write $f(x) = \max_{y\in\Delta^n} \sum_{i=1}^n y_i f_i(x)$ with $f_i(x) \coloneqq [A^\top x]_i$. Then the assumptions on the norm of A correspond to ensuring f_i is 1-Lipschitz with respect to the ℓ_2-norm in Definition 18 and ℓ_1-norm in Definition 19 (which in turn implies f is 1-Lipschitz in the respective norms). This normalization is performed to simplify expressions as in [8]. (In particular, [8] also considers the more general problem where each f_i can be any smooth, Lipschitz, convex function.)

Recently, [8, Cor. 8.2] achieved a state-of-the-art runtime in certain parameter regimes of $\widetilde{O}(nd + n(d/\epsilon)^{2/3} + d\epsilon^{-2})$ for obtaining an ϵ-optimal point for (P-MG). However, unlike previous algorithms for (P-MG) (see Section 1.3 for an extended discussion), their algorithm does not yield an ϵ-optimal point for (D-MG) with the same runtime.

Our instantiation of the dual-extraction framework in Algorithm 3 and the accompanying guarantee Theorem 21 resolves this asymmetry between the complexity of obtaining a primal versus dual ϵ-optimal point by obtaining an ϵ-optimal point of (D-MG) with the same runtime of $\widetilde{O}(nd + n(d/\epsilon)^{2/3} + d\epsilon^{-2})$. At the end of Section 4.1, we observe that Theorem 21 also yields a new state-of-the-art runtime for the primal (P-MG) in the setting of Definition 19 due to the symmetry of the constraint sets and ψ.

Algorithm 3 Dual extraction for matrix games.

Input: $(\psi, \mathcal{X}, \Delta^n)$-simplex setup (Definition 11), iteration count $K \in \mathbb{N}$, dual-regularization schedule $(\lambda_k \in \mathbb{R}_{>0})_{k=0}^{K-1}$, primal-accuracy schedule $(\epsilon_k \in \mathbb{R}_{>0})_{k=1}^{K}$, DRPOSP oracle (Definition 14)

$y_0 \coloneqq \frac{1}{n}\mathbf{1}$

for $k = 1, 2, \ldots, K$ **do**

  $\Lambda_k = \sum_{j=0}^{k-1} \lambda_j$

  $[q_k]_i \propto \prod_{j=0}^{k-1} [y_j]_i^{\lambda_j/\Lambda_k},\ \forall i \in [n]$ // Note: $q_k \in \Delta^n$

  $x_k = \mathrm{DRPOSP}(q_k, \Lambda_k, \epsilon_k, \frac{1}{10K})$

  $[y_k]_i \propto [q_k]_i \cdot \exp(\Lambda_k^{-1} \cdot [A^\top x_k]_i),\ \forall i \in [n]$ // $y_k = \operatorname{argmax}_{y \in \Delta^n} \left\{ x_k^\top A y - \Lambda_k V_{q_k}(y) \right\}$

return y_K

Before giving the guarantee Theorem 21 for Algorithm 3, the following lemma provides a runtime bound for the DRPOSP oracle when the success probability is 9/10 (see Appendix B.1 in the full version for the proof). In particular, Lemma 20 shows that adding dual regularization to (P-MG) does not increase the complexity of obtaining an ϵ-optimal point over the guarantee of [8, Cor. 8.2] discussed above.

▶ **Lemma 20** (DRPOSP oracle for matrix games). *In the settings of Definitions 18 and 19, for any $q \in \Delta^n_{>0}$, $\epsilon_{\mathrm{p}} > 0$, and $\lambda > 0$, with success probability at least 9/10, there exists an algorithm which returns an expected ϵ_{p}-optimal point of $f_{\lambda,q}$ with runtime $\widetilde{O}(nd + n(d/\epsilon_{\mathrm{p}})^{2/3} + d\epsilon_{\mathrm{p}}^{-2})$. (Equivalently, per Definition 14, we have that DRPOSP$(q, \lambda, \epsilon_{\mathrm{p}}, 1/10)$ can be implemented with this runtime.)*

Now for the main guarantee (we defer the proof to the full version):

▶ **Theorem 21** (Guarantee for Algorithm 3). *In the settings of Definitions 18 and 19, given target error $\epsilon > 0$ and with success probability at least 9/10, Algorithm 3 with dual-regularization and primal-accuracy schedules given by*

$$\lambda_k = 2^k \frac{\epsilon}{4 \ln n} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\epsilon}{4K} \text{ for } k \in [K]$$

for $K = \left\lceil \max\left\{ \log_2 \frac{\ln n}{\epsilon^2}, 1 \right\} \right\rceil + 10$ returns an expected ϵ-optimal point for (D-MG)*, and can be implemented with runtime $\widetilde{O}(nd + n(d/\epsilon)^{2/3} + d\epsilon^{-2})$.*

The primal perspective

As alluded to above, the guarantee of Theorem 21 also implies a new state-of-the-art runtime for the primal (P-MG) in the setting of Definition 19. This follows because in the matrix games simplex setup, (P-MG) and (D-MG) are symmetric in terms of their constraint sets, so we can obtain an expected ϵ-optimal point for (P-MG) via Theorem 21 by negating and treating (P-MG) as if it were the dual problem. Formally (we defer the proof to the full version):

▶ **Corollary 22** (Guarantee for (P-MG) in the matrix games simplex setup). *In the setting of Definition 19, there exists an algorithm which, given target error $\epsilon > 0$ and with success probability at least 9/10, returns an expected ϵ-optimal point for* (P-MG) *with runtime $\widetilde{O}(nd + d(n/\epsilon)^{2/3} + n\epsilon^{-2})$.*

See the full version for a discussion of how this runtime compares to the prior art.

4.2 CVaR at level α DRO

In this section, we instantiate $\psi(x, y) \coloneqq \sum_{i=1}^n y_i f_i(x)$ for convex, bounded, and G-Lipschitz (with respect to the Euclidean norm) functions $f_i : \mathbb{R}^d \to \mathbb{R}$.[4] Given a compact, convex set $\mathcal{X}$ and $\alpha \in [1/n, 1]$, the primal and dual problem for CVaR at level α are as follows (we explain the reason for the notation $\bar{f}$ as opposed to f in the full version; in short, we apply the framework to a proxy objective):

$$\underset{x\in\mathcal{X}}{\text{minimize}}\ \bar{f}(x) \qquad \text{for } \bar{f}(x) \coloneqq \max_{y\in\Delta^n, \|y\|_\infty \le \frac{1}{\alpha n}} \sum_{i=1}^n y_i f_i(x), \text{ and} \tag{P-CVaR}$$

$$\underset{y\in\Delta^n, \|y\|_\infty \le \frac{1}{\alpha n}}{\text{maximize}}\ \phi(y) \text{ for } \phi(y) \coloneqq \min_{x\in\mathcal{X}} \sum_{i=1}^n y_i f_i(x). \tag{D-CVaR}$$

Our complexity model in this section is the number of computations of the form $(f_i(x), \nabla f_i(x))$ for $x \in \mathcal{X}$ and $i \in [n]$. We refer to the evaluation of $(f_i(x), \nabla f_i(x))$ for a given $x \in \mathcal{X}$ and $i \in [n]$ as a single *first-order query*. Omitting the Lipschitz constant G and bounds on the range of the f_i's and size of $\mathcal{X}$ for clarity, [29, Sec. 4] gave an algorithm which returns an expected[5] ϵ-optimal point of (P-CVaR) with $\widetilde{O}(\alpha^{-1}\epsilon^{-2})$ first-order queries, and also proved a matching lower bound up to logarithmic factors when n is sufficiently large. However, to the best of our knowledge, the best known complexity for obtaining an expected ϵ-optimal point of (D-CVaR) is $\widetilde{O}(n\epsilon^{-2})$ via a primal-dual method based on [33]; see also [13, 32, 5]. In our main guarantee for this section, Theorem 24, we apply Algorithm 2 to obtain an expected ϵ-optimal point of (D-CVaR) with complexity $\widetilde{O}(\alpha^{-1}\epsilon^{-2} + n)$, which always improves upon or matches $\widetilde{O}(n\epsilon^{-2})$ since $\alpha \in [1/n, 1]$.

Toward stating our main guarantee, we encapsulate the formal assumptions of [29, Sec. 2] in the following definition:

▶ **Definition 23** (CVaR at level α setup)**.** *We assume $\mathcal{X}$ is nonempty, closed, convex, and satisfies $\|x - y\|_2 \le R$ for all $x, y \in \mathcal{X}$. We also assume, for all $i \in [n]$, that f_i is convex, G-Lipschitz with respect to $\|\cdot\|_2$, and satisfies $f_i(x) \in [0, M]$ for all $x \in \mathcal{X}$.*

We ultimately obtain the following guarantee via Algorithm 2. Note that the upper bound on ϵ in Theorem 24 is without loss of generality since if $\epsilon \ge M$, any feasible point is ϵ-optimal. We defer the proof to the full version.

▶ **Theorem 24** (Guarantee for (D-CVaR))**.** *In the setting of Definition 23 with target error $\epsilon \in (0, 4M)$ and $\alpha \in [1/n, 1]$, there exists an algorithm which computes an expected ϵ-optimal point of* (D-CVaR) *with complexity $\widetilde{O}(n + G^2R^2\alpha^{-1}\epsilon^{-2})$.*

5 Obtaining critical points of convex functions

In this section, our goal is to obtain an approximate critical point of a convex, β-smooth function $h : \mathbb{R}^n \to \mathbb{R}$, given access to a gradient oracle for h. We show that our general framework yields an algorithm with the optimal query complexity for this problem. In

[4] Note that we do not require the functions f_i to be differentiable. Here, it is important that Definition 9 only requires $\psi(x, \cdot)$ to be differentiable.

[5] To be precise, [29] gives a $\widetilde{O}(\alpha^{-1}\epsilon^{-2})$-complexity high probability bound in Theorem 2. They do not state a $\widetilde{O}(\alpha^{-1}\epsilon^{-2})$-complexity expected suboptimality bound explicitly in a theorem, but they note in the text above Theorem 2 that such a bound follows immediately from Propositions 3 and 4 in their paper.

Section 5.1, we give the formal problem definition and some important preliminaries. In Section 5.2, we give the setup for applying our main framework Algorithm 2 to this problem and a sketch of why the resulting algorithm works. In Section 5.3, we formally state the resulting algorithm for obtaining an approximate critical point of h and prove that it achieves the optimal rate using the guarantees associated with Algorithm 2.

5.1 Preliminaries for Section 5

Throughout Section 5, we fix $\|\cdot\|$ to be the standard Euclidean norm over $\mathbb{R}^n$. We assume $h : \mathbb{R}^n \to \mathbb{R}$ is convex, β-smooth with respect to $\|\cdot\|$, and $\Delta \coloneqq h(x_0) - \inf_{x\in\mathbb{R}^n} h(x) < \infty$ for an arbitrary initialization point $x_0 \in \mathbb{R}^n$. We access h through a gradient oracle. For $\gamma > 0$, our goal will be to obtain a γ-critical point of h, i.e., a point $x \in \mathbb{R}^n$ such that $\|\nabla h(x)\| \le \gamma$. Instead of operating on h itself, our algorithm will operate on a regularized version of h:

$$f(x) \coloneqq h(x) + \frac{\gamma^2}{16\Delta}\|x - x_0\|^2. \tag{11}$$

This notation was chosen to mirror the notation of Section 3.1; f will be the primal function when we apply the framework. Let $x_f^\star$ denote the unique global minimum of f. The following corollary of Lemma 13 in Appendix C in the full version summarizes the key properties of f:

▶ **Corollary 25** (Properties of the regularized function f)**.** *We have*
1. $\|x_f^\star - x_0\| \le 4\Delta/\gamma$.
2. *If* $u \in \mathbb{R}^n$ *is such that* $\|\nabla f(u)\| \le \gamma/4$*, then* $\|\nabla h(u)\| \le \gamma$.

Proof. This follows immediately from Lemma 13 in the full version with $\alpha \leftarrow \frac{\gamma^2}{8\Delta}$ and $\nu \leftarrow \gamma/4$. ◀

The second part of Corollary 25 says that to find a γ-critical point of h, it suffices to find a $(\gamma/4)$-critical point of f. Furthermore, clearly a single query to ∇h suffices to obtain ∇f at a point. As a result, we will focus on finding a $(\gamma/4)$-critical point of f. Furthermore, Corollary 25 may be of independent interest since it trivially allows one to achieve a gradient query complexity of $O\left(\sqrt{\beta\Delta}/\gamma\right)$ via a method which achieves query complexity $O\left(\sqrt{\beta\|x_0 - x_h^\star\|/\gamma}\right)$ (for $x_h^\star$ defined as some minimizer of h over $\mathbb{R}^n$, assuming one exists); see Section 1.3.

The reason we perform this regularization before applying our framework is it enables us to obtain a sufficiently tight bound on $V_{y_0}(y^\star)$ (equivalently, a small enough value of B when we ultimately apply Corollary 17). It is possible to apply the framework more directly to h, but it is not clear how to do so in a way that achieves an optimal complexity.

Finally, we provide a notation guide for Section 5 in Table 1, which may be useful to reference as additional notation is introduced in Sections 5.2 and 5.3.

5.2 Instantiating the framework

For this application, we instantiate

$$\psi(x, y) \coloneqq \langle x, y\rangle - f^*(y).$$

Recall that ψ is the *Fenchel game* [1, 43, 12, 23]; see Section 1.1 for a discussion of why it is a natural choice in this setting. For the rest of Section 5, we fix a $(\psi, \mathcal{X} \coloneqq B_R^n(x_0), \mathcal{Y} \coloneqq \mathbb{R}^n, f^*)$-unbounded setup (Definition 10) with $R \coloneqq 5\Delta/\gamma$. f^* is a valid choice for the dgf because

Table 1 Notation guide for Section 5.

Notation	Description	Section
$\|\cdot\|$	Euclidean norm	5.1
h	Convex, β-smooth function	
γ	Target critical point error for h	
x_0	Arbitrary initialization point	
Δ	$h(x_0) - \inf_{x\in\mathbb{R}^n} h(x) < \infty$	
$f(x)$	$h(x) + \frac{\gamma^2}{16\Delta}\|x - x_0\|^2$	
$x_f^\star$	The global minimizer of f	
$\psi(x, y)$	$\langle x, y\rangle - f^*(y)$	5.2
R	$5\Delta/\gamma$	
$\mathcal{X}$	$B_R^n(x_0)$	
$\mathcal{Y}$	$\mathbb{R}^n$	
dgf r	f^*	
$\phi(y)$	$\langle x_0, y\rangle - R\|y\| - f^*(y)$	
λ_k	$2^k/32$	5.3
ϵ_k	$\Delta/(64 \cdot 1.5^k)$	
CGM	Fast composite gradient method oracle	

f^* is differentiable and $\left(\beta + \frac{\gamma^2}{8\Delta}\right)^{-1}$-strongly convex [38, Thm. 6.11]. The strong convexity of f^* also implies that Assumption 6 holds. Note that the associated primal function $x \mapsto \max_{y\in\mathbb{R}^n} \psi(x, y)$ is precisely $f^{**} = f$ (hence the choice of notation in (11)), and the dual function is given by

$$\phi(y) = \min_{x\in B_R^n(x_0)} \{\langle x, y\rangle - f^*(y)\} = \left\langle x_0 - R\frac{y}{\|y\|}, y\right\rangle - f^*(y) = \langle x_0, y\rangle - R\|y\| - f^*(y).$$

Next, the following lemma fulfills part of the outline given in Section 1.1 by showing that approximately optimal points for the dual objective (D) must have small norm. We defer the proof to the full version.

▶ **Lemma 26** (Bounding the norm by dual suboptimality). *If $y \in \mathbb{R}^n$ is ϵ-optimal for* (D) *for some $\epsilon > 0$, then $\|y\| \le \epsilon\gamma/\Delta$.*

We now derive the oracles of Definitions 12 and 13. Regarding Definition 12, for the rest of Section 5 we restrict DRPO$(\cdot)$ to denote a deterministic implementation of the DRPO oracle, since we can always obtain a deterministic implementation in this application. Then the following corollary is an immediate consequence of a more general lemma given in Appendix C in the full version which characterizes the properties of the Fenchel game with added dual regularization; see also Section 1.1.

▶ **Corollary 27.** *The set of valid output points of $DRPO(q \in \mathbb{R}^n, \lambda > 0, \epsilon_{\mathrm{p}} > 0)$ is precisely*

$$\operatorname*{argmin}^{\epsilon_{\mathrm{p}}}_{x\in B_R^n(x_0)} (1+\lambda)\cdot f\left(\frac{x + \lambda\nabla f^*(q)}{1+\lambda}\right), \text{ and}$$

$$DRBR(q \in \mathbb{R}^n, \lambda > 0, x \in B_R^n(x_0)) = \nabla f\left(\frac{x + \lambda\nabla f^*(q)}{1+\lambda}\right).$$

Proof. Apply Lemma 14 in the full version with $g \leftarrow f$. ◀

Taken together, Lemma 26 and Corollary 27 nearly immediately imply that Algorithm 2 can be applied to the above setup to obtain a $(\gamma/4)$-critical point of f (and therefore a γ-critical point of h). In particular, we will apply the schedules of Corollary 17 to certify that the output y_K of Algorithm 2 is close in distance to an ϵ-optimal point for (D) for an appropriate choice of $\epsilon > 0$. Then Lemma 26 and a triangle inequality yield a bound on $\|y_K\|$. Finally, since

$$y_K \coloneqq \mathrm{DRBR}(q_K, \Lambda_K, x_K) = \nabla f\left(\frac{x_K + \Lambda_K \nabla f^*(q_K)}{1 + \Lambda_K}\right)$$

by Corollary 27, we have that $\frac{x_K+\Lambda_K\nabla f^*(q_K)}{1+\Lambda_K}$ is an approximate critical point of f (and therefore h). One may worry about the presence of $\nabla f^*(q_K)$ here and, more generally, the presence of $\nabla f^*(q)$ in the expressions for the oracles in Corollary 27. However, $\nabla f^*(\cdot)$ never needs to be evaluated explicitly since per the alternate expression for q_k given in Line 3 of Algorithm 2, note that q_k was itself computed as the gradient of f at a point (recall the dgf is f^* and $f = f^{**}$), in which case ∇f^* simply undoes this operation by Lemma 16 in the full version.

We formalize this sketch and provide a complexity guarantee in the next section. We also reframe this sketch and treat the sequence of $\frac{x_k+\Lambda_k\nabla f^*(q_k)}{1+\Lambda_k}$ terms as our iterates (as opposed to the sequence of x_k's), as this leads to a simpler statement and interpretation of the resulting algorithm.

5.3 The resulting algorithm and guarantee

We now formalize the sketch given at the end of the previous section, state the resulting algorithm, and provide a complexity guarantee. But first, we define a subroutine which will be used by the algorithm to implement the DRPO oracle:

▶ **Definition 28** (CGM oracle [40, 36]). *A $(\zeta > 0, w \in \mathbb{R}^n, \epsilon > 0)$-fast composite gradient method oracle, $CGM(\zeta, w, \epsilon)$, returns an ϵ-minimizer of f over $x \in B_\zeta^n(w)$, i.e., an element of $\operatorname{argmax}^{\epsilon}_{x \in B_\zeta^n(w)} f(x)$, using at most $O\left(1 + \sqrt{\frac{\beta\zeta^2}{\epsilon}}\right)$ queries to ∇f.*

For example, implementations with a small constant can be found in [40] or [36, Sec. 6.1.3]. The implementation of the CGM oracle falls under fast gradient methods for composite minimization, where letting g denote a convex, β-smooth function and Ψ a "simple regularizer" (a quadratic in our case), the goal is to minimize $\tilde{g}(x) \coloneqq g(x) + \Psi(x)$ with the same complexity as it takes to minimize g. The domain constraint can also be baked into the regularizer Ψ by adding an indicator.

Our method for computing a γ-critical point of h is given in Algorithm 4, with the associated guarantee in Theorem 30. We note that the decision to introduce the equivalent notation z_0 for x_0 in Line 1 is aesthetic (to make Line 5 simpler to state and interpret). Furthermore, we state Algorithm 4 for general schedules $(\lambda_k)_{k=0}^{K-1}$ and $(\epsilon_k)_{k=1}^{K}$ for clarity, but ultimately we will choose the schedules given in Theorem 30, which correspond to particularizing the schedules of Corollary 17 to this setting. With this choice of schedules, $\Lambda_k \approx 2^k$ and $\epsilon_k \approx \Delta/1.5^k$ so that $\frac{\epsilon_k}{1+\Lambda_k} \approx \Delta/3^k$. As a result, Algorithm 4 can be interpreted as optimizing f in a sequence of balls where the radius and target error are both decreasing geometrically, and the center is a convex combination of the past iterates. While we choose the iteration count K to be logarithmic in the problem parameters, we avoid multiplicative

logarithmic factors in the total complexity because the ratio ζ^2/ϵ in the complexity of the CGM oracle call (to borrow the notation of Definition 28) in Line 5 of Algorithm 4 is $\approx \frac{R^2}{4^k} \cdot \frac{3^k}{\Delta}$ at the k-th iteration, meaning it is collapsing geometrically.

Algorithm 4 Algorithm for obtaining a γ-critical point of h.

Input: Sequences $(\lambda_k)_{k=0}^{K-1}$ and $(\epsilon_k)_{k=1}^{K}$ specified in Theorem 30, iteration count $K \in \mathbb{N}$, CGM oracle (Definition 28)

1 $z_0 := x_0$
2 **for** $k = 1, 2, \ldots, K$ **do**
3 $\quad \Lambda_k = \sum_{j=0}^{k-1} \lambda_j$
4 $\quad w_k = \frac{z_0 + \sum_{j=0}^{k-1} \lambda_j z_j}{1+\Lambda_k}$
5 $\quad z_k = \mathrm{CGM}\left(\frac{R}{1+\Lambda_k}, w_k, \frac{\epsilon_k}{1+\Lambda_k}\right)$ // $z_k \in \operatorname{argmin}^{\epsilon_k/(1+\Lambda_k)}_{z \in B^n_{R/(1+\Lambda_k)}(w_k)} f(z)$
6 **return** z_K

Toward analyzing Algorithm 4, we first connect the sequence of iterates z_k produced by Algorithm 4 to the sequence of iterates x_k, y_k, q_k produced by Algorithm 2 with the same input parameters. Namely, we are formalizing the comment made at the end of Section 5.2 about reframing the sequence of iterates to achieve a more interpretable algorithm. We defer the proof to the full version.

▶ **Lemma 29** (Connecting Algorithm 4 to Algorithm 2). *Consider Algorithm 2 with input given by a $(\psi, B_R^n(x_0), \mathbb{R}^n, f^*)$-unbounded setup (Definition 10); $y_0 := \nabla f(x_0)$; and K, $(\epsilon_k)_{k=1}^K$, and $(\lambda_k)_{k=0}^{K-1}$ as in Algorithm 4. Then letting $(z_k)_{k=0}^K$ denote the sequence of iterates generated by Algorithm 4, the following are valid sequences of iterates for Algorithm 2:*

$$q_k = \nabla f\left(\frac{1}{\Lambda_k}\sum_{j=0}^{k-1} \lambda_j z_j\right) \quad \text{for } k \in [K], \tag{12}$$

$$x_k = (1+\Lambda_k) z_k - \sum_{j=0}^{k-1} \lambda_j z_j \quad \text{for } k \in [K], \text{ and} \tag{13}$$

$$y_k = \nabla f(z_k) \quad \text{for } k \in \{0\} \cup [K]. \tag{14}$$

Having connected Algorithm 4 to Algorithm 2, we can apply the schedules given in Corollary 17 to show that Algorithm 4 returns a γ-critical point of h with an optimal complexity. We defer the proof to the full version.

▶ **Theorem 30** (Guarantee for Algorithm 4). *For any[6] $\gamma \in (0, \sqrt{2\beta\Delta})$ and with $K = O(\log(\beta\Delta/\gamma))$, the output of Algorithm 4 with schedules given by*

$$\lambda_k = \frac{2^k}{32} \text{ for } k \in \{0\} \cup [K-1] \text{ and } \epsilon_k = \frac{\Delta}{64 \cdot 1.5^k} \text{ for } k \in [K] \tag{15}$$

satisfies $\|\nabla h(z_K)\| \le \gamma$, and the algorithm makes at most $O\left(\frac{\sqrt{\beta\Delta}}{\gamma}\right)$ gradient queries to h.

[6] The restriction on γ is without loss of generality since $\|\nabla h(x_0)\| \le \sqrt{2\beta\Delta}$ by smoothness. We add it because it simplifies the analysis.

References

1 Jacob D Abernethy and Jun-Kun Wang. On frank-wolfe and equilibrium computation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

2 Ilan Adler. The equivalence of linear programs and zero-sum games. *International Journal of Games Theory*, Volume 42:165-177, February 2013.

3 Zeyuan Allen-Zhu and Elad Hazan. Optimal black-box reductions between optimization objectives. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 1614–1622, Red Hook, NY, USA, 2016. Curran Associates Inc.

4 Hilal Asi, Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Stochastic bias-reduced gradient methods. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2024. Curran Associates Inc.

5 Yair Carmon and Danielle Hausler. Distributionally robust optimization via ball oracle acceleration. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

6 Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Thinking inside the ball: Near-optimal minimization of the maximal loss. In Mikhail Belkin and Samory Kpotufe, editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 866–882. PMLR, 15–19 August 2021. URL: `http://proceedings.mlr.press/v134/carmon21a.html`.

7 Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. Recapp: Crafting a more efficient catalyst for convex optimization. In *International Conference on Machine Learning*, 2022.

8 Yair Carmon, Arun Jambulapati, Yujia Jin, and Aaron Sidford. *A Whole New Ball Game: A Primal Accelerated Method for Matrix Games and Minimizing the Maximum of Smooth Functions*, pages 3685–3723. Society for Industrial and Applied Mathematics, 2024. `doi:10.1137/1.9781611977912.130`.

9 Yair Carmon, Yujia Jin, Aaron Sidford, and Kevin Tian. Variance reduction for matrix games. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

10 Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff. Sublinear optimization for machine learning. *J. ACM*, 59(5), November 2012. `doi:10.1145/2371656.2371658`.

11 Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 938–942, New York, NY, USA, 2019. Association for Computing Machinery. `doi:10.1145/3313276.3316303`.

12 Michael B. Cohen, Aaron Sidford, and Kevin Tian. Relative lipschitzness in extragradient methods and a direct recipe for acceleration. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 62:1–62:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPICS.ITCS.2021.62`.

13 Sebastian Curi, Kfir Y. Levy, Stefanie Jegelka, and Andreas Krause. Adaptive sampling for stochastic risk-averse learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

14 G. B. Dantzig. Linear programming and extensions, 1953.

15 Jelena Diakonikolas and Puqian Wang. Potential function-based framework for minimizing gradients in convex and min-max optimization. *SIAM Journal on Optimization*, 32(3):1668–1697, 2022. `doi:10.1137/21M1395302`.

16 John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49, June 2021.

17 I. Ekeland, Roger Temam, Society for Industrial, and Applied Mathematics. *Convex analysis and variational problems*. Classics in applied mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa., 1999.

18 Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2540–2548. JMLR.org, 2015. URL: `http://proceedings.mlr.press/v37/frostig15.html`.

19 Michael B. Giles. Multilevel monte carlo path simulation. *Operations Research*, 56(3):607–617, June 2008. `doi:10.1287/OPRE.1070.0496`.

20 Michael B. Giles. Multilevel monte carlo methods. *Acta Numerica*, 24:259–328, May 2015. `doi:10.1017/S096249291500001X`.

21 G. N. Grapiglia and Yurii Nesterov. Tensor methods for finding approximate stationary points of convex functions. *Optimization Methods and Software*, 37(2):605–638, 2022. `doi:10.1080/10556788.2020.1818082`.

22 Michael D. Grigoriadis and Leonid G. Khachiyan. A sublinear-time randomized approximation algorithm for matrix games. *Operations Research Letters*, 18(2):53–58, 1995. `doi:10.1016/0167-6377(95)00032-0`.

23 Yujia Jin, Aaron Sidford, and Kevin Tian. Sharper rates for separable minimax and finite sum optimization via primal-dual extragradient methods. In Po-Ling Loh and Maxim Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 4362–4415. PMLR, 02–05 July 2022. URL: `https://proceedings.mlr.press/v178/jin22b.html`.

24 Donghwan Kim and Jeffrey A. Fessler. Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions. *Journal of Optimization Theory and Applications*, 188(1):192–219, January 2021. `doi:10.1007/S10957-020-01770-2`.

25 Jaeyeon Kim, Asuman Ozdaglar, Chanwoo Park, and Ernest K. Ryu. Time-reversed dissipation induces duality between minimizing gradient norm and function value, 2023. `arXiv:2305.06628`.

26 Jaeyeon Kim, Chanwoo Park, Asuman Ozdaglar, Jelena Diakonikolas, and Ernest K. Ryu. Mirror duality in convex optimization, 2024. `arXiv:2311.17296`.

27 Guanghui Lan, Yuyuan Ouyang, and Zhe Zhang. Optimal and parameter-free gradient minimization methods for convex and nonconvex optimization, 2023. `arXiv:2310.12139`.

28 Jongmin Lee, Chanwoo Park, and Ernest K. Ryu. A geometric structure of acceleration and its role in making gradients small fast. In *Neural Information Processing Systems*, 2021.

29 Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8847–8860. Curran Associates, Inc., 2020.

30 Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3384–3392, Cambridge, MA, USA, 2015. MIT Press.

31 M. Minsky and S. Papert. Perceptrons: An introduction to computational geometry, 1988.

32 Hongseok Namkoong and John C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2216–2224, Red Hook, NY, USA, 2016. Curran Associates Inc.

33 A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009. `doi:10.1137/070704277`.

34 A.S Nemirovsky. On optimality of krylov's information when solving linear operator equations. *Journal of Complexity*, 7(2):121–130, 1991. `doi:10.1016/0885-064X(91)90001-E`.

35 A.S Nemirovsky. Information-based complexity of linear operator equations. *Journal of Complexity*, 8(2):153–175, 1992. doi:10.1016/0885-064X(92)90013-2.

36 Yurii Nesterov. *Lectures on Convex Optimization*. Springer Publishing Company, Incorporated, 2nd edition, 2018.

37 Yurii Nesterov, Alexander Gasnikov, Sergey Guminov, and Pavel Dvurechensky. Primal-dual accelerated gradient methods with small-dimensional relaxation oracle. *Optimization Methods and Software*, 36(4):773–810, July 2021. doi:10.1080/10556788.2020.1731747.

38 Francesco Orabona. A modern introduction to online learning, 2023. arXiv 1912.13213. arXiv:1912.13213.

39 Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8:171–176, 1958.

40 Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization, 2008.

41 J. v. Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, December 1928.

42 Jan van den Brand, Yin Tat Lee, Yang P. Liu, Thatchaphol Saranurak, Aaron Sidford, Zhao Song, and Di Wang. Minimum cost flows, mdps, and ℓ_1-regression in nearly linear time for dense instances. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 859–869, New York, NY, USA, 2021. Association for Computing Machinery.

43 Jun-Kun Wang, Jacob Abernethy, and Kfir Y. Levy. No-regret dynamics in the fenchel game: a unified framework for algorithmic convex optimization. *Mathematical Programming*, 205(1-2):203–268, May 2024. doi:10.1007/S10107-023-01976-Y.