Information Complexity of Mixed-integer Convex Optimization

Amitabh Basu[1], Hongyi Jiang[2], Phillip Kerger[1], and Marco Molinaro[3]

[1]Department of Applied Mathematics and Statistics, Johns Hopkins University, {abasu9,pkerger}@jhu.edu
[2]School of Civil and Environmental Engineering, Cornell University, hj348@cornell.edu
[3]Microsoft Research (Redmond) and Computer Science Department, PUC-Rio, mmolinaro@microsoft.com

Abstract

We investigate the information complexity of mixed-integer convex optimization under different types of oracles. We establish new lower bounds for the standard first-order oracle, improving upon the previous best known lower bound. This leaves only a lower order linear term (in the dimension) as the gap between the lower and upper bounds. This is derived as a corollary of a more fundamental "transfer" result that shows how lower bounds on information complexity of continuous convex optimization under different oracles can be transferred to the mixed-integer setting in a black-box manner.

Further, we (to the best of our knowledge) initiate the study of, and obtain the first set of results on, information complexity under oracles that only reveal *partial* first-order information, e.g., where one can only make a binary query over the function value or subgradient at a given point. We give algorithms for (mixed-integer) convex optimization that work under these less informative oracles. We also give lower bounds showing that, for some of these oracles, every algorithm requires more iterations to achieve a target error compared to when complete first-order information is available. That is, these oracles are provably less informative than full first-order oracles for the purpose of optimization.

Keywords: Mixed-integer optimization, convex optimization, information complexity, lower bounds

1 First-order information complexity

We consider the problem class of *mixed-integer convex optimization*:

$$\inf\{f(\mathbf{x},\mathbf{y}) : (\mathbf{x},\mathbf{y}) \in C, (\mathbf{x},\mathbf{y}) \in \mathbb{Z}^n \times \mathbb{R}^d\}, \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$ is a convex (possibly nondifferentiable) function and $C \subseteq \mathbb{R}^n \times \mathbb{R}^d$ is a closed, convex set. Given $\varepsilon \geq 0$, we wish to report a point $(\mathbf{x},\mathbf{y}) \in C \cap (\mathbb{Z}^n \times \mathbb{R}^d)$ such that $f(\mathbf{x},\mathbf{y}) \leq f(\mathbf{x}',\mathbf{y}') + \varepsilon$ for all $(\mathbf{x}',\mathbf{y}') \in C \cap (\mathbb{Z}^n \times \mathbb{R}^d)$. Such a point will be called an *ε-approximate solution* and points in $C \cap (\mathbb{Z}^n \times \mathbb{R}^d)$ will be called *feasible solutions*. We say that $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are the *integer-valued decision variables* or simply the *integer variables* of the problem, and $\mathbf{y}_1, \ldots, \mathbf{y}_d$ are called the *continuous variables*.

The notion of *information complexity* (a.k.a. *oracle complexity* or *analytical complexity*) goes back to foundational work by Nemirovski and Yudin [12] on convex optimization (without integer variables) and is based on the following. An algorithm for reporting an ε-approximate solution to an instance (f, C) must be "given" the instance somehow. Allowing only instances with explicit, algebraic descriptions (e.g., the case of linear programming) can be restrictive in some settings. To work with more general, nonlinear instances, the algorithm is allowed to make queries to an oracle to collect information about the instance. More formally, we have the following definition.

Definition 1. *An oracle $\mathcal{O}$ for an optimization problem class $\mathcal{I}$ is given by a family $\mathcal{Q}$ of possible queries along with a set H of possible answers or responses. A query $q \in \mathcal{Q}$ is a function $q : \mathcal{I} \to H$. We say that $q(I) \in H$ is the* answer *or* response *to the query q for the instance $I \in \mathcal{I}$.*

Any algorithm using such an oracle to find an ε-approximate solution for an instance makes queries about the instance in a sequence according to some strategy depending on the queries made and answers received, which we define formally as its *query strategy*.

Definition 2. *A* query strategy *is a function $D : (\mathcal{Q} \times H)^* \to \mathcal{Q}$, where $(\mathcal{Q} \times H)^*$ denotes the set of all finite sequences over $\mathcal{Q} \times H$, including the empty sequence. The* transcript $\Pi(D, I)$ of a strategy D on an instance $I = (f, C)$ *is the sequence of query and response pairs $(q_i, q_i(I))$, $i = 1, 2, \ldots$ obtained when one applies D on I, i.e., $q_1 = D(\emptyset)$ and $q_i = D((q_1, q_1(I)), \ldots, (q_{i-1}, q_{i-1}(I)))$ for $i \geq 2$.*

If different instances with no common ε-approximate solution produce the same transcript for the queries an algorithm has made, then the algorithm cannot tell them apart and will be unable to reliably report an ε-solution for those instances after those queries. The goal is to design a query strategy that can report an ε-approximate solution after making the smallest number of queries. This motivates the following definition of information complexity:

Definition 3. *Given a family of instances $\mathcal{I}$ and access to an oracle $\mathcal{O}$, the* ε-information complexity $\text{icomp}_\varepsilon(D, I, \mathcal{O})$ of an instance I for a query strategy D, *is defined as the minimum natural number k such that the set of all instances in $\mathcal{I}$ for which $\mathcal{O}$ returns the same responses as the instance I to the first k queries of D have a common ε-approximate solution. The* ε-information complexity of the problem class $\mathcal{I}$ with respect to the oracle $\mathcal{O}$, *is defined as*

$$\text{icomp}_\varepsilon(\mathcal{I}, \mathcal{O}) := \inf_D \sup_{I \in \mathcal{I}} \text{icomp}_\varepsilon(D, I, \mathcal{O})$$

where the infimum is taken over all query strategies.

Thus, to prove an upper bound u on $\text{icomp}_\varepsilon(\mathcal{I}, \mathcal{O})$, it suffices to construct a query strategy that requires, in the worst case, at most u queries to narrow down to a collection of instances that all have a common ε-approximate solution. On the other hand, to establish a lower bound of ℓ on $\text{icomp}_\varepsilon(\mathcal{I}, \mathcal{O})$, one needs to show that for any query strategy D, there exists a collection of instances in $\mathcal{I}$ that give the same responses to the first ℓ queries of D (on these instances), and there is no point in $\mathbb{R}^n \times \mathbb{R}^d$ that is a common ε-approximate solution to all these instances.

While we introduce information complexity allowing for any general choice of oracle, the standard oracle that has been studied over the past several decades for convex optimization is the so-called *(full-information) first-order oracle*, which has two types of queries indexed by points in $\mathbb{R}^n \times \mathbb{R}^d$: i) a *separation oracle* query indexed by a point $\mathbf{z} \in \mathbb{R}^{n+d}$ reports "YES" if $\mathbf{z} \in C$ and otherwise reports a separating hyperplane for $\mathbf{z}$ and C, ii) a *subgradient oracle* query indexed by a point $\mathbf{z} \in \mathbb{R}^{n+d}$ reports $f(\mathbf{z})$ and a subgradient for f at $\mathbf{z}$. Tight lower and upper bounds (differing by only a small constant factor) on the number of queries required were obtained by Nemirovski and Yudin in their seminal work [12] for the case with no integer variables; roughly speaking, the bound is $\Theta\left(d\log\left(\frac{1}{\varepsilon}\right)\right)$. These insights were extended to the mixed-integer setting in [13, 3, 2], with the best known lower and upper bounds stated in [2].

Observe that the response to any separation/subgradient query is a vector in $\mathbb{R}^{n+d}$. Thus, each query reveals at least $n+d$ bits of information about the instance. A more careful accounting that measures the "amount of information" accrued would track the total number of bits of information obtained as opposed to just the total number of oracle queries made. A natural question, posed in [2], is whether the bounds from the classical analysis would change if one uses this new measure of the total number of bits, as opposed to the number of queries. The intuition, roughly, is that one should need a factor $(n+d)\log\left(\frac{1}{\varepsilon}\right)$ larger than the number of first-order queries, because one should need to probe at least $\log\left(\frac{1}{\varepsilon}\right)$ bits in $n+d$ coordinates to recover the full subgradient/separating hyperplane (up to desired approximations). We attempt to make some progress on this question in this paper.

The above discussion suggests that one should consider oracles that return a desired bit of a desired coordinate of the separating hyperplane vector or subgradient. However, one can imagine making other binary queries on the instance; for example, one can pick a direction and ask for the sign of the inner product of the subgradient and this direction. In fact, one can consider more general binary queries that have nothing to do with subgradients/separating hyperplanes. If one allows *all* possible binary queries, i.e., one can use any function from the space of instances to $\{0,1\}$ as a query, then one can simply ask for the appropriate bits of the true minimizer and in $O((n+d)\log(1/\varepsilon))$ queries, one can get an ε-approximate solution. A matching lower bound follows from a fairly straightforward counting argument. Thus, allowing for all possible binary queries gives the same information complexity bound as the original Nemirovski-Yudin bound with subgradient queries in the $n=0$ (no integer variables) case, but is an exponential improvement when $n\geq 1$ (see [2] and the discussion below). What this shows is that the bounds on information complexity can be quite different under different oracles. With all possible binary queries, while each query reveals only a single bit of information, the queries themselves are a much richer class and this compensates to give the same bound in the continuous case and exponentially better bounds in the presence of integer variables. Thus, to get a better understanding of this trade-off, we restrict to queries that still extract information by only acting "locally".

1.1 Our contributions

Oracles based on first-order information. Our first contribution is formalizing this notion of general "local" queries. While we focus on first-order information, our framework can be readily extended to consider, for example, information from higher-order derivatives.

Definition 4. *An* oracle using first-order information $\mathcal{O}(\mathcal{G},\mathcal{H})$ *consists of two parts:*

1. *For every* $\mathbf{z}\in[-R,R]^{n+d}$*, there exist three maps* $\mathbf{g}_{\mathbf{z}}^{\text{sep}}:\mathcal{I}_{n,d,R,\rho,M}\to\mathbb{R}^{n+d}$, $\mathbf{g}_{\mathbf{z}}^{\text{val}}:\mathcal{I}_{n,d,R,\rho,M}\to\mathbb{R}$*, and* $\mathbf{g}_{\mathbf{z}}^{\text{sub}}:\mathcal{I}_{n,d,R,\rho,M}\to\mathbb{R}^{n+d}$ *such that for all* $(f,C)\in\mathcal{I}_{n,d,R,\rho,M}$ *the following properties hold.*
   (a) $C\subseteq\{\mathbf{z}'\in\mathbb{R}^{n+d}:\langle\mathbf{g}_{\mathbf{z}}^{\text{sep}}(f,C),\mathbf{z}'\rangle<\langle\mathbf{g}_{\mathbf{z}}^{\text{sep}}(f,C),\mathbf{z}\rangle\}$ *if* $\mathbf{z}\notin C$ *and* $\mathbf{g}_{\mathbf{z}}^{\text{sep}}(f,C)=\mathbf{0}$ *if* $\mathbf{z}\in C$*. In other words,* $\mathbf{g}_{\mathbf{z}}^{\text{sep}}(f,C)$ *returns a (normal vector to a) separating hyperplane if* $\mathbf{z}\notin C$*. We will assume that a nonzero response* $\mathbf{g}_{\mathbf{z}}^{\text{sep}}(f,C)$ *has norm 1, since scalings do not change the separation property.*
   (b) $\mathbf{g}_{\mathbf{z}}^{\text{val}}(f,C)=f(\mathbf{z})$*. In other words,* $\mathbf{g}_{\mathbf{z}}^{\text{val}}(f,C)$ *returns the function value for* f *at* $\mathbf{z}$*.*
   (c) $\mathbf{g}_{\mathbf{z}}^{\text{sub}}(f,C)\in\partial f(\mathbf{z})$*, where* $\partial f(\mathbf{z})$ *denotes the subdifferential (the set of all subgradients) of* f *at* $\mathbf{z}$*. In other words,* $\mathbf{g}_{\mathbf{z}}^{\text{sub}}(f,C)$ *returns a subgradient for* f *at* $\mathbf{z}$*.*

Such maps will be called first-order maps. *A collection of first-order maps, one for every* $\mathbf{z}$*, is called a* first-order chart *and will be denoted by* $\mathcal{G}$.

2. *There are three sets of functions* $\mathcal{H}^{\mathrm{sep}}$, $\mathcal{H}^{\mathrm{val}}$*, and* $\mathcal{H}^{\mathrm{sub}}$ *and with domains* $\mathbb{R}^{n+d}$, $\mathbb{R}$ *and* $\mathbb{R}^{n+d}$ *respectively. We will use the notation* $\mathcal{H} = \mathcal{H}^{\mathrm{sep}} \cup \mathcal{H}^{\mathrm{val}} \cup \mathcal{H}^{\mathrm{sub}}$. $\mathcal{H}$ *will be called the collection of* permissible queries of the oracle.

An algorithm for instances of (1) using $\mathcal{O}(\mathcal{G},\mathcal{H})$ can, at any iteration, choose a point $\mathbf{z}$ and a function $h \in \mathcal{H}$ and receive the response $h(\mathbf{g}_{\mathbf{z}}^{\mathrm{sep}}(\widehat{f},\widehat{C}))$, $h(\mathbf{g}_{\mathbf{z}}^{\mathrm{val}}(\widehat{f},\widehat{C}))$ or $h(\mathbf{g}_{\mathbf{z}}^{\mathrm{sub}}(\widehat{f},\widehat{C})$, depending on whether $h \in \mathcal{H}^{\mathrm{sep}}$, $h \in \mathcal{H}^{\mathrm{val}}$ or $h \in \mathcal{H}^{\mathrm{sub}}$, where $\widehat{f}$ and $\widehat{C}$ are the objective function and feasible region, respectively, of the unknown instance. Hence, queries to an oracle $\mathcal{O}(\mathcal{G},\mathcal{H})$ using first-order information are indexed by $(\mathbf{z}, h)$, $\mathbf{z} \in \mathbb{R}^n \times \mathbb{R}^d, h \in \mathcal{H}$. Since the goal of this paper is to provide bounds for different types of such oracles, i.e., with different permissible queries $\mathcal{H}$, let us define some cases of interest.

Definition 5 (Examples of oracles)**.**

1. *(Full-information first-order oracle) When* $\mathcal{H}$ *consists only of the identity functions, i.e.,* $h^{\mathrm{sep}}(\mathbf{g}_{\mathbf{z}}^{\mathrm{sep}}(\widehat{f},\widehat{C})) = \mathbf{g}_{\mathbf{z}}^{\mathrm{sep}}(\widehat{f},\widehat{C})$, $h^{\mathrm{val}}(\mathbf{g}_{\mathbf{z}}^{\mathrm{val}}(\widehat{f},\widehat{C})) = \mathbf{g}_{\mathbf{z}}^{\mathrm{val}}(\widehat{f},\widehat{C})$ *and* $h^{\mathrm{sub}}(\mathbf{g}_{\mathbf{z}}^{\mathrm{sub}}(\widehat{f},\widehat{C})) = \mathbf{g}_{\mathbf{z}}^{\mathrm{sub}}(\widehat{f},\widehat{C})$*, we recover a full-information first-order oracle.*

2. *(Bit oracle) Let* $\mathcal{H}^{bit}$ *be the set of binary queries that return a desired bit (of a desired coordinate) of the binary representation of* $\mathbf{g}_{\mathbf{z}}^{\mathrm{sep}}(\widehat{f},\widehat{C})$, $\mathbf{g}_{\mathbf{z}}^{\mathrm{val}}(\widehat{f},\widehat{C})$ *or* $\mathbf{g}_{\mathbf{z}}^{\mathrm{sub}}(\widehat{f},\widehat{C})$. *Let* $\mathcal{H}^{bit^*}$ *be the* shifted *bit oracle that additionally returns a desired bit of* $\mathbf{g}_{\mathbf{z}}^{\mathrm{val}}(\widehat{f},\widehat{C}) + u$*, for any* $u \in \mathbb{R}$*, i.e.* $\mathcal{H}^{bit^*}$ *allows querying a bit of the function value shifted by some number.*

3. *(Inner product threshold queries) Let*
$$\begin{aligned}\mathcal{H}^{dir} := &\{h_{\mathbf{u},c}^{\mathrm{sep}} : h_{\mathbf{u},c}^{\mathrm{sep}}(\mathbf{g}_{\mathbf{z}}^{\mathrm{sep}}(\widehat{f},\widehat{C})) = sgn(\langle \mathbf{u}, \mathbf{g}_{\mathbf{z}}^{\mathrm{sep}}(\widehat{f},\widehat{C})\rangle - c), \mathbf{u} \in \mathbb{R}^{n+d}, c \in \mathbb{R}\}\\ &\cup \{h_{u,c}^{\mathrm{val}} : h_{u,c}^{\mathrm{val}}(\mathbf{g}_{\mathbf{z}}^{\mathrm{val}}(\widehat{f},\widehat{C})) = sgn(u \cdot \mathbf{g}_{\mathbf{z}}^{\mathrm{val}}(\widehat{f},\widehat{C}) - c), u \in \mathbb{R}, c \in \mathbb{R}\}\\ &\cup \{h_{\mathbf{u},c}^{\mathrm{sub}} : h_{\mathbf{u},c}^{\mathrm{sub}}(\mathbf{g}_{\mathbf{z}}^{\mathrm{sub}}(\widehat{f},\widehat{C})) = sgn(\langle \mathbf{u}, \mathbf{g}_{\mathbf{z}}^{\mathrm{sub}}(\widehat{f},\widehat{C})\rangle - c), \mathbf{u} \in \mathbb{R}^{n+d}, c \in \mathbb{R}\},\end{aligned}$$
where sgn denotes the sign function, be the set of binary queries that answers whether the inner product of the separating hyperplane, function value or subgradient, with a vector or a number of choice $\mathbf{u}$ *or* u *in the appropriate space, is at least some value* c *or not. We write these as* $\mathcal{H}^{dir}$ *since these queries allow for the choice of a "direction"* $\mathbf{u}$*, or a number* u *in the function value case, as part of the query.*

4. *When* $\mathcal{H}$ *is the the set of all possible binary functions on* $\mathbb{R}^n \times \mathbb{R}^d$ *for the separating hyperplanes,* $\mathbb{R}$ *for the functions values, and* $\mathbb{R}^n \times \mathbb{R}^d$ *for the subgradients, we will call the resulting oracle the* general binary oracle based on $\mathcal{G}$.

These now give us a variety of oracles using first-order information, that clearly provide very different information for each query depending on the choice of permissible queries $\mathcal{H}$. Note that different first-order charts will result in different oracles of each of these types that may give different answers at any point, depending on which separating hyperplane/subgradient the oracle's first-order map selects at those points for that instance.

We are now ready to state our quantitative results for lower and upper bounds on the information complexity of mixed-integer convex optimization under different oracles; see Table 1 for a summary. It is not hard to see that we need to restrict the set of possible instances $\mathcal{I}$ in order to have meaningful (finite) information complexity $\mathrm{icomp}_\varepsilon(\mathcal{I},\mathcal{O})$. We will focus on the following standard parameterization.

Definition 6. *Define $\mathcal{I}_{n,d,R,\rho,M}$ to be the set of all instances of* (1) *such that:*

(i) C is contained in the box $\{\mathbf{z} \in \mathbb{R}^n \times \mathbb{R}^d : \|\mathbf{z}\|_\infty \leq R\}$. The case $C = \{\mathbf{z} \in \mathbb{R}^n \times \mathbb{R}^d : \|\mathbf{z}\|_\infty \leq R\}$ will be called unconstrained.

(ii) If $(\mathbf{x}^\star, \mathbf{y}^\star)$ is an optimal solution of the instance, then there exists $\hat{\mathbf{y}} \in \mathbb{R}^d$ satisfying $\{(\mathbf{x}^\star, \mathbf{y}) : \|\mathbf{y} - \hat{\mathbf{y}}\|_\infty \leq \rho\} \subseteq C$. In other words, there is a "strictly feasible" point $(\mathbf{x}^\star, \hat{\mathbf{y}})$ in the same fiber as the optimum $(\mathbf{x}^\star, \mathbf{y}^\star)$.

(iii) f is Lipschitz continuous with respect to the $\|\cdot\|_\infty$-norm with Lipschitz constant M on $\{\mathbf{x}\} \times [-R, R]^d$ for all $\mathbf{x} \in [-R, R]^n \cap \mathbb{Z}^n$. In other words, for any $(\mathbf{x}, \mathbf{y}), (\mathbf{x}, \mathbf{y}') \in (\mathbb{Z}^n \times \mathbb{R}^d) \cap [-R, R]^{n+d}$ with $\|\mathbf{y} - \mathbf{y}'\|_\infty \leq R$, $|f(\mathbf{x}, \mathbf{y}) - f(\mathbf{x}, \mathbf{y}')| \leq M\|\mathbf{y} - \mathbf{y}'\|_\infty$ with the convention that $\infty - \infty = 0$.

Type of first-order oracle $\mathcal{O}(\mathcal{G}, \mathcal{H})$	Variables	Lower bound	Upper bound
$\mathcal{H}$ is hereditary	Mixed	$\Omega(2^n \ell)$, where $\ell \leq \text{icomp}_\varepsilon(\mathcal{I}_{0,d,R,\rho,M}, \mathcal{O}(\mathcal{G}, \mathcal{H}))$ (Theorem 7)	
Full-information first-order oracle	Mixed	$\Omega\left(2^n d \log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right)$ (Corollary 8)	$O\left(2^n d(n+d) \log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right)$ (Oertel [13], Basu-Oertel [3])
$\mathcal{H}^{bit}, \mathcal{H}^{bit^*}, \mathcal{H}^{dir}$, or General Binary Queries	Mixed	$\tilde{\Omega}\left(2^n \max\left\{d^{\frac{8}{7}}, d \log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right\}\right)$ (Theorem 9)	$O\left(2^n d\,(n+d)^2 \log^2\left(\frac{(n+d)MR}{\min\{\rho,1\}\varepsilon}\right)\right)$ (Theorem 10)
	Continuous	$\tilde{\Omega}\left(\max\left\{d^{\frac{8}{7}}, d \log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right\}\right)$ (Theorem 9)	$O\left(d^2 \log^2\left(\frac{dMR}{\min\{\rho,1\}\varepsilon}\right)\right)$ (Theorem 11)
General Binary	Mixed	$\tilde{\Omega}\left(2^n \max\left\{d^{\frac{8}{7}}, d \log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right\}\right)$ (Theorem 9)	$O\left(\log\lvert\mathcal{I}\rvert + 2^n d(n+d) \log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right)$ (Corollary 13)
	Continuous	$\tilde{\Omega}\left(\max\left\{d^{\frac{8}{7}}, d \log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right\}\right)$ (Theorem 9)	$O\left(\log\lvert\mathcal{I}\rvert + d \log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right)$ (Corollary 13)

Table 1: Summary of results on the information complexity of mixed-integer convex optimization for the class of instances $\mathcal{I}_{n,d,R,\rho,M}$ that have n integer variables, d continuous variables, the feasible region lies in the box $[-R, R]^{n+d}$ and has a "ρ-deep feasible point" on the optimal fiber, and the objective function is M-Lipschitz with respect to ℓ_∞ (see Definition 6). The table presents simplified bounds showing only the main parameters.

Lower bounds. Our first result is a "transfer" theorem that will be a powerful tool for obtaining concrete mixed-integer lower bounds under different oracles. This theorem lifts lower bounds for unconstrained optimization from the continuous to the mixed-integer setting. In particular, if one has a lower bound ℓ with respect to an oracle using first-order information (Definition 4) for the information complexity for some family of purely continuous instances, then one can "transfer" that lower bound to the mixed-integer case as $\Omega(2^n \ell)$ with access to the "same" oracle in the $n + d$ dimensional space. For this notion, we require the set of permissible queries $\mathcal{H}$ to be *hereditary*. Roughly speaking, this means that the set of queries has the same richness on a purely continuous space as it is in a mixed-integer space. We formally define hereditary queries in Section 2, and note that all of the types of permissible queries discussed in Definition 5 satisfy this property, except for $\mathcal{H}^{bit}$ (the slightly enhanced $\mathcal{H}^{bit^*}$ queries are hereditary).

Theorem 7. *Let $\mathcal{H}_{n,d}$ be any class of hereditary permissible queries, and assume $\mathcal{H}_{0,d}$ contains function threshold queries h_c that answer $h_c(\mathbf{g}_{\mathbf{z}}^{\mathrm{val}}(\widehat{f}, \widehat{C})) := sgn(\mathbf{g}_{\mathbf{z}}^{\mathrm{val}}(\widehat{f}, \widehat{C}) + c)$ for any $c \in \mathbb{R}$. Let $\varepsilon \geq 0$. Suppose, for some $d \geq 1$, there exists a class $\mathcal{I} \subseteq \mathcal{I}_{0,d,R,\rho,M}$ of continuous convex (unconstrained) optimization problems in $\mathbb{R}^d$, and a first order chart $\mathcal{G}_0$ for $\mathcal{I}$ such that $\mathrm{icomp}_\varepsilon(\mathcal{I}, \mathcal{O}(\mathcal{G}_0, \mathcal{H}_{0,d})) \geq \ell$. Suppose further that all instances in $\mathcal{I}$ have the same optimal value. Then, for any number of integer variables $n \geq 1$, there is a first order chart $\mathcal{G}_n$ such that $\mathrm{icomp}_\varepsilon(\mathcal{I}_{n,d,R,\rho,M}, \mathcal{O}(\mathcal{G}_n, \mathcal{H}_{n,d})) \geq 2^{n-1}\ell$.*

As a first consequence of this transfer theorem we obtain a sharpened lower bound for the standard full-information first-order oracle case for mixed-integer problems. For this setting, Basu [2] proved the lower bound of $\Omega\big(2^n \cdot d \log\big(\frac{2R}{3\rho}\big)\big)$. However, this bound is independent of the Lipschitz constant M of the objective function, and thus does not capture the hardness of the problem as M increases. By applying Theorem 7 to the classical lower bound of $\Omega\big(d \log\big(\frac{MR}{\varepsilon}\big)\big)$ for continuous convex optimization with the standard first-order oracle by Nemirovski and Yudin [12], and combining the result with the existing mixed-integer lower bound, we obtain the following improved bound.

Corollary 8. *There exists a first-order chart $\mathcal{G}$ such that for the full-information first-order oracle based on $\mathcal{G}$ (i.e., $\mathcal{H}$ consists of the identity functions) we have*

$$\mathrm{icomp}_{\,\varepsilon}(\mathcal{I}_{n,d,R,\rho,M}, \mathcal{O}(\mathcal{G}, \mathcal{H})) = \Omega\left(2^n\left(1 + d\log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right)\right).$$

Moving on to "non-standard" oracles, we consider mixed-integer convex optimization under the general binary oracle. Recall from Definition 5 that this means that the algorithm can make any binary query on subgradients/separating hyperplanes. Despite the power of these queries, we prove a separation between the information complexity under the standard full-information first-order oracle and the general binary oracle, i.e., the latter provides quantitatively less information for solving the problem. For example, in the pure continuous setting, $O(d)$ queries suffice (ignoring the logarithmic dependence on other parameters) under the full-information first-order oracle. However, we show that $\Omega(d^{8/7})$ queries are needed under the general binary oracle. More precisely, we show the following lower bound.

Theorem 9. *For every $n \geq 0$, there exists a first-order chart $\mathcal{G}$ such that for the general binary oracle based on $\mathcal{G}$ we have*

$$\mathrm{icomp}_{\,\varepsilon}(\mathcal{I}_{n,d,R,\rho,M}, \mathcal{O}(\mathcal{G}, \mathcal{H})) = \tilde{\Omega}\left(2^n\left(1 + \max\left\{d^{\frac{8}{7}}, d\log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right\}\right)\right),$$

where $\tilde{\Omega}$ hides polylogarithmic factors in d.

We note that, since $\mathcal{H}^{bit}$, $\mathcal{H}^{bit^*}$ and $\mathcal{H}^{dir}$ are more restrictive than the general binary oracle, this lower bound applies to oracles with those permissible queries as well. The proof of this result relies on a connection between information complexity and *memory constrained* algorithms for convex optimization, and the recent lower bound for the latter from [11] (in addition to Theorem 7 for lifting the result to the mixed-integer case).

Upper bounds. We now present upper bound results that illustrate the connection between information complexity based on full-information first-order oracles and information complexity based on binary queries on separating hyperplanes and subgradients. We first formalize the intuition that by making roughly $O\left((n+d)\log\left(\frac{1}{\varepsilon}\right)\right)$ bit or inner product sign queries on a separating hyperplane or subgradient, one should have enough information to solve the problem as with full information (Theorems 10 and 11). Next, in Theorem 12 and Corollary 13, we show how this natural bound can be improved in certain settings.

Theorem 10. *Assume $d \geq 1$. For $U > 0$, consider the subclass of instances of $\mathcal{I}_{n,d,R,\rho,M}$ whose objective function values lie in $[-U, U]$, and the fiber over the optimal solution contains a $\mathbf{z}$ such that the $(n+d)$-dim ρ-radius ℓ_∞ ball centered at $\mathbf{z}$ is contained in C. There exists a query strategy for this subclass that reports an ε-approximate solution by making at most*

$$O\left(2^n d\,(n+d)\log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right)\cdot\left((n+d)\log\left(\frac{(n+d)MR}{\rho\varepsilon}\right)+\log\frac{U}{\varepsilon}\right)$$

queries to an oracle $\mathcal{O}(\mathcal{G},\mathcal{H})$, where $\mathcal{G}$ is any first-order chart and $\mathcal{H}$ is either $\mathcal{H}^{\text{bit}}$ or $\mathcal{H}^{\text{dir}}$.

Prescribing an *a priori* range for objective function values is not a serious restriction for two reasons: i) The difference between the maximum and the minimum values of an objective function in $\mathcal{I}_{n,d,R,\rho,M}$ is at most $2MR$, and ii) All optimization problems whose objective functions differ by a constant are equivalent. We also comment that while we assume $d \geq 1$ in Theorem 10, similar bounds can be established for the $d = 0$ (pure integer) case. We omit this here because a unified expression for the $d = 0$ and $d \geq 1$ cases becomes unwieldy and difficult to parse.

The main idea behind Theorem 10 is to show that existing methods with the best known information complexity for mixed-integer convex optimization that use full-information first-order oracles can also work with approximate separation and subgradient oracles that return desired approximations of the true vectors (with no loss in the information complexity). Then one shows that one can produce these approximations with roughly $O\left((n+d)\log\left(\frac{1}{\varepsilon}\right)\right)$ bit or inner product sign queries on a separating hyperplane or subgradient. With bit queries, this is just a matter of probing enough bits of each coordinate of the vector. The case with inner product sign queries is a bit more involved and our main tool is a result that shows how to approximate any vector up to desired accuracy with such queries (Lemma 30).

Subsequently, using similar techniques we present an enhanced upper bound for the scenario where $n = 0$ (pure continuous case).

Theorem 11. *For $U > 0$, consider the subclass of instances of $\mathcal{I}_{n,d,R,\rho,M}$ where $n = 0$ (pure continuous case) and the objective function values lie in $[-U, U]$. There exists a query strategy for this subclass that reports an ε-approximate solution by making at most*

$$O\left(d\log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right)\cdot\left(d\log\left(\frac{dMR}{\rho\varepsilon}\right)+\log\frac{U}{\varepsilon}\right)$$

queries to an oracle $\mathcal{O}(\mathcal{G},\mathcal{H})$, where $\mathcal{G}$ is any first-order chart and $\mathcal{H}$ is either $\mathcal{H}^{\text{bit}}$ or $\mathcal{H}^{\text{dir}}$.

Finally, we provide a kind of transfer result that allows one to transfer algorithms designed for full-information first-order oracles to the (harder) setting of a general binary oracle.

Theorem 12. *Suppose there exists an algorithm that reports an ε-approximate solution for instances in $\mathcal{I}_{n,d,\rho,M,R}$ with at most u queries to the full-information first-order oracle based on a first-order chart $\mathcal{G}$. Then, for any subclass of finitely many instances $\mathcal{I} \subset \mathcal{I}_{n,d,R,\rho,M,R}$, there* *exists a query strategy for this subclass using the general binary oracle based on $\mathcal{G}$ that reports an ε-approximate solution by making at most*

$$O\big(\log|\mathcal{I}| + u\big)$$

queries.

Using the centerpoint-based algorithm from [13, 3], we obtain the following corollary:

Corollary 13. *Given any subclass of finitely many instances $\mathcal{I} \subset \mathcal{I}_{n,d,R,\rho,M}$ and any first-order chart $\mathcal{G}$, there exists a query strategy for this subclass using the general binary oracle based on $\mathcal{G}$ that reports an ε-approximate solution by making at most*

$$O\left(\log|\mathcal{I}| + 2^n\, d\,(n+d)\log\left(\frac{dMR}{\min\{\rho,1\}\varepsilon}\right)\right)$$

queries. In the (pure continuous) case of $n = 0$, $O\left(\log|\mathcal{I}| + d\log\left(\frac{MR}{\min\{\rho,1\}\varepsilon}\right)\right)$ queries suffice.

In particular, when the number of instances under consideration is $|\mathcal{I}| = O(2^{2^n d(n+d)})$, Corollary 13 gives a strictly better upper bound than Theorem 10. Similarly, for $n = 0$, in the case when $|\mathcal{I}| = O(2^d)$, we get a better upper bound compared to Theorem 11; in fact, we beat the lower bound provided by Theorem 9. This demonstrates that even with exponentially many instances under consideration, the case of finite instances yields a lower information complexity than the case of infinitely many instances. We point out that the first-order chart $\mathcal{G}$ must be known to implement the query strategy in Theorem 12. In contrast, the algorithms in Theorems 10 and 11 are oblivious of the first order chart, i.e., they work with any first order chart.

1.2 Discussion and future avenues

The concept of information complexity in continuous convex optimization and its study go back several decades, and it is considered a fundamental question in convex optimization. In comparison, much less work on information complexity has been carried out in the presence of integer constrained variables. Nevertheless, we believe there are important and challenging questions that come up in that domain that are worth studying. Further, even within the context of continuous convex optimization, the notion of information complexity has almost exclusively focused on the number of full-information first-order queries. As we hope to illustrate with the results of this paper, considering other kinds of oracles leads to very interesting questions at the intersection of mathematical optimization and information theory. In particular, the study of binary oracles promises to give a more refined understanding of the fundamental question "How much information about an optimization instance do we need to be able to solve it with provable guarantees?". For instance, establishing *any superlinear (in the dimension)* lower bound for the continuous problem with binary oracles, like the one in Theorem 9, seems to be nontrivial. In fact, the results from [11], on which Theorem 9 is based, were considered a breakthrough in establishing superlinear lower bounds on space complexity of convex optimization. Even so, the right bound is conjectured to be quadratic in the dimension (see Theorem 11) and our Theorem 9 is far from that at this point. These other oracles also have a practical motivation. Obtaining exact first-order information may be difficult or impossible in many practical situations, and one has to work with approximations of separating hyperplanes and subgradients. The binary oracles can be viewed as providing these approximations and information complexity under these oracles becomes important from a practical standpoint.

We thus view the results of this paper as expanding our understanding of information complexity of optimization in two different dimensions: what role does the presence of integer variables play and what role does the nature of the oracle play (with or without integer variables)? For the role of integer variables, in the pure optimization case Theorem 7 provides a lifting of lower bound from the continuous case. Allowing for constraints, Corollary 8 brings the lower bound closer to the best known upper bound on information complexity based on the classical subgradient oracle. The remaining gap is now simply a factor linear in the dimension. A conjecture in convex geometry first articulated in [13, Conjecture 4.1.20] and elaborated upon in [3, 2] would resolve this and would show that the right bound is essentially equal to the lower bound we prove in this paper.

Beyond the contributions discussed above, our work also opens up new future directions for study. We believe the following additional conjectures to be good catalysts for future research, especially in regard to understanding the interplay of integer variables and other oracles.

The first conjecture is a generalization of our Theorem 7 to incorporate constraints as well. This would make this "transfer" tool more powerful, and would, for example, give Corollary 8 as a special case without appealing to [2] for the feasibility lower bound.

Conjecture 1. *If there exist continuous,* ***constrained*** *convex optimization instances such that ℓ is a lower bound for this family on the information complexity with respect to an oracle, then for every $n \geq 1$, there exist mixed-integer instances with n integer variables such that the information complexity of these mixed-integer instances is lower bounded by $\Omega(2^n \cdot \ell)$ for the same oracle.*

Another consequence of resolving this conjecture is that if future research on the information complexity of continuous convex optimization results in better/different lower bounds based on feasibility, these would immediately imply new lower bounds for the mixed-integer case. For instance, we believe the following conjecture to be true for the mixed-integer convex optimization problem.

Conjecture 2. *There exists a first-order chart $\mathcal{G}$ such that the general binary oracle based on $\mathcal{G}$ has information complexity $\Omega\left(2^n\left(1 + d^2 \log\left(\frac{MR}{\rho\varepsilon}\right)\right)\right)$.*

A version of Conjecture 2 is also stated in the language of "memory-constrained" algorithms in [16, 11] for the continuous case (see Section 3 below); the way we have stated the conjecture here presents its transfer to the mixed-integer case.

Analogously, it would be nice to have "transfer" theorems for upper bounds as well. In the spirit of Theorems 10, 11 and 12, we believe a useful result would be a theorem that takes upper bound results proved in the full-information first-order oracle setting and obtains upper bound results in the general binary oracle setting. A use case of such a result would be the following: if the upper bound for the general mixed-integer problem with full-information first-order oracles is improved by resolving the convex geometry conjecture mentioned above (and we believe the lower bound is correct and the upper bound is indeed loose), then this would also give better upper bounds for the general binary oracle setting. Thus, we make the following conjecture.

Conjecture 3. *If there exists a query strategy with worst case information complexity $u(n, d, R, \rho, M, \mathcal{G})$ under the full-information first-order oracle based on a first-order chart $\mathcal{G}$, then there exists a query strategy with worst case information complexity bounded by*

$$u(n, d, R, \rho, M, \mathcal{G}) \cdot O\left((n + d) \log\left(\frac{MR}{\rho\varepsilon}\right)\right)$$

under the general binary oracle based on $\mathcal{G}$.

We focus on oracles that use first order information in this paper (Definitions 4 and 5). Oracles that use "zero-order information" have also been studied in the literature, beginning with the seminal work of Yudin and Nemirovski [12]; see [8] for an exposition of how those ideas can be used in the mixed-integer setting and [5] for an exposition in the nonconvex setting. Such oracles report function values only for the objective, with no subgradient information, and only report membership for the constraints, with no separating hyperplanes. A related oracle is the "value comparison" oracle that has found many applications. These oracles comprise of questions of the form "Is $f(\mathbf{z}) \leq f(\mathbf{z}')$?", with no access to the subgradients of f. Such algorithms are particularly useful in learning from users' behaviors, since while a user typically cannot accurately report its (dis)utility value $f(\mathbf{z})$ for an option $\mathbf{z}$, it can more reliably compare the values $f(\mathbf{z})$ and $f(\mathbf{z}')$ of two options; see [10, 14] and references therein for discussions and algorithms in the continuous convex case. The mixed-integer setting under the value comparison oracle has been extensively studied in recent work [4, 6, 7, 15]. The ideas in this paper can also be adapted to give algorithms for mixed-integer convex optimization using the comparison oracle, but we do not undertake a deeper study here. There seems to be scope for future research in this direction, especially in tying together these different strands of ideas for "zero order information".

The remainder of this paper is dedicated to the formal proofs of our main results discussed above.

2 Proof of Theorem 7

The high-level idea for the proof of Theorem 7 is to construct difficult mixed-integer instances by taking hard instances of the continuous case, "placing" one of them on each fiber $\mathbf{x} \times \mathbb{R}^d, \mathbf{x} \in \{0,1\}^n$, and interpolating between fibers appropriately. We do this in a way such that effectively one needs to solve the continuous problems obtained by restricting to each fiber, which leads the $\Omega(2^n \ell)$ lower bound from an ℓ lower bound on the continuous problems – namely, there will be one difficult function from the continuous case placed on each of the 2^n fibers, so if one can't do better than solving each of them separately, one ends up with an $\Omega(2^n \ell)$ lower bound. To make this idea work, the interpolation needs to be done in a way that no query in the full $[0,1]^n \times \mathbb{R}^d$ space reveals information about two (or more) of the continuous functions placed on different fibers, or reveals significantly more information about a function on a fiber than a query on that fiber would. For example, we need to ensure that a single query at the point $(\frac{1}{2}, \ldots, \frac{1}{2}, \mathbf{y})$ for $\mathbf{y} \in R^d$ does not reveal information about multiple functions on different fibers.

2.1 Game-theoretic perspective

So far we have described the information complexity of optimization using an oracle $\mathcal{O}$ over a family of instances $\mathcal{I}$ based on having an optimization algorithm that in each round t makes a query q_t to $\mathcal{O}$ and receives as answer the result $q_t(\widehat{I})$ for the unknown instance $\hat{I}$ it is trying to optimize. However, for obtaining lower bounds on the information complexity, it is more helpful to consider the algorithm as interacting with an *adversary* for the family of instances under $\mathcal{O}$, instead of the unknown instance $\hat{I}$. More precisely, at round t, the adversary receives the query q_t of the algorithm and produces, possibly based on all the previous queries $q_1, \ldots, q_{t-1}$, a response r_t. The only requirement is that there must always exist at least one instance $\bar{I} \in \mathcal{I}$ that is *consistent* with all of its responses, namely $r_t = q_t(\bar{I})$ for all t, under the oracle $\mathcal{O}$ being considered. With each such response, the set of instances that are consistent with all responses given may change, motivating the following definition:

Definition 14. *Given a class of instances $\mathcal{I}$, an oracle $\mathcal{O}$, and a transcript of query-response pairs $(q_1, r_1), ..., (q_t, r_t)$, the set of* surviving instances *for $(q_1, r_1), ..., (q_t, r_t)$ under $\mathcal{O}$ is*

$$\{I \in \mathcal{I} : q_j(I) = r_j \, \forall \, j \in [t]\},$$

i.e., the set of instances consistent with the responses in the transcript under the oracle $\mathcal{O}$. When all instances in $\mathcal{I}$ are unconstrained, let the set of surviving functions *be the set of functions corresponding to the surviving instances.*

We say that an adversary **Adv** is *ε-hard for ℓ rounds* if for any algorithm **Alg**, after ℓ rounds there are surviving instances in $\mathcal{I}$ that do not have a common ε-approximate solution, i.e., if $q_1, \dots, q_\ell$ and $r_1, \dots, r_\ell$ are **Alg**'s queries and **Adv**'s responses, respectively, then there is a collection of instances $\mathcal{J} \subset \mathcal{I}$ that have no common ε-approximate solution but such that $r_t = q_t(I)$ for all $I \in \mathcal{J}$ and $t = 1, \dots, \ell$. Since the sets of ε-approximate solutions of instances in $\mathcal{I}_{n,d,R,\rho,M}$ are compact convex sets, this collection $\mathcal{J}$ of instances may always be taken to be finite[1]. Intuitively, the existence of such an adversary should imply that no algorithm can reliably report an ε-approximate solution within ℓ iterations, that is, $\mathrm{icomp}_\varepsilon(\mathcal{O}, \mathcal{I}) > \ell$. The next result shows that this adversary-based perspective is indeed equivalent to information complexity, and may be of independent interest (for a proof see Appendix A).

Lemma 15. *Consider a class of instances $\mathcal{I}$ and an oracle $\mathcal{O}$. Then* $\mathrm{icomp}_\varepsilon(\mathcal{I}, \mathcal{O}) > \ell$ *if and only if there exists an adversary under $\mathcal{O}$ using $\mathcal{I}$ that is ε-hard for ℓ rounds.*

2.2 Proof for the full-information first-order oracle

The full proof of Theorem 7 is a bit technical and requires a few conceptual connections. For a better exposition, we first prove the theorem in the case that the oracle is the full-information first-order oracle. As $\mathcal{H}$ thus consists of the identity maps, throughout this subsection we will write oracles using first-order information as $\mathcal{O}(\mathcal{G})$, where $\mathcal{G}$ is the corresponding first order chart.

Given the assumption of the theorem and the equivalent adversarial perspective from Lemma 15, assume there is a family of continuous, unconstrained instances $\mathcal{I}_{cont} \subseteq \mathcal{I}_{0,d,R,\rho,M}$, all with the same optimal value OPT, and a full-information first-order adversary **Adv-Cont** for $\mathcal{I}$ that is ε-hard for $\ell - 1$ rounds. Let us use $\mathcal{F}_{cont}$ to denote the objective functions of the instances $\mathcal{I}_{cont}$. In the full-information first-order case, queries of an optimization algorithm consist of points $\mathbf{y}_1, \mathbf{y}_2, \dots \in \mathbb{R}^d$, and either query the function value or the subgradient. For simplicity, let us allow the algorithm to query *both* the function value and subgradient in a single query, so that the queries become simply $\mathbf{y}_1, \mathbf{y}_2, \dots \in \mathbb{R}^d$ and the responses of an adversary consist of a sequence of consistent function values and subgradients, namely a sequence $(v_1, \mathbf{g}_1), (v_2, \mathbf{g}_2), \dots \in \mathbb{R} \times \mathbb{R}^d$ such that there is some $f \in \mathcal{F}_{cont}$ satisfying $v_t = f(\mathbf{y}_t)$ and $\mathbf{g}_t \in \partial f(\mathbf{y}_t)$ for all rounds t.

To prove the theorem, we will construct a full-information first-order adversary **Adv-MI** for a family of mixed-integer instances over $\{0,1\}^n \times \mathbb{R}^d$ that is ε-hard for $2^n \ell - 1$ rounds. As alluded to before, the very high-level is to place a copy of the continuous adversary **Adv-Cont** on each of the continuous fibers $\mathbf{x} \times \mathbb{R}^d$ for $\mathbf{x} \in \{0,1\}^n$. In fact, we will work with a slightly modified version of the continuous adversary that is constructed next.

2.2.1 Modifying the continuous adversary Adv-Cont

For the mixed-integer adversary **Adv-MI**, it will be important to render a fiber "useless" for the optimization algorithm after it queries (close to) this fiber too many times, so as to intuitively

[1]If a collection of compact sets has empty intersection, then there exists a finite subcollection that already has empty intersection.

force it to query (close to) other fibers, or gain no new information otherwise. This will be done by modifying the continuous adversary **Adv-Cont** such that whenever it is probed ℓ or more times, it commits to answering all future queries consistently with a *single* function that has optimal value $>$ OPT $+\, \varepsilon$; since our mixed-integer instances will be constructed to have optimal value OPT, gathering more information about the function on such fibers will not help the algorithm solve the mixed-integer problem. To do this, the modified continuous adversary will also keep track of the set S of surviving functions (Definition 14) given its responses. More precisely, here are its main properties.

Lemma 16. *There is a family of convex functions $\overline{\mathcal{F}}_{cont}$ corresponding to instances $\mathcal{I}_{0,d,R,\rho,M}$ of the purely continuous case, a first-order chart $\mathcal{G}$ and a full-information first-order adversary **Adv-Cont+** that, for any algorithm **Alg**, maintains a set of functions $S_t \subseteq \overline{\mathcal{F}}_{cont}$ for every query-response round t with the following properties:*

1. *In every round $t \geq 1$, all functions in S_t are consistent with the responses returned by **Adv-Cont+** thus far, under some oracle using first-order information $\mathcal{O}(\mathcal{G})$.*
2. *In the first $t \leq \ell - 1$ rounds, there is a finite collection of functions in $S_t \cap \mathcal{F}_{cont}$ that do not share an ε-approximate solution. In particular, **Adv-Cont+** is still ε-hard for $\ell - 1$ rounds.*
3. *For all rounds $t \geq 1$, S_t is closed under taking maxima of finitely many of its elements, and also contains a function that has minimum value $>$ OPT $+\, \varepsilon$.*
4. *For rounds $t \geq \ell$, S_t contains a single function with minimum value $>$ OPT $+\, \varepsilon$.*

Item 1. means that for rounds $t < \ell$, there exists a full-information first-order oracle $\mathcal{O}(\mathcal{G})$ such that S_t is exactly the set of surviving functions under $\mathcal{O}(\mathcal{G})$ given the responses produced by **Adv-Cont+** up to round t. Hence, we will refer to this S_t as the set of *surviving functions maintained by **Adv-Cont+** at round t.* We now make precise our modification to the continuous adversary **Adv-Cont** and prove Lemma 16. As a preliminary, let $\overline{\mathcal{F}}_{cont}$ denote the closure of $\mathcal{F}_{cont}$ under taking maxima of finitely many functions, i.e. for any finite collection $\mathcal{J} \subset \mathcal{F}_{cont}$, $\max_{f \in \mathcal{J}}\{f\} \in \overline{\mathcal{F}}_{cont}$. Notice these functions are still convex. The following lemma highlights the key property of $\mathcal{F}_{cont}$ we will make use:

Lemma 17. *Let $\mathcal{J} \subset \mathcal{F}$ be a finite set. If $f \in \mathcal{J}$ do not have a common ε-solution, then the pointwise maximium function $\max_{f \in \mathcal{J}}\{f\}$ has minimum value greater than $OPT + \varepsilon$.*

Proof. Suppose for sake of contradiction that there exists a point $\mathbf{z}$ such that $\max_{f \in \mathcal{J}}\{f(\mathbf{z})\} \leq OPT + \varepsilon$. Then $f(\mathbf{z}) \leq OPT + \varepsilon$ for all $f \in \mathcal{J}$, which means $\mathbf{z}$ is an ε-solution for all $f \in \mathcal{J}$, which contradicts the assumption that they do not share an ε-solution. $\square$

We now formally describe **Adv-Cont+**, and then prove that it satisfies the invariants of Lemma 16.

Procedure 1. Adv-Cont+

Initialize set of surviving functions $S_0 = \overline{\mathcal{F}}_{cont}$

For each round $t = 1, 2 \ldots$:

1. Receive query point $\mathbf{y}_t \in \mathbb{R}^d$ from the optimization algorithm
2. If $t \leq \ell - 1$: Send $\mathbf{y}_t$ to the adversary **Adv-Cont**, receiving back a value v_t and subgradient $\mathbf{g}_t$. Obtain S_t by removing from S_{t-1} the functions f that are not consistent with this response for any first-order chart $\mathcal{G}$, namely where $f(\mathbf{y}_t) \neq v_t$ or $\mathbf{g}_t \notin \partial f(\mathbf{y}_t)$.

Send the response $(v_t, \mathbf{g}_t)$ to the optimization algorithm.

3. If $t = \ell$: Since **Adv-Cont** if ε-hard for $\ell - 1$ rounds, there is a finite collection of functions $\{f_1, ..., f_k\} \subset S_{t-1} \cap \mathcal{F}_{cont}$ that do not share an ε-solution. Define their pointwise maxima $f_{\max} = \max\{f_1, ..., f_k\}$ and set $S_{t+k} = \{f_{\max}\}$, for all $k = 0, 1, 2....$

   Set the value v_t to be $f_{\max}(\mathbf{y}_t)$ and set $\mathbf{g}_t$ to be a subgradient in $\partial f_{\max}(\mathbf{y}_t)$ (consistent with what the first order chart $\mathcal{G}_0$ gives for $f_1, \dots, f_k$ at $\mathbf{y}_t$, if $\mathbf{y}_t$ has been queried in an earlier round), and send the response $(v_t, \mathbf{g}_t)$ to the optimization algorithm.

4. If $t > \ell$: Let $f_{\max}$ be the only function in S_{t-1}. If $\mathbf{y}_t$ was queried in an earlier round k, answer $(v_k, \mathbf{g}_k)$. Otherwise, set the value v_t to be $f_{\max}(\mathbf{y}_t)$ and set $\mathbf{g}_t$ to be any subgradient in $\partial f_{\max}(\mathbf{y}_t)$, and send the response $(v_t, \mathbf{g}_t)$ to the optimization algorithm.

Proof of Lemma 16. We will proceed by induction on the number of rounds t. The lemma clearly holds for S_0, so suppose it holds for S_{t-1}.

If $t \le \ell - 1$, then S_t satisfies Item 1 due to to the update rule in the procedure for obtaining S_t, since all functions that are not consistent with the given response are removed. More precisely, since the responses given are those produced by **Adv-Cont**, these functions in S_t are consistent with the responses under exactly the oracle $\mathcal{O}(\mathcal{G}_0)$ that **Adv-Cont** is hard under. S_t also satisfies Item 2 because **Adv-Cont** is assumed to be ε-hard for $\ell - 1$ rounds, so there exists a finite collection of functions $\{f_1, ..., f_k\} \subset \mathcal{F}_{cont}$ with no common ε-solution that are consistent with all responses given to **Adv-Cont+** by **Adv-Cont**; thus S_t contains them. For Item 3, to show the closure under taking maxima, we need to argue that if functions $f_1, ..., f_k$ were not removed from S, then neither was $\max(f_1, ..., f_k)$. Since $f_1, ..., f_k$ are convex, then $\partial f_j(\mathbf{y}) \subset \partial \max\{f_1, ..., f_k\}(\mathbf{y})$ for any j such that $f_j(\mathbf{y}) = \max\{f_1(\mathbf{y}), ..., f_k(\mathbf{y})\}$. Hence, if $f_1, ..., f_k$ all have function value v_t and subgradient $\mathbf{g}_t$ at $\mathbf{y}$, then so does $\max\{f_1, ..., f_k\}$, so $\max\{f_1, ..., f_k\}$ was not removed from S, as desired. Furthermore, if $f_1, ..., f_k$ are taken to be the functions guaranteed by Item 2, Lemma 17 implies that $\max\{f_1, ..., f_k\}$ has optimal value greater than OPT $+ \varepsilon$, so since we just showed $\max\{f_1, ..., f_k\} \in S_t$, the remainder of item 3 follows.

If $t = \ell$, S_t contains the single function $f_{\max}$, which has optimal value greater than OPT $+ \varepsilon$ by its construction as a consequence of Lemma 17. Hence, Item 3 follows. To prove item 1, we will use that S_{t-1} satisfies Item 1, and by Item 3 applied to S_{t-1}, $f_{\max}$ is consistent with the responses returned by the procedure up to round $t - 1$. For round t itself, consistency follows from the definition of v_t and $\mathbf{g}_t$, and so $f_{\max}$ is consistent with all responses given. Item 2 does not apply in this case and Item 4 is immediate by the construction of $S_t := \{f_{\max}\}$.

If $t > \ell$, then $S^t = S^{t-1} = \{f_{\max}\}$ and it suffices to check that the response $(v_t, \mathbf{g}_t)$ is compatible with $f_{\max}$, which follows immediately from the definition of the response.

Hence, Items 1-4 of the lemma follow. It remains to show that **Adv-Cont+** is indeed a well-defined adversary under a full-information first-order oracle. For rounds $t \le \ell - 1$, this is inherited from **Adv-Cont**, while for $t \ge \ell$, this is ensured because if the queried point $\mathbf{y}_t$ is the same as $\mathbf{y}_{t'}$ for some round $t' < t$, **Adv-Cont+** provides the same response in round t as in round t'. Thus, there is indeed a first-order chart $\mathcal{G}$ (derived from the first order map $\mathcal{G}_0$ for **Adv-Cont**) such that **Adv-Cont+** is an adversary under the corresponding full-information first-order oracle $\mathcal{O}(\mathcal{G})$. □

2.2.2 Constructing the mixed-integer adversary Adv-MI

We now construct the family $\mathcal{F}_{MI}$ of functions over $\mathbb{R}^n \times \mathbb{R}^d$ used to transfer the lower bound to the mixed-integer setting, along with the adversary **Adv-MI** for that family. We call functions over $\mathbb{R}^n \times \mathbb{R}^d$ *full-dimensional* to distinguish them from the functions over $\mathbb{R}^d$, the continuous part of the problem. As indicated previously, these full-dimensional functions ψ in $\mathcal{F}_{MI}$ will be obtained by considering combinations of selecting one function $f_{\bar{\mathbf{x}}}$ from $\overline{\mathcal{F}}_{cont}$ for each of the mixed-integer fibers $\bar{\mathbf{x}} \times \mathbb{R}^d, \bar{\mathbf{x}} \in \{0,1\}^n$, letting ψ equal the appropriate function selected over each corresponding fiber, and applying an interpolation scheme between the fibers. This interpolation is illustrated in Figure 1 and described in detail later in this section.

For the behavior of **Adv-MI**, we instantiate a copy of the modified continuous adversary **Adv-Cont+** on each fiber. Whenever the optimization algorithm queries a point $(\bar{\mathbf{x}}, \mathbf{y})$ on a fiber, we send $\mathbf{y}$ to the continuous adversary on the fiber and report back the response $(v, \mathbf{g})$ received, although $\mathbf{g}$ needs to be appropriately lifted to the full $\mathbb{R}^n \times \mathbb{R}^d$ space to be consistent with the way we interpolate the functions between the fibers. If the optimization algorithm only probes on these fibers, then it is intuitive that such an adversary would be ε-hard for $2^n\ell - 1$ rounds: informally, up to this round, at least one of the 2^n fibers that has received no more than $\ell - 1$ queries, so using the hardness of **Adv-Cont+** (Item 2 of Lemma 16) we can obtain full-dimensional functions that do not share an ε-approximate solution, which confirm the desired ε-hardness of the mixed integer adversary **Adv-MI**.

The crucial element is how to deal with queries on points outside of the mixed-integer fibers. If such queries provide the algorithm with more information about the full-dimensional functions $\mathcal{F}_{MI}$ than queries on the fibers do, then we may not have full-dimensional functions with no common ε-approximate solution surviving for $2^n\ell - 1$ rounds. To handle this issue, the interpolation used to define the full-dimensional functions ψ guarantees that its behavior on a fractional point $(\tilde{\mathbf{x}}, \mathbf{y}) \notin \{0,1\}^n \times \mathbb{R}^d$ is completely determined by the value of the function $f_{\bar{\mathbf{x}}}(\mathbf{y})$ from $\overline{\mathcal{F}}_{cont}$ selected for the fiber $\bar{\mathbf{x}} \times \mathbb{R}^d$, where $\bar{\mathbf{x}} \in \{0,1\}^n$ is the closest 0/1 point to $\tilde{\mathbf{x}}$. Thus, **Adv-MI** can also answer such a query at a fractional point by making a query to the appropriate continuous adversary **Adv-Cont+** on $\{\bar{\mathbf{x}}\} \times \mathbb{R}^d$, and the hardness of the latter can still be leveraged.

We now formally define the functions $\mathcal{F}_{MI}$ and the adversary **Adv-MI**.

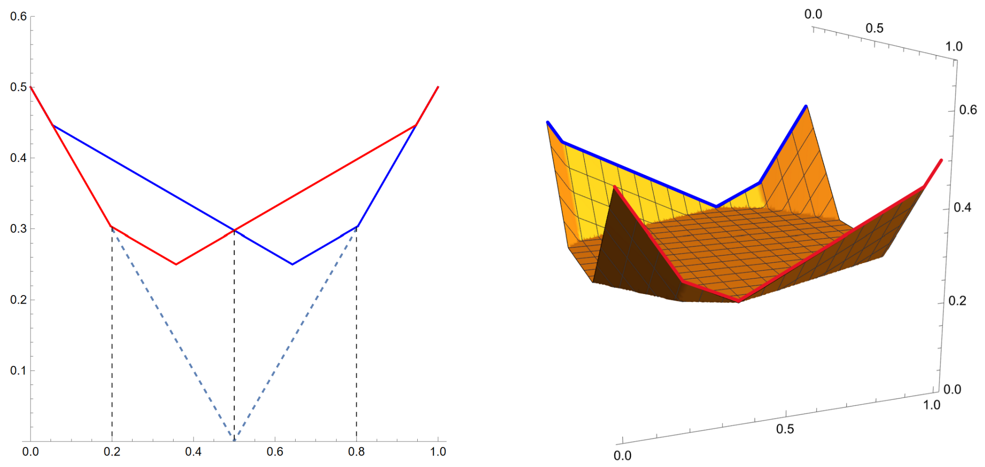

Figure 1: (Left) Illustration of two possible functions $f_0, f_1 \in \overline{\mathcal{F}}_{cont}$ (blue and red) of the continuous adversary **Adv-Cont+**, for $d = 1$. (Right) Illustration of the function $\psi_{(f_0,f_1)}$ for the mixed-integer adversary **Adv-MI** obtained by placing the functions f_0 and f_1 on the fibers $\{0\} \times \mathbb{R}$ and $\{1\} \times \mathbb{R}$ and interpolating appropriately between the fibers.

Construction of the functions $\mathcal{F}_{MI}$. For a 0/1 point $\bar{\mathbf{x}} \in \{0,1\}^n$ and a function $f \in \bar{\mathcal{F}}_{cont}$ in $\mathbb{R}^d$, we first define its (convex) extension to the full-dimensional space $\mathbb{R}^{n+d}$ as

$$\hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}, \mathbf{y}) = \max\Big\{ f(\mathbf{y}) + \langle \mathbf{M}_{\bar{\mathbf{x}}}, \mathbf{x} - \bar{\mathbf{x}} \rangle \ , \ \mathrm{OPT} \Big\}, \tag{2}$$

with $\mathbf{M}_{\bar{\mathbf{x}}} := 3MR \cdot sgn(\bar{\mathbf{x}} - 0.5 \cdot \mathbf{1})$, where sgn denotes the sign function. This construction effectively places f along the $\mathbf{y}$ space at the fiber $\bar{\mathbf{x}}$ and extends it in each of the $\mathbf{x}$ variables via a linear function with slope $\pm 3MR$, in a way that it decreases the value as it moves into the unit cube, or equivalently, away from $\bar{\mathbf{x}}$; it then truncates the final value to being at least OPT; see Figure 2 for an illustration. We note for later use that wherever the extension is not truncated by OPT, a subgradient is given by appending the vector $\mathbf{M}_{\bar{\mathbf{x}}}$ to a subgradient of f, and otherwise the all zeroes vector is a subgradient. More precisely, we have

$$\partial \hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}, \mathbf{y}) \supseteq \begin{cases} \{\mathbf{M}_{\bar{\mathbf{x}}}\} \times \partial f(\mathbf{y}) & , \text{if } \hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}, \mathbf{y}) > \mathrm{OPT} \\ \{\mathbf{0}\} & , \text{otherwise.} \end{cases} \tag{3}$$

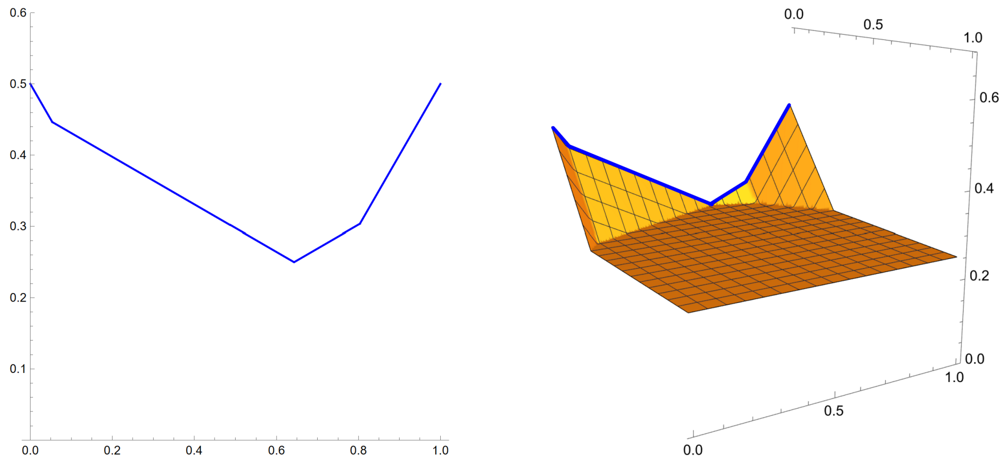

Figure 2: An example of a possible function from $f \in \mathcal{F}_{cont}$ (left) together with an illustration of its truncated extension $\hat{f}_0(\mathbf{x}, \mathbf{y})$ (right) as constructed in (2).

Given a collection $\mathtt{F} = (f_{\bar{\mathbf{x}}})_{\bar{\mathbf{x}}}$ with one function $f_{\bar{\mathbf{x}}} \in \bar{\mathcal{F}}_{cont}$ for each 0/1 point $\bar{\mathbf{x}}$, we combine them into the convex function

$$\psi_{\mathtt{F}}(\mathbf{x}, \mathbf{y}) := \max_{\bar{\mathbf{x}} \in \{0,1\}^n} \hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}, \mathbf{y}), \tag{4}$$

where we abuse notation slightly and write the convex extension $\widehat{(f_{\bar{\mathbf{x}}})}_{\bar{\mathbf{x}}}$ as simply $\hat{f}_{\bar{\mathbf{x}}}$ to simplify the notation. As mentioned above, a crucial property of these functions is that their behavior between the fibers is determined by the behavior on the closest fiber. Intuitively, the slope $\pm 3MR$ guarantees that as the $\mathbf{x}$ argument moves away from the base fiber $\bar{\mathbf{x}}$ of each extended function $\hat{f}_{\bar{\mathbf{x}}}$, $\hat{f}_{\bar{\mathbf{x}}}$ decreases rapidly enough so that the maximum in (4) is always achieved by the extended function at the closest fiber to $\mathbf{x}$. Figure 1 illustrates this, where one can see that both functions placed on the fibers get fully truncated in between the fibers. To make this precise, let $r(\mathbf{x}) : [0,1]^n \to \{0,1\}^n$ map any $\mathbf{x}$ in the box to its closest 0/1 point in ℓ_∞-norm, that is $r(\mathbf{x}) := \mathrm{argmin}_{\mathbf{x}'}\{\|\mathbf{x} - \mathbf{x}'\|_\infty : \mathbf{x}' \in \{0,1\}^n\}$.

Lemma 18. *For every collection $F = (f_{\bar{\mathbf{x}}})_{\bar{\mathbf{x}}}$, for every point $(\mathbf{x}, \mathbf{y}) \in [0,1]^n \times [-R, R]^d$ we have*

$$\psi_F(\mathbf{x}, \mathbf{y}) = \hat{f}_{r(\mathbf{x})}(\mathbf{x}, \mathbf{y}), \qquad \textit{and} \qquad \partial \psi_F(\mathbf{x}, \mathbf{y}) = \partial \hat{f}_{r(\mathbf{x})}(\mathbf{x}, \mathbf{y})$$

Proof. Define $B_{\bar{\mathbf{x}}} := \left\{\mathbf{x}' \in \mathbb{R}^n : \|\mathbf{x}' - \bar{\mathbf{x}}\|_\infty \le \frac{1}{3}\right\}$ for every $\bar{\mathbf{x}} \in \{0,1\}^n$. Consider an arbitrary $(\mathbf{x}, \mathbf{y}) \in [0,1]^n \times [-R, R]^d$.

Case 1: $\mathbf{x} \notin B_{\bar{\mathbf{x}}}$ for any $\bar{\mathbf{x}} \in \{0,1\}^n$. This implies that for every $\bar{\mathbf{x}} \in \{0,1\}^n$, we have

$$\begin{aligned} f_{\bar{\mathbf{x}}}(\mathbf{y}) + \langle \mathbf{M}_{\bar{\mathbf{x}}}, \mathbf{x} - \bar{\mathbf{x}} \rangle &= f_{\bar{\mathbf{x}}}(\mathbf{y}) + \sum_{j \in [n]} 3MR \cdot sgn(\bar{\mathbf{x}}_j - 0.5) \cdot (\mathbf{x}_j - \bar{\mathbf{x}}_j) \\ &\le f_{\bar{\mathbf{x}}}(\mathbf{y}) - 3MR \cdot \max_{j \in [n]} |\mathbf{x}_j - \bar{\mathbf{x}}_j| < f_{\bar{\mathbf{x}}}(\mathbf{y}) - MR \le \mathrm{OPT}, \end{aligned}$$

Thus, $\hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}, \mathbf{y}) = \mathrm{OPT}$ for all $\bar{\mathbf{x}} \in \{0,1\}^n$. As a result, $\psi_F(\mathbf{x}, \mathbf{y}) = \mathrm{OPT} = \hat{f}_{r(\mathbf{x})}(\mathbf{x}, \mathbf{y})$.

Moreover, since $f_{\bar{\mathbf{x}}}(\mathbf{y}) + \langle \mathbf{M}_{\bar{\mathbf{x}}}, \mathbf{x} - \bar{\mathbf{x}} \rangle < \mathrm{OPT}$ for all $\bar{\mathbf{x}} \in \{0,1\}^n$, there exists a neighborhood of $(\mathbf{x}, \mathbf{y})$ such that for any point $(\mathbf{x}', \mathbf{y}')$ in the neighborhood, it holds that $f_{\bar{\mathbf{x}}}(\mathbf{y}') + \langle \mathbf{M}_{\bar{\mathbf{x}}}, \mathbf{x}' - \bar{\mathbf{x}} \rangle < \mathrm{OPT}$, and $\hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}', \mathbf{y}') = \mathrm{OPT}$ for all $\bar{\mathbf{x}} \in \{0,1\}^n$. As a result, $\partial \hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}, \mathbf{y}) = \{\mathbf{0}\}$ for all $\bar{\mathbf{x}} \in \{0,1\}^n$, and $\partial \psi_F(\mathbf{x}, \mathbf{y}) = \{\mathbf{0}\}$.

Case 2: $\mathbf{x} \in B_{\bar{\mathbf{x}}}$ for some $\bar{\mathbf{x}} \in \{0,1\}^n$. In this case, $r(\mathbf{x}) = \bar{\mathbf{x}}$. This is because for any $\tilde{\mathbf{x}} \in \{0,1\}^n \backslash \{\bar{\mathbf{x}}\}$, we have that

$$\|\mathbf{x} - \tilde{\mathbf{x}}\|_\infty \ge \frac{2}{3} > \frac{1}{3} \ge \|\mathbf{x} - \bar{\mathbf{x}}\|_\infty. \tag{5}$$

It is also true that $\hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}, \mathbf{y}) \ge \mathrm{OPT} = \hat{f}_{\tilde{\mathbf{x}}}(\mathbf{x}, \mathbf{y})$ for any $\tilde{\mathbf{x}} \ne \bar{\mathbf{x}}$, which holds due to the result from Case 1.

Moreover, the arguments from Case 1 and (5) imply that there exists a neighborhood of $(\mathbf{x}, \mathbf{y})$ such that for any point $(\mathbf{x}', \mathbf{y}')$ in the neighborhood, it holds that $r(\mathbf{x}') = r(\mathbf{x}) = \bar{\mathbf{x}}$, and $\hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}', \mathbf{y}') \ge \mathrm{OPT} = \hat{f}_{\tilde{\mathbf{x}}}(\mathbf{x}', \mathbf{y}')$ for all $\tilde{\mathbf{x}} \ne \bar{\mathbf{x}}$. As a result, $\psi_F(\mathbf{x}', \mathbf{y}') = \hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}', \mathbf{y}') = \hat{f}_{r(\mathbf{x}')}(\mathbf{x}', \mathbf{y}') = \hat{f}_{r(\mathbf{x})}(\mathbf{x}', \mathbf{y}')$ and $\partial \psi_F(x, y) = \partial \hat{f}_{\bar{\mathbf{x}}}(\mathbf{x}, \mathbf{y}) = \partial \hat{f}_{r(\mathbf{x})}(\mathbf{x}, \mathbf{y})$. □

Construction of the mixed-integer adversary Adv-MI. We finally describe **Adv-MI** in Procedure 2. Its main property is captured in the following invariant.

Procedure 2. Adv-MI

Instantiate a copy of **Adv-Cont+** on each fiber $\mathbf{x} \in \{0,1\}^n$, and let $S(\mathbf{x})$ denote the set of surviving functions maintained in every round by this copy, initialized to $\overline{\mathcal{F}}_{cont}$.

For each round $t = 1, 2 \ldots$:

1. **Adv-MI** receives the query $(\mathbf{x}_t, \mathbf{y}_t)$ from the algorithm. Send $\mathbf{y}_t$ to the adversary **Adv-Cont+** associated with the closest fiber $r(\mathbf{x}_t)$, which then returns a value v and subgradient $\mathbf{g}$, and updates its maintained set of surviving functions $S(r(\mathbf{x}_t))$ of its fiber $r(\mathbf{x}_t)$.

2. **Adv-MI** returns as its response to the query $(\mathbf{x}_t, \mathbf{y}_t)$ the value

$$\tilde{v}_t = \max\left\{v + \langle \mathbf{M}_{r(\mathbf{x}_t)}, \mathbf{x}_t - r(\mathbf{x}_t) \rangle \ , \ \mathrm{OPT}\right\},$$

and as subgradient returns either $\tilde{\mathbf{g}}_t = (\mathbf{M}_{\bar{\mathbf{x}}}, \mathbf{g})$ or $\tilde{\mathbf{g}}_t = \mathbf{0}$ depending whether $\tilde{v}_t > \mathrm{OPT}$ or not (i.e., whether $\hat{f}_{r(\mathbf{x})}$ was truncated at $(\mathbf{x}_t, \mathbf{y}_t)$ or not), respectively.

Invariant 1. *There exists a first order chart $\mathcal{G}$ (derived from $\mathcal{G}_0$) such that, for any algorithm, the sets $S(\mathbf{x})$, $\mathbf{x} \in \{0,1\}^n$ maintained by **Adv-MI** satisfy the following property.*

*In every round, for every collection $\mathtt{F} = (f_{\mathbf{x}})_{\mathbf{x}\in\{0,1\}^n}$ of current surviving functions $f_{\mathbf{x}} \in S(\mathbf{x})$ for $\mathbf{x} \in \{0,1\}^n$, the function $\psi_{\mathtt{F}}$ is consistent with the response returned by **Adv-MI** under the full-information first-order oracle $\mathcal{O}(\mathcal{G})$, i.e., $\psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t) = \tilde{v}_t$ and $\tilde{\mathbf{g}}_t \in \partial\psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t)$.*

Notice that Invariant 1 is indeed maintained after each response in Step 2 of Procedure 2: For every collection $\mathtt{F} = (f_{\bar{\mathbf{x}}})_{\bar{\mathbf{x}}}$ of still surviving functions $f_{\bar{\mathbf{x}}} \in S(\bar{\mathbf{x}})$, by the consistency guarantee of **Adv-Cont+** (Item 1 of Lemma 16) the function $f_{r(\mathbf{x}_t)}$ selected for the fiber $r(\mathbf{x}_t)$ has value v and subgradient $\mathbf{g}$ at $\mathbf{y}_t$; thus, Lemma 18 combined with (2) implies that the function $\psi_{\mathtt{F}}$ has value

$$\psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t) = \hat{f}_{r(\mathbf{x})}(\mathbf{x}_t, \mathbf{y}_t) = \max\left\{ f_{r(\mathbf{x}_t)}(\mathbf{y}_t) + \langle \mathbf{M}_{r(\mathbf{x}_t)}, \mathbf{x}_t - r(\mathbf{x}_t)\rangle \ , \ \mathrm{OPT}\right\} = \tilde{v}_t,$$

and similarly from (3) we see that $\tilde{\mathbf{g}}$ is a subgradient in $\partial\psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t) = \partial\hat{f}_{r(\mathbf{x}_t)}(\mathbf{x}_t, \mathbf{y}_t)$, as desired.

We now prove that **Adv-MI** is ε-hard for $2^n\ell - 1$ rounds; using Lemma 15, this implies Theorem 7 for the case of full-information first-order oracle. Suppose the optimization algorithm runs for fewer than $2^n \cdot \ell$ iterations. Then there is a fiber $\mathbf{x}^* \in \{0,1\}$ where **Adv-MI** sent at most $\ell - 1$ queries to the adversary **Adv-Cont+** of the fiber $\mathbf{x}^*$. Thus, by the guarantee of the latter (Item 2 of Lemma 16), the surviving set $S(\mathbf{x}^*)$ has some finite collection of functions $f^1_{\mathbf{x}^*}, \dots, f^k_{\mathbf{x}^*}$ with no common ε-approximate solution. Consider the collections $\mathtt{F}^1, \dots, \mathtt{F}^k$ of surviving functions that have $f^1_{\mathbf{x}^*}, \dots, f^k_{\mathbf{x}^*}$, respectively, for the fiber $\mathbf{x}^*$ and any function $f_{\bar{\mathbf{x}}} \in S(\bar{\mathbf{x}})$ with optimal value $> \mathrm{OPT} + \varepsilon$ for each of the other fibers $\bar{\mathbf{x}} \neq \mathbf{x}^*$, which exist on each of the other fibers by Item 3 of Lemma 16. By Invariant 1, all functions $\psi_{\mathtt{F}^1}, \dots, \psi_{\mathtt{F}^k}$ are compatible with the responses returned by **Adv-MI**. The desired ε-hardness of **Adv-MI** then follows from the following claim, which then concludes the proof.

Claim 19. *The functions $\psi_{\mathtt{F}^1}, \dots, \psi_{\mathtt{F}^k}$ share no common ε-approximate solution.*

Proof. From the construction above, we have that $\mathtt{F}^\dagger := \mathtt{F}^1 \backslash \{f^1_{\mathbf{x}^*}\} = \mathtt{F}^2 \backslash \{f^2_{\mathbf{x}^*}\} \dots = \mathtt{F}^k \backslash \{f^k_{\mathbf{x}^*}\}$. Due to (4) and the definitions of $\mathtt{F}^1, \dots, \mathtt{F}^k$, for any fiber $\bar{\mathbf{x}} \neq \mathbf{x}^*$ and $f_{\bar{\mathbf{x}}} \in \mathtt{F}^\dagger$, it follows that $\psi_{\mathtt{F}^1}(\bar{\mathbf{x}}, \mathbf{y}) = \dots = \psi_{\mathtt{F}^k}(\bar{\mathbf{x}}, \mathbf{y}) = f_{\bar{\mathbf{x}}}(\mathbf{y}) > \mathrm{OPT} + \varepsilon$. Thus, the ε-approximate solutions for the functions $\psi_{\mathtt{F}^1}, \dots, \psi_{\mathtt{F}^k}$ only exist within the fiber $\mathbf{x}^*$. Given that the ε-approximate solutions of $f^1_{\mathbf{x}^*}, \dots, f^k_{\mathbf{x}^*}$ are disjoint, and considering that $\psi_{\mathtt{F}^j}(\mathbf{x}^*, \mathbf{y}) = f^j_{\mathbf{x}^*}(\mathbf{y})$ for all $j \in [k]$, we can conclude our proof. □

2.3 Proof of Theorem 7 for general oracles

We now prove Theorem 7 in full generality. We will do this using the exact same family of difficult functions $\psi_{\mathtt{F}}$ from (4), and also with the same idea of constructing a mixed-integer adversary that produces its answers by making queries to an adversary for the continuous problems on the fibers. Since the mixed-integer adversary will need to answer queries made in the full $\mathbb{R}^n \times \mathbb{R}^d$ space by making queries in the continuous space $\mathbb{R}^d$ on each fiber, we will require that the set of permissible queries that can be made to the continuous adversary is, in some sense, as rich as the queries allowed in the full space. For example, if one allows full-information queries to be made in $\mathbb{R}^n \times \mathbb{R}^d$, but only binary queries to be made in $\mathbb{R}^d$ to the continuous adversary, one would struggle to determine how the mixed-integer adversary should answer those full-information queries by making only binary queries to the adversaries for the continuous subproblems. Specifically, for a query at $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^d$, knowing how $\mathbf{x}$ affects the function values and subgradients of $\psi_{\mathtt{F}}$, the mixed-integer adversary needs to be able to determine what response to give by making a suitably chosen query about $f_{r(\mathbf{x})}$ to the continuous adversary. We formalize this requirement of having the same richness of queries for the continuous subproblems as for the full $\mathbb{R}^n \times \mathbb{R}^d$ space with the concept of *hereditary queries*.

Hereditary queries. For simplicity, we define the notion of hereditary queries for unconstrained problems (i.e., only for value/subgradient queries), but we remark that the same idea can be applied to separation queries as well.

Definition 20. *Let $\{\mathcal{H}^{\text{val}}_{n,d}\}_{n,d\in\mathbb{N}}$ and $\{\mathcal{H}^{\text{sub}}_{n,d}\}_{n,d\in\mathbb{N}}$ be classes of permissible function value and subgradient queries, respectively, with response sets (codomains) $H^{\text{val}}_{n,d}$ and $H^{\text{sub}}_{n,d}$. $\{\mathcal{H}^{\text{val}}_{n,d}\}_{n,d\in\mathbb{N}}$ and $\{\mathcal{H}^{\text{sub}}_{n,d}\}_{n,d\in\mathbb{N}}$ are said to be* hereditary *if the following holds for all $n, d \in \mathbb{N}$ and functions $\mathcal{M} : \{0,1\}^n \to \mathbb{R}^n$. For any $\mathbf{x} \in \{0,1\}^n$, $\delta \in \mathbb{R}$, $h^{\text{val}} \in \mathcal{H}^{\text{val}}_{n,d}$, and $h^{\text{sub}} \in \mathcal{H}^{\text{sub}}_{n,d}$, there exists $h^{\text{val}}_* \in \mathcal{H}^{\text{val}}_{0,d}$, $h^{\text{sub}}_* \in \mathcal{H}^{\text{sub}}_{0,d}$ and functions $B^{\text{val}} : H^{\text{val}}_{0,d} \to H^{\text{val}}_{n,d}$, $B^{\text{sub}} : H^{\text{sub}}_{0,d} \to H^{\text{sub}}_{n,d}$ such that*

$$B^{\text{val}}(h^{\text{val}}_*(v)) = h_{\text{val}}(v+\delta) \qquad \forall v \in \mathbb{R}, \tag{6}$$

$$B^{\text{sub}}(h^{\text{sub}}_*(v,\mathbf{g})) = h_{\text{sub}}(\mathcal{M}(\mathbf{x}),\mathbf{g}) \;\; \forall \mathbf{g} \in \mathbb{R}^d \tag{7}$$

Intuitively, a class of queries being hereditary has the consequence that if for a point $(\mathbf{x},\mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^d$, one knows exactly the $\mathbf{x}$ component $\mathcal{M}(\mathbf{x})$ of the subgradient, then one can simulate a query in the $\mathbb{R}^n \times \mathbb{R}^d$ space by only making a query on the $\mathbb{R}^d$ space, and similarly that there are queries rich enough to consider shifted function values $v + \delta$, where the interpretation is that δ is the effect $\mathbf{x}$ has on the overall function value – see (2).

Example 21. *We show that natural premissible queries, as from Definition 5, are hereditary. Let $\mathcal{M}(\mathbf{x}), \delta$ be as in Definition 20.*

1. *(Full-information first-order oracle) If $\mathcal{H}^{\text{val}}_{n,d}$ and $\mathcal{H}^{\text{sub}}_{n,d}$ are simply the identity functions, then we can take B^{val} to be $B^{\text{val}}(v) = v + \delta$ and take B^{sub} to be the "lifting/rotation" map $B^{\text{sub}}(\mathbf{g}) = (\mathcal{M}(\mathbf{x}), \mathbf{g})$, noting that h^{sub}, h^{val}, h^{sub}_*, and h^{val}_* are all the identity functions.*

2. *(General binary oracle) For the general binary oracle based on a first-order chart $\mathcal{G}$, B^{val} and B^{sub} can be taken to be the identity map from $\{0,1\}$ to $\{0,1\}$, and one can take $h^{\text{val}}_*(v) = h(v+\delta)$, $h^{\text{sub}}_* = h(\mathcal{M}(\mathbf{x}),\mathbf{g})$, which are permissible queries since all binary queries are permissible.*

3. *(Shifted bit oracle) If $\mathcal{H}^{\text{val}}_{n,d}$ and $\mathcal{H}^{\text{sub}}_{n,d}$ are from a shifted bit oracle $\mathcal{H}^{bit^*}$, then B^{val} can be taken to be the identity map from $\{0,1\}$ to $\{0,1\}$. For a query on the function value, if h^{val} reports some bit of $v+\delta$, then the appropriate hereditary query is exactly the query h^{val}_* such that $h^{\text{val}}_*(v) = h(v+\delta)$, i.e. using the shift $u = \delta$ in the notation of Definition 5. A subgradient bit query $h^{\text{sub}}(\mathcal{M}(\mathbf{x}),\mathbf{g})$ returns a bit of either $\mathcal{M}(\mathbf{x})$ or $\mathbf{g}$, so there are two cases.*

   i) h^{sub} returns the j^{th} bit of the k^{th} entry of $\mathcal{M}(\mathbf{x})$. Set $B^{\text{sub}}(\cdot)$ to return exactly that bit of $\mathcal{M}(\mathbf{x})$, no matter the input to B^{sub}, so h^{sub}_* may be chosen arbitrarily.

   ii) h^{sub} returns the j^{th} bit of the k^{th} entry of $\mathbf{g}$. Set B^{sub} to be the identity map from $\{0,1\}$ to $\{0,1\}$, and set h^{sub}_* to return the desired bit of $\mathbf{g}$.

4. *(Inner product threshold queries) If $\mathcal{H}^{\text{val}}_{n,d}$ and $\mathcal{H}^{\text{sub}}_{n,d}$ consist of the inner product threshold queries, take both B^{val} and B^{sub} to be the identity maps. For function value queries $h^{\text{val}}_{u,c}(v) = sgn(u \cdot (v+\delta) - c)$, use*

$$h^{\text{val}}_*(v) := h^{\text{val}}_{u,c-u\delta}(v) = sgn(uv - (c - u\delta)),$$

since then $h^{\mathrm{val}}_*(v) = sgn(uv - (c - u\delta)) = sgn(u \cdot (v+\delta) - c) = h^{\mathrm{val}}_{u,c}(v)$ as desired.
For subgradient queries $h^{\mathrm{sub}}_{\mathbf{u}}(\mathcal{M}(x), \mathbf{g}) = sgn(\langle \mathbf{u}, \mathcal{M}(x), \mathbf{g}\rangle - c)$, with $\mathbf{u} \in \mathbb{R}^{n+d}$, denote by $\mathbf{u}_n$ the vector of the first n entries of $\mathbf{u}$, and by $\mathbf{u}_d$ the vector of the last d entries of $\mathbf{u}$. One may use

$$h^{\mathrm{sub}}_*(\mathbf{g}) := h^{\mathrm{sub}}_{\mathbf{u}_d, c-\langle \mathbf{u}_n, \mathcal{M}(\mathbf{x})\rangle}(\mathbf{g}) = sgn\Big(\langle \mathbf{u}_d, \mathbf{g}\rangle - (c - \langle \mathbf{u}_n, \mathcal{M}(\mathbf{x})\rangle)\Big),$$

since then we similarly have

$$h^{\mathrm{sub}}_*(\mathbf{g}) = sgn\Big(\langle \mathbf{u}_d, \mathbf{g}\rangle - (c - \langle \mathbf{u}_n, \mathcal{M}(\mathbf{x})\rangle)\Big) = sgn(\langle \mathbf{u}, (\mathcal{M}(x), \mathbf{g})\rangle - c) = h^{\mathrm{sub}}_{\mathbf{u}}(\mathcal{M}(x), \mathbf{g})$$

as desired. For these hereditary queries, note that $h^{\mathrm{val}}_{u,c-u\delta}$ and $h^{\mathrm{sub}}_{\mathbf{u}_d, c-\langle \mathbf{u}_n, \mathcal{M}(\mathbf{x})\rangle}$ are indeed in $\mathcal{H}^{\mathrm{val}}_{0,d}$ and $\mathcal{H}^{\mathrm{sub}}_{0,d}$, respectively, since $u \in \mathbb{R}$, $\mathbf{u}_n \in \mathbb{R}^n$, and $\mathbf{u}_d \in \mathbb{R}^d$.

We remark here that $\mathcal{H}^{bit}$, without the permitted "shifts" allowed in $\mathcal{H}^{bit^*}$ is not hereditary, as it may not satisfy condition (6) for the function values for all δ; however, any lower bounds obtained with $\mathcal{H}^{bit^*}$ must also hold for $\mathcal{H}^{bit}$, since the former is a richer class of queries.

Definition of the adversary. We will define **Adv-Cont+** analogously as in the full-information case, now receiving queries in $\mathcal{H}$ according to the oracle setting considered. **Adv-Cont+** will be ε-hard for ℓ rounds answering queries from $\mathcal{H}$, and after $\ell - 1$ rounds it commits to a single surviving function with optimal value $> OPT + \varepsilon$. As queries for general oracles using first-order information consist of a point $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^n \times \mathbb{R}^d$ and a permissible query $h \in \mathcal{H}$, let us write $(\mathbf{x}, \mathbf{y}, h)$ for notational simplicity to denote such a query.

We describe here the behavior of **Adv-Cont+** in the general oracle case, and such that it satisfies the same invariant of Lemma 16 as in the full-information case, i.e. it is ε-hard for ℓ rounds and only keeps a single surviving function with optimal value at least $OPT + \varepsilon$ after ℓ queries have been made.

Procedure 3. Adv-Cont+

Initialize set of survived functions $S_0 = \overline{\mathcal{F}}_{cont}$

For each round $t = 1, 2 \ldots$:

1. Receive query point $(\mathbf{y}_t, h_t) \in \mathbb{R}^d \times \mathcal{H}$ from the optimization algorithm.
2. If $t \le \ell - 1$: Send $(\mathbf{y}_t, h_t)$ to the adversary **Adv-Cont**, receiving back the answer α. Obtain S_t by removing from S_{t-1} the functions f that are not consistent with this response under any first-order chart $\mathcal{G}$, namely f for which $h_t(f(\mathbf{y}_t)) \neq \alpha$ if $h_t \in \mathcal{H}^{\mathrm{val}}$, or for which there does not exist a $\mathbf{g}_t \in \partial f(\mathbf{y}_t)$ such that $h_t(f(\mathbf{y}_t), \mathbf{g}_t) = \alpha$ if $h_t \in \mathcal{H}^{\mathrm{sub}}$.

   Send the response α to the optimization algorithm.
3. If $t = \ell$: Since **Adv-Cont** is ε-hard for $\ell - 1$ rounds, there is a finite collection of functions $\{f_1, ..., f_k\} \subset S_{t-1} \cap \mathcal{F}_{cont}$ that do not share an ε-solution. Define their pointwise maxima $f_{\max} = \max\{f_1, ..., f_k\}$ and set $S_{t+k} = \{f_{\max}\}$, for all $k = 0, 1, 2 \ldots.$

   Set the value v_t to be $f_{\max}(\mathbf{y}_t)$ and set $\mathbf{g}_t$ to be a subgradient in $\partial f_{\max}(\mathbf{y}_t)$ (consistent with what the first order chart $\mathcal{G}_0$ gives for $f_1, \ldots, f_k$ at $\mathbf{y}_t$, if $\mathbf{y}_t$ has been queried in an earlier round), and send the response $h_t(v_t)$ or $h_t(\mathbf{g}_t)$ to the optimization algorithm, according to whether $h_t \in \mathcal{H}^{\mathrm{val}}$ or $h_t \in \mathcal{H}^{\mathrm{sub}}$ respectively.
4. If $t > \ell$: Let $f_{\max}$ be the only function in S_{t-1}. If $\mathbf{y}_t$ was queried in an earlier round k, answer $(v_k, \mathbf{g}_k)$. Otherwise, as in the step above, set the value v_t to be $f_{\max}(\mathbf{y}_t)$ and set

$\mathbf{g}_t$ to be any subgradient in $\partial f_{\max}(\mathbf{y}_t)$, and again send the appropriate response $h_t(v_t)$ or $h_t(\mathbf{g}_t)$ to the optimization algorithm.

Proving that this **Adv-Cont+** satisfies the invariant from Lemma 16 follows exactly the same steps as in the full-information case. Hence, using this **Adv-Cont+** and the same family of functions $\psi_{\mathtt{F}}$ from (4), we will be able to construct **Adv-MI** for this general oracle case to satisfy a version of Invariant 1, slightly modified for this general case to refine what we mean by functions being consistent with responses given to the more general queries. To achieve this, let **Adv-MI** operate according to the following procedure.

Procedure 4. Adv-MI

Instantiate a copy of **Adv-Cont+** on each fiber $\mathbf{x} \in \{0,1\}^n$, and let $S(\mathbf{x})$ denote the set of surviving functions maintained by this copy, initialized to $\overline{\mathcal{F}}_{cont}$.

Set $\mathbf{M}_x := 3MR \cdot sgn(\mathbf{x} - 0.5 \cdot \mathbf{1})$, for any $\mathbf{x} \in \{0,1\}^n$, as in (2). For each round $t = 1, 2 \ldots$:

1. **Adv-MI** receives the query $(\mathbf{x}_t, \mathbf{y}_t, h_t)$ from the algorithm. Set $\delta := \langle \mathbf{M}_{r(\mathbf{x}_t)}, \mathbf{x}_t - r(\mathbf{x}_t) \rangle$.
2. Send the function value threshold query that answers"is $f_{r(\mathbf{x}_t)}(\mathbf{y}_t) \leq OPT + \delta$?" to the adversary **Adv-Cont+** of the closest fiber $r(\mathbf{x}_t)$, which responds and updates its set of surviving functions $S(r(\mathbf{x}_t))$.

   If the answer is yes, **Adv-MI** responds $h_t(OPT)$ or $h_t(\mathbf{0})$ to the original query, according to whether $h_t \in \mathcal{H}^{\text{val}}$ or $h_t \in \mathcal{H}^{\text{sub}}$, respectively.

   Else, determine an appropriate B and hereditary query h_* as from Definition 20 using $r(\mathbf{x}_t)$, $\mathbf{M}_{r(\mathbf{x}_t)}$, δ and h_t, and send the query $(\mathbf{y}_t, h_*)$ to the adversary **Adv-Cont+** of the closest fiber $r(\mathbf{x}_t)$, which then returns some answer α and updates its set of surviving functions $S(r(\mathbf{x}_t))$. **Adv-MI** returns $B(\alpha)$ as its response to the original query.

Invariant 2. *There exists a first order chart $\mathcal{G}$ (derived from $\mathcal{G}_0$) such that, for any algorithm, the sets $S(\mathbf{x})$, $\mathbf{x} \in \{0,1\}^n$ maintained by **Adv-MI** satisfy the following property.*

*In every round, for every collection $\mathtt{F} = (f_{\mathbf{x}})_{\mathbf{x} \in \{0,1\}^n}$ of current surviving functions $f_{\mathbf{x}} \in S(\mathbf{x})$ for $\mathbf{x} \in \{0,1\}^n$, the function $\psi_{\mathtt{F}}$ is consistent with the response returned by **Adv-MI** under under the oracle $\mathcal{O}(\mathcal{H}, \mathcal{G})$, i.e., the response **Adv-MI** gives is equal to $h_t(\tilde{v}_t, \tilde{\mathbf{g}}_t)$ for $\tilde{v}_t = \psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t)$ and some $\tilde{\mathbf{g}}_t \in \partial \psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t)$.*

We claim that the Invariant 2 is maintained after each response in Step 2 of Procedure 4.

If **Adv-Cont+** answers yes, then $\psi_F(\mathbf{x}_t, \mathbf{y}_t) = OPT$ and $\mathbf{0} \in \partial \psi_F(\mathbf{x}_t, \mathbf{y}_t)$ for any collection of surviving functions $\mathtt{F} = (f_{\bar{\mathbf{x}}})_{\bar{\mathbf{x}}}$, since all the extensions $\hat{f}_{\bar{\mathbf{x}}}$ as in (2) that are consistent with this affirmative response in Step 2 must be truncated at $(\mathbf{x}_t, \mathbf{y}_t)$. To see this, note that due to the choice of δ, the query "is $f_{r(\mathbf{x}_t)}(\mathbf{y}_t) \leq OPT + \delta$?" is equivalent to asking whether $\hat{f}_{r(\mathbf{x}_t)}$ has value OPT at $(\mathbf{x}_t, \mathbf{y}_t)$ (i.e., was truncated), and Lemma 18 guarantees that we only need to consider $\hat{f}_{r(\mathbf{x}_t)}$ for $\psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t)$.

If the answer is no, then Lemma 18 combined with (2) implies that for every choice $\mathtt{F}$ of surviving functions from each fiber, the function $\psi_{\mathtt{F}}$ has value

$$\psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t) = \hat{f}_{r(\mathbf{x})}(\mathbf{x}_t, \mathbf{y}_t) = \max\left\{ f_{r(\mathbf{x}_t)}(\mathbf{y}_t) + \langle \mathbf{M}_{r(\mathbf{x}_t)}, \mathbf{x}_t - r(\mathbf{x}_t) \rangle \ , \ \text{OPT} \right\} = f_{r(\mathbf{x}_t)}(\mathbf{y}_t) + \delta, \tag{8}$$

and for its subgradient we have

$$\mathbf{g} \in \partial \hat{f}_{r(\mathbf{x}_t)} \implies (\mathbf{M}_{r(\mathbf{x}_t)}, \mathbf{g}) \in \partial \psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t), \tag{9}$$

by Lemma 18 and (3).

Suppose first that $h_t \in \mathcal{H}^{\text{val}}$ was a function value query, and denote by B^{val} and h_*^{val} the transformation and hereditary query that **Adv-MI** uses, according to definition 20, giving $B^{\text{val}}(h_*^{\text{val}}(v)) = h_t(v+\delta)$ for all $v \in \mathbb{R}$. For every collection $\mathtt{F} = (f_{\bar{\mathbf{x}}})_{\bar{\mathbf{x}}}$ of surviving functions $f_{\bar{\mathbf{x}}} \in S(\bar{\mathbf{x}})$, by the consistency guarantee of **Adv-Cont+** (Item 1 of Lemma 16), the function $f_{r(\mathbf{x}_t)}$ selected for the fiber $r(\mathbf{x}_t)$ has response $h_*^{\text{val}}(f_{r(\mathbf{x}_t)}(\mathbf{y}_t)) = \alpha$. Then, from the definition of hereditary queries and (8), we have $B^{\text{val}}(h_*^{\text{val}}(f_{r(\mathbf{x}_t)}(\mathbf{y}_t))) = h_t(f_{r(\mathbf{x}_t)}(\mathbf{y}_t) + \delta) = h_t(\psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t))$, and so $\psi_{\mathtt{F}}$ is indeed consistent with the response $B^{\text{val}}(\alpha)$ provided by **Adv-MI**.

If instead $h \in \mathcal{H}^{\text{sub}}$ was a subgradient query, again denote B^{sub} and h_*^{sub} as the appropriate transformation and hereditary query, with $B^{\text{sub}}(h_*^{\text{sub}}(\mathbf{g})) = h(\mathcal{M}(\mathbf{x}), \mathbf{g})$ for all $\mathbf{g} \in \mathbb{R}^d$. Again, for every choice $\mathtt{F}$ of surviving functions on the fibers, the function $f_{r(\mathbf{x}_t)}$ on $r(\mathbf{x}_t)$ has $h_*^{\text{sub}}(\mathbf{g}) = \alpha$, for some $\mathbf{g} \in \partial f_{r(\mathbf{x}_t)}(\mathbf{y}_t)$. Then, from the definition of hereditary queries and (9), $B^{\text{sub}}(h_*^{\text{sub}}(\mathbf{g})) = h_t(\mathbf{M}_{r(\mathbf{x}_t)}, \mathbf{g}) = h_t(\mathbf{g}_\psi)$, with $\mathbf{g}_\psi \in \partial \psi_{\mathtt{F}}(\mathbf{x}_t, \mathbf{y}_t)$. Hence, whether h_t is a function value or subgradient query, all functions $\psi_{\mathtt{F}}$ for choices $\mathtt{F}$ of the surviving functions on the fibers are consistent with the responses given by **Adv-MI**, for the oracle $\mathcal{O}(\mathcal{G}, \mathcal{H})$ with permissible queries $\mathcal{H}$ and the first-order chart $\mathcal{G}$ from Invariant 2.

We now prove that **Adv-MI** is ε-hard for $2^{n-1}\ell - 1$ rounds, thus proving Theorem 7 in the general case. Since **Adv-MI** makes at most 2 queries to **Adv-Cont+** in every round (Step 2) of Procedure 4, if the optimization algorithm runs for fewer than $2^{n-1} \cdot \ell$ iterations, there is a fiber $\mathbf{x}^* \in \{0,1\}$ where **Adv-MI** sent at most $\ell - 1$ hereditary queries to the adversary **Adv-Cont+** of the fiber $\mathbf{x}^*$. Thus, by the guarantee of the latter (Item 2 of Lemma 16), the surviving set $S(\mathbf{x}^*)$ has some finite collection of functions $f_{\mathbf{x}^*}^1, \dots, f_{\mathbf{x}^*}^k$ with no common ε-approximate solution, and the remainder of the proof follows as in the full-information case, by considering $\psi_{\mathtt{F}^1}, \dots, \psi_{\mathtt{F}^k}$ that have $f_{\mathbf{x}^*}^1, \dots, f_{\mathbf{x}^*}^k$ on the fiber $\mathbf{x}^*$, and some functions with optimal value greater than $OPT + \varepsilon$ on all the other fibers.

3 Proof of Theorem 9

To demonstrate Theorem 9, we need to introduce the idea of *information memory* of any query strategy/algorithm.

Definition 22. *A first-order query strategy with* information memory *comprises three functions:*

1. $\phi_{\text{query}} : \{0,1\}^* \to [-R,R]^n \times [-R,R]^d$
2. $\phi_{\text{update}}^{\text{sep}} : \left(\mathbb{R}^n \times \mathbb{R}^d\right) \times \{0,1\}^* \to \{0,1\}^*$
3. $\phi_{\text{update}}^{\text{val}} : \mathbb{R} \times \{0,1\}^* \to \{0,1\}^*$
4. $\phi_{\text{update}}^{\text{sub}} : \left(\mathbb{R}^n \times \mathbb{R}^d\right) \times \{0,1\}^* \to \{0,1\}^*$,

where $\{0,1\}^*$ denotes the set of all binary strings (finite sequences over $\{0,1\}$), including the empty string.

Given access to a first-order chart $\mathcal{G}$, the query strategy maintains an information memory r_k *at every iteration $k \geq 0$, which is a finite length binary string in $\{0,1\}^*$, with r_0 initialized as the empty string. At every iteration $k = 1, 2, \dots$, the query strategy computes $\mathbf{z}_k := \phi_{query}(r_{k-1})$ and* *updates its memory using either* $r_k = \phi^{\text{sep}}_{\text{update}}\left(\mathbf{g}^{\text{sep}}_{\mathbf{z}_k}(\widehat{f}, \widehat{C}), r_{k-1}\right)$, $r_k = \phi^{\text{val}}_{\text{update}}\left(\mathbf{g}^{\text{val}}_{\mathbf{z}_k}(\widehat{f}, \widehat{C}), r_{k-1}\right)$ *or* $r_k = \phi^{\text{sub}}_{\text{update}}\left(\mathbf{g}^{\text{sub}}_{\mathbf{z}_k}(\widehat{f}, \widehat{C}), r_{k-1}\right)$*, where* $(\widehat{f}, \widehat{C})$ *is the unknown true instance. After finitely many iterations, the query strategy does a final computation based on its information memory and reports an* ε*-approximate solution, i.e., there is a final function* $\phi_{\text{fin}} : \{0, 1\}^* \to \mathbb{Z}^n \times \mathbb{R}^d$.

The information memory complexity *of an algorithm for an instance is the maximum length of its information memory* r_k *over all iterations* k *during the processing of this instance.*

The following proposition allows us to relate the information memory complexity of first-order algorithms with information complexity under access to a general binary oracle using first-order information.

Proposition 23. *Let* $\mathcal{G}$ *be a first-order chart. For any first-order query strategy* $\mathcal{A}$ *with information memory that uses* $\mathcal{G}$*, there exists a query strategy* $\mathcal{A}'$ *using the general binary oracle based on* $\mathcal{G}$*, such that for any instance* (f, C)*, if* $\mathcal{A}$ *stops after* T *iterations with information memory complexity* Q*,* $\mathcal{A}'$ *stops after making at most* $Q \cdot T$ *oracle queries.*

Conversely, for any query strategy $\mathcal{A}'$ *using the general binary oracle based on* $\mathcal{G}$*, there exists a first-order query strategy* $\mathcal{A}$ *with information memory such that for any instance* (f, C)*, if* $\mathcal{A}'$ *stops after* T *iterations,* $\mathcal{A}$ *stops after making at most* T *iterations with information memory complexity at most* T*.*

Proof. Let $\mathcal{A}$ be a first-order query strategy with information memory. We can simulate $\mathcal{A}$ by the query strategy whose queries are precisely the bits of the information memory state r_k at each iteration k of $\mathcal{A}$. More formally, the query is $\mathbf{z} = \phi_{\text{query}}(r_{k-1})$ and $h(\cdot) = (\phi^{\text{sep}}_{\text{update}}(\cdot, r_{k-1}))_i$, $h(\cdot) = (\phi^{\text{val}}_{\text{update}}(\cdot, r_{k-1}))_i$, or $h(\cdot) = (\phi^{\text{sub}}_{\text{update}}(\cdot, r_{k-1}))_i$, depending on which type of query was made, where i indexes different bits of the corresponding binary string.

Conversely, given a query strategy $\mathcal{A}'$ based on the general binary oracle, we can simulate it with a first-order query strategy with information memory where in each iteration, we simply append the new bit queried by $\mathcal{A}'$ to the current state of the memory. □

We need the following result derived from Marsden et al. [11] on information memory complexity.

Theorem 24. [11, Theorem 1] *For every* $\delta \in [0, 1/4]$*, there is a class of instances* $\mathcal{I} \subseteq \mathcal{I}_{n,d,R,\rho,M}$*, where* $n = 0$*, and a first-order chart* $\mathcal{G}$ *such that any first-order query strategy with information memory must have either* $d^{1.25-\delta}$ *information memory complexity (in the worst case) or make at least* $\tilde{\Omega}(d^{1+\frac{4}{3}\delta})$ *iterations (in the worst case).*

Proof of Theorem 9. In the case when $n = 0$, we can set $\delta = \frac{3}{28}$ in Theorem 24 to obtain that any first-order query strategy uses either $d^{8/7}$ information memory or makes at least $\tilde{\Omega}(d^{8/7})$ iterations. Using the second part of Proposition 23, we obtain the lower bound of $\tilde{\Omega}(d^{8/7})$ on the number of queries made by any query strategy using the general binary oracle based on $\mathcal{G}$.

Applying Theorem 7 enables us to extend the bound to the mixed-integer scenario $(n > 0)$. Further, by integrating this with Corollary 8, we can obtain the desired bound. □

4 Proof of Theorems 10 and 11

We will use $B_\infty(\mathbf{p}, \delta)$ to denote the ℓ_∞ ball of radius δ centered at $\mathbf{p} \in \mathbb{R}^n \times \mathbb{R}^d$, i.e., $B_\infty(\mathbf{p}, \delta) = \{\mathbf{z} \in \mathbb{R}^n \times \mathbb{R}^d : \|\mathbf{z} - \mathbf{p}\|_\infty \le \delta\}$. Recall that we consider the subclass of instances $(f, C) \in \mathcal{I}_{n,d,R,\rho,M}$ such that the fiber containing the optimal solution also contains a point that is ρ-*deep in* C, that is: if $(\mathbf{x}^*, \mathbf{y}^*) \in \mathbb{Z}^n \times \mathbb{R}^d$ is an optimal solution for this instance, then there is a point $(\mathbf{x}^*, \bar{\mathbf{y}})$ such that the full-dimensional ball $B_\infty((\mathbf{x}^*, \bar{\mathbf{y}}), \rho)$ is contained in C. We use $\mathcal{I}^{deep}_{n,d,R,\rho,M}$ to denote this subclass of instances. We will use $C_{-\rho} := \{\mathbf{z} \in C : B_\infty(\mathbf{z}, \rho) \subseteq C\}$ to denote the set of all ρ-deep points in C.

Our strategy for proving Theorems 10 and 11 is to: 1) solve the the problems using approximate subgradients/separating hyperplanes; 2) use bit queries/inner product sign queries to construct such approximations.

For the first item, we use an algorithm designed by Oertel [13] (see also [3]) based on the concept of a *centerpoint*: this is a point in the convex set where every halfspace supported on it cuts off a significant (mixed-integer) volume of the set. The algorithm maintains an outer relaxation P of the feasible region C in every iteration, and repeatedly applies separation or subgradient-based cuts through the centerpoint of P. The assumption that the feasible region contains a ball (in the optimal fiber) establishes a volume lower bound that essentially limits the number of iterations of the algorithm. While the original algorithm in [13, 3] uses exact separation/subgradient oracles, we show, not surprisingly, that approximate ones suffice. To prove Theorem 11, we employ a similar approach. However, due to the continuous nature of the setting, we can obtain a better upper bound compared to Theorem 10 by applying a stronger bound on the centerpoints from Grünbaum [9].

The next item is to construct approximate separation/subgradient oracles by making only a limited number of binary queries on the separating hyperplanes and/or subgradients. In case of bit queries $\mathcal{H}^{\text{bit}}$ this can be easily done by querying enough bits of the latter. The case of inner product sign queries $\mathcal{H}^{\text{dir}}$, where we can pick a direction $\mathbf{a}$ and ask "Is $\langle \mathbf{a}, \mathbf{g} \rangle \geq 0$?" for the subgradient or separating hyperplane $\mathbf{g}$, is more interesting. It boils down to approximating the vector $\mathbf{g}$ (subgradient/separating hyperplane) using few such queries.[2]

To formalize the first item, we begin by defining three approximate oracles as follows.

Definition 25. *We have the following:*

- *An* ε-approximate separation oracle $\hat{\mathbf{g}}^{\text{sep}}$ *is such that* $\hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C) = \mathbf{0}$ *iff* $\bar{\mathbf{z}}$ *belongs to* C*, and otherwise the cut* $\langle \hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C), \mathbf{z} \rangle \leq \langle \hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C), \bar{\mathbf{z}} \rangle$ *is valid for all* ε-deep points $\mathbf{z} \in C_{-\varepsilon}$.
- *An* ε-approximate value cut oracle $\hat{\mathbf{g}}^{\text{sub}}$ *is such that for every* $\mathbf{z}$ *such that* $\langle \hat{\mathbf{g}}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C), \mathbf{z} \rangle \geq \langle \hat{\mathbf{g}}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C), \bar{\mathbf{z}} \rangle$*, we have* $f(\mathbf{z}) \geq f(\bar{\mathbf{z}}) - \varepsilon$.
- *An* ε-approximate value comparison oracle *is such that for every function* $f : [-R, R]^{n+d} \to [-U, U]$ *and every pair of points* $\mathbf{z}, \mathbf{z}'$ *we obtain the answer to the query "Is* $f(\mathbf{z}) \leq f(\mathbf{z}') + \varepsilon$*?"*.

Then the first item can be formalized as the following.

Theorem 26. *There exists an algorithm that, for any* $M, R > 0$, $0 < \varepsilon \leq MR$ *and* $\rho > 0$, *can report an* ε*-approximate solution for every instance in* $\mathcal{I}^{deep}_{n,d,R,\rho,M}$*, using at most*

$$O\left(2^n (n+d) d \log\left(\frac{MR}{\min\{\rho, 1\}\varepsilon}\right)\right)$$

oracle calls, given access to any ρ'*-approximate separation oracle,* ε'*-approximate value cut oracle, and* ε'*-approximate value comparison oracle with* $\rho' = \frac{\varepsilon' \rho}{4MR}$ *and* $\varepsilon' = \frac{\varepsilon}{6}$*.*

[2]This is related to (actively) learning the linear classifier whose normal is given by $\mathbf{g}$ [1]. These methods can perhaps be adapted to our setting, but we present a different and self-contained statement and proof. See the discussion at the end of Section 4.2.

For the continuous setting with $n = 0$, the bound can be improved to

$$O\left(d \log\left(\frac{MR}{\min\{\rho, 1\}\varepsilon}\right)\right).$$

We postpone the proof of Theorem 26 to Section 4.1.

The next lemma shows that one can implement the approximate oracles from Definition 25 using bit queries and inner product sign queries.

Lemma 27. *Consider a first-order chart $\mathcal{G}$. Let $f : \mathbb{R}^{n+d} \to \mathbb{R}$ be a convex M-Lipschitz function taking values in $[-U, U]$, and $C \subseteq [-R, R]^{n+d}$ a convex set.*

Then for every pair of points $\bar{\mathbf{z}}, \bar{\mathbf{z}}' \in [-R, R]^{n+d}$, we can obtain an ε-approximate separation oracle vector $\hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)$, an ε-approximate value cut vector $\hat{\mathbf{g}}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C)$, and an ε-approximate value comparison between $\bar{\mathbf{z}}$ and $\bar{\mathbf{z}}'$ using either a sequence of bit queries from $\mathcal{H}^{\text{bit}}$, or a sequence of inner product sign queries from $\mathcal{H}^{\text{dir}}$, on the separating hyperplane $\mathbf{g}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)$, the subgradient $\mathbf{g}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C)$ and the function value $\mathbf{g}^{\text{val}}_{\bar{\mathbf{z}}}(f, C)$. The number of required queries to implement the approximate oracles is $O\left((n+d)\log\frac{(n+d)R}{\varepsilon}\right)$, $O\left((n+d)\log\frac{(n+d)MR}{\varepsilon}\right)$ and $O\left(\log\frac{U}{\varepsilon}\right)$ respectively, for both $\mathcal{H}^{\text{bit}}$ and $\mathcal{H}^{\text{dir}}$.

The proof of Lemma 27 is deferred to Section 4.2.

Proof. Theorems 10 and 11 follow from Theorem 26 and Lemma 27. ☐

4.1 Proof of Theorem 26

We first describe the centerpoint algorithm for convex optimization due to Oertel [13] (see also [3]). Let the *mixed-integer volume* of a (Borel) set $U \in \mathbb{R}^{n+d}$ be $\mu(U) := \sum_{\mathbf{x}\in\mathbb{Z}^n} \text{vol}_d(U \cap (\{\mathbf{x}\} \times \mathbb{R}^d))$, where vol_d is the d-dimensional Lebesgue measure. The following notion is the main element of the algorithm.

Theorem 28 (Mixed-integer centerpoint [13, 3]). *For any compact convex set $C \subseteq \mathbb{R}^{n+d}$, there is a point $\mathbf{z} \in C \cap (\mathbb{Z}^n \times \mathbb{R}^d)$ (called a* mixed-integer centerpoint*) such that for every halfspace H with $\mathbf{z}$ on its boundary, we have $\mu(C \cap H) \geq \frac{1}{2^n(d+1)}\,\mu(C)$.*

Algorithm 1 below is the centerpoint-based algorithm for solving mixed-integer convex optimization problems from [13], restated in terms of approximate separation ($\hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)$), value cut ($\hat{\mathbf{g}}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C)$) and value comparison oracles.

To analyze this algorithm we need the following technical lemma regarding deep points in instances in $\mathcal{I}^{deep}_{n,d,R,\rho,M}$.

Lemma 29. *For every instance (f, C) in $\mathcal{I}^{deep}_{n,d,R,\rho,M}$, and for every $0 < \varepsilon < 2MR$, there is an ε-approximate solution $\mathbf{z}$ such that the ball $B_\infty(\mathbf{z}, \frac{\varepsilon\rho}{2MR})$ is contained in C.*

Proof. Let $\mathbf{z}^* = (\mathbf{x}^*, \mathbf{y}^*)$ be an optimal solution for the instance, and let $\bar{\mathbf{z}} = (\mathbf{x}^*, \bar{\mathbf{y}})$ be such that $B_\infty(\bar{\mathbf{z}}, \rho)$ is contained in C. For $\alpha = \frac{\varepsilon}{2MR}$, we claim that the point $\mathbf{z} := (1-\alpha)\mathbf{z}^* + \alpha\bar{\mathbf{z}}$ has the desired properties. First, by convexity of C we have that the desired ball $B_\infty(\mathbf{z}, \frac{\varepsilon\rho}{2MR}) = (1-\alpha)\mathbf{z}^* + \alpha B_\infty(\bar{\mathbf{z}}, \rho)$ is contained in C. In addition, since $\mathbf{z} - \mathbf{z}^* = \alpha \cdot (0, \bar{\mathbf{y}} - \mathbf{y}^*)$ and f is M-Lipschitz over the integer fibers, we have

$$f(\mathbf{z}) \;\leq\; f(\mathbf{z}^*) + \alpha M \cdot \|\mathbf{y}^* - \bar{\mathbf{y}}\|_\infty \;\leq\; f(\mathbf{z}^*) + \varepsilon,$$

where the last inequality uses that $\mathbf{y}^*, \bar{\mathbf{y}} \in [-R, R]^d$ and the definition of α; so $\mathbf{z}$ is an ε-approximate solution. This concludes the proof. ☐

Algorithm 1 Centerpoints

1. Initialize the version set $P_0 := [-R, R]^{n+d}$ and the collection of feasible points $F = \emptyset$. For iterations $t = 0, \dots, T-1$:

   (a) Let $\mathbf{z}_t \in P_t \cap (\mathbb{Z}^n \times \mathbb{R}^d)$ be a mixed-integer centerpoint of P_t given by Lemma 28.

   (b) If the ρ'-approximate separation oracle says that $\mathbf{z}_t$ is infeasible for C, add the cut $\langle \hat{\mathbf{g}}^{\text{sep}}_{\mathbf{z}_t}(f, C), \mathbf{z}\rangle \le \langle \hat{\mathbf{g}}^{\text{sep}}_{\mathbf{z}_t}(f, C), \mathbf{z}_t\rangle$ to P_t, namely set $P_{t+1} = P_t \cap \{\mathbf{z} : \langle \hat{\mathbf{g}}^{\text{sep}}_{\mathbf{z}_t}(f, C), \mathbf{z}\rangle \le \langle \hat{\mathbf{g}}^{\text{sep}}_{\mathbf{z}_t}(f, C), \mathbf{z}_t\rangle\}$

   (c) Else, add $\mathbf{z}_t$ to the set of feasible solutions F, and add the cut from the ε'-approximate value cut oracle, namely set $P_{t+1} = P_t \cap \{\mathbf{z} : \langle \hat{\mathbf{g}}^{\text{sub}}_{\mathbf{z}_t}(f, C), \mathbf{z}\rangle \le \langle \hat{\mathbf{g}}^{\text{sub}}_{\mathbf{z}_t}(f, C), \mathbf{z}_t\rangle\}$.

2. Finally, return a point $\hat{\mathbf{z}}$ from F that has approximately the minimum value among all solutions in F, namely such that $f(\hat{\mathbf{z}}) \le \min_{\mathbf{z}\in F} f(\mathbf{z}) + \varepsilon'$. This can be accomplished by asking $|F| - 1$ queries to the ε'-approximate value comparison oracle.

Proof of Theorem 26. We show that Algorithm 1 with number of iterations set as

$$T = 2^n (n+d)(d+1)\ln\left(\frac{2R}{\min\{\rho', 1\}}\right) \in O\left(2^n (n+d) d \ln\left(\frac{MR}{\min\{\rho, 1\}\varepsilon}\right)\right)$$

has the desired properties First, regarding the number of oracle queries performed: in each iteration it performs at most 2 approximate separation/value cut queries, and in Step 2 it performs $|F| - 1 \le T$ approximate value comparison queries. In total, the algorithm performs at most $3T$ queries, giving the desired complexity.

Now we show that the algorithm returns an ε-optimal solution. For that, it suffices to show that for this value of T, the set of feasible solutions F contains an $\frac{\varepsilon}{2}$-optimal solution.

Using Lemma 29, let $\bar{\mathbf{z}}$ be an ε'-approximate solution such that the ball $B_\infty(\bar{\mathbf{z}}, \frac{\varepsilon'\rho}{2MR}) = B_\infty(\bar{\mathbf{z}}, 2\rho')$ is contained in C. Thus, the ball $B_\infty(\bar{\mathbf{z}}, \rho')$ is contained in $C_{-\rho'}$. Since the cut added to P_t, whether in Step (b) or (c), goes through the centerpoint $\mathbf{z}_t$, the mixed-integer volume of P_t is reduced by a factor of at least $(1 - \frac{1}{2^n(d+1)})$ in each iteration (to simplify the notation let $\alpha := \frac{1}{2^n(d+1)}$). The definition of T shows that the last set P_T has mixed-integer volume at most

$$(1-\alpha)^T \mu(P_0) = (1-\alpha)^T (2R)^{n+d} \le e^{-T\alpha}(2R)^{n+d} \le \left(\min\{\rho', 1\}\right)^{n+d} \le \left(\min\{\rho', 1\}\right)^d. \quad (10)$$

Let X be the intersection of $B_\infty(\bar{\mathbf{z}}, \rho')$ with the mixed-integer fiber containing $\bar{\mathbf{z}}$. X has the same structure as an ℓ_∞ ball of radius ρ' in $\mathbb{R}^d$, and thus has volume at least $(2\rho')^d$, which is strictly bigger than the right-hand side of (10). This means that some mixed-integer point from X is cut off by one of the hyperplanes applied by the algorithm. However, such a hyperplane cannot be one added in Step (b), since $B_\infty(\bar{\mathbf{z}}, \rho') \subseteq C_{-\rho'}$ and the cuts in that step are valid for $C_{-\rho'}$. Thus, there is an iteration t that added a Step (c) approximate value cut that cut off a point $\tilde{\mathbf{z}} \in X$. Thus, $\langle \hat{\mathbf{g}}^{\text{sub}}_{\mathbf{z}_t}(f, C), \tilde{\mathbf{z}}\rangle > \langle \hat{\mathbf{g}}^{\text{sub}}_{\mathbf{z}_t}(f, C), \mathbf{z}_t\rangle$. Since this is an ε'-approximate value cut, we get that $f(\tilde{\mathbf{z}}) \ge f(\mathbf{z}_t) - \varepsilon'$. Since f is M-Lipschitz on the fiber containing $\bar{\mathbf{z}}$ and $\tilde{\mathbf{z}}$, and the ℓ_∞ distance between $\bar{\mathbf{z}}$ and $\tilde{\mathbf{z}}$ is at most ρ', we get

$$f(\mathbf{z}_t) \le f(\tilde{\mathbf{z}}) + \varepsilon' \le f(\bar{\mathbf{z}}) + \rho' M + \varepsilon' \le \text{OPT} + 2\varepsilon' + \rho' M \le \text{OPT} + \frac{\varepsilon}{2},$$

where the last inequality uses that $\rho' = \frac{\varepsilon'\rho}{4MR} \le \frac{\varepsilon'}{M}$ and that $\varepsilon' = \frac{\varepsilon}{6}$.

This shows that the set of feasible solutions F contains an $\frac{\varepsilon}{2}$-approximate solution, namely the above $\mathbf{z}_t$, as desired. This concludes the proof for the mixed-integer setting.

For the continuous setting with $n = 0$, the improved bound follows from using an improved bound on centerpoints due to Grünbaum [9]. Specifically, α in the left hand side of (10) can be taken to be $\frac{1}{e}$, where e is Euler's constant. $\square$

4.2 Proof of Lemma 27

The proof will be divided into two parts: first, we show how to obtain approximate oracles using bit queries, after which we show how to do the same using inner product sign queries.

Obtaining approximate oracles using bit queries. Since f is M-Lipschitz with respect to the ℓ_∞ norm, any subgradient has ℓ_∞ norm at most M. Thus, letting $\varepsilon' := \frac{\varepsilon}{2(n+d)R}$, we will query the sign and the bits indexed by the integers $\lceil \log M \rceil, \lceil \log M \rceil - 1, \ldots, -\lfloor \log \frac{1}{\varepsilon'} \rfloor$ of each coordinate of $\mathbf{g}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C)$ (nonnegative integers index the bits before the decimal, and negative integers index the bits after the decimal in the binary representation). This can be done by querying the bits of $\mathbf{g}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C)$ for a total of $(n+d)(\log \frac{M}{\varepsilon'} + 2)$ queries – for each coordinate, one queries $\log M + \log(\frac{1}{\varepsilon}) + 1$ bits for the desired precision and one additional bit for the sign. This gives a vector $\hat{\mathbf{g}}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C)$ such that $\|\mathbf{g}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C) - \hat{\mathbf{g}}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C)\|_\infty \le \sum_{i > \log \frac{1}{\varepsilon'}} \frac{1}{2^i} \le \varepsilon'$.

Then $\hat{\mathbf{g}}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C)$ is an ε-approximate value cut. Let $\mathbf{g} := \mathbf{g}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C)$ and $\hat{\mathbf{g}} := \hat{\mathbf{g}}^{\text{sub}}_{\bar{\mathbf{z}}}(f, C)$ to simplify notation. For every $\mathbf{z} \in [-R, R]^{n+d}$ such that $\langle \hat{\mathbf{g}}, \mathbf{z} \rangle \ge \langle \hat{\mathbf{g}}, \bar{\mathbf{z}} \rangle$ we have by convexity of f

$$\begin{aligned} f(\mathbf{z}) - f(\bar{\mathbf{z}}) \ge \langle \mathbf{g}, \mathbf{z} - \bar{\mathbf{z}} \rangle = \underbrace{\langle \hat{\mathbf{g}}, \mathbf{z} - \bar{\mathbf{z}} \rangle}_{\ge 0} &+ \langle \mathbf{g} - \hat{\mathbf{g}}, \mathbf{z} - \bar{\mathbf{z}} \rangle \\ &\ge -\|\mathbf{g} - \hat{\mathbf{g}}\|_\infty \cdot \|\mathbf{z} - \bar{\mathbf{z}}\|_1 \\ &\ge -2\varepsilon'(n+d)R \;=\; -\varepsilon, \end{aligned} \tag{11}$$

where the second inequality follows from Hölder's inequalty, and so $\hat{\mathbf{g}}$ has the desired property.

For $\hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)$, recall that by assumption the separating vector $\mathbf{g}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)$ has unit length, and hence $\|\mathbf{g}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)\|_\infty \le 1$. Then querying the sign and the bits indexed by $0, -1, \ldots, -\log \frac{1}{\varepsilon'}$ of each coordinate of $\mathbf{g}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)$ we obtain a vector $\hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)$ such that $\|\mathbf{g}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C) - \hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)\|_\infty \le \varepsilon'$.

We claim that $\hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)$ is an ε-approximate separation oracle, namely the inequality $\langle \hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C), \mathbf{z} \rangle \le \langle \hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C), \bar{\mathbf{z}} \rangle$ holds for all $\mathbf{z} \in C_{-\varepsilon}$. As before, to simplify the notation we use $\mathbf{g} := \mathbf{g}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)$ and $\hat{\mathbf{g}} := \hat{\mathbf{g}}^{\text{sep}}_{\bar{\mathbf{z}}}(f, C)$. For every $\mathbf{z} \in [-R, R]^{n+d}$ we have

$$\langle \hat{\mathbf{g}}, \mathbf{z} \rangle = \langle \mathbf{g}, \mathbf{z} \rangle + \langle \hat{\mathbf{g}} - \mathbf{g}, \mathbf{z} \rangle \le \langle \mathbf{g}, \mathbf{z} \rangle + \|\hat{\mathbf{g}} - \mathbf{g}\|_\infty \cdot \|\mathbf{z}\|_1 \le \langle \mathbf{g}, \mathbf{z} \rangle + \varepsilon' R(n+d) = \langle \mathbf{g}, \mathbf{z} \rangle + \frac{\varepsilon}{2}. \tag{12}$$

Now we claim that for every point $\mathbf{z} \in C_{-\varepsilon}$ we have $\langle \mathbf{g}, \mathbf{z} \rangle \le \langle \mathbf{g}, \bar{\mathbf{z}} \rangle - \varepsilon$: since the inequality $\langle \mathbf{g}, \mathbf{x} \rangle \le \langle \mathbf{g}, \bar{\mathbf{z}} \rangle$ is valid for the ball $B(\mathbf{z}, \varepsilon) \subseteq C$, we have

$$\langle \mathbf{g}, \bar{\mathbf{z}} \rangle \ge \max_{\mathbf{w} \in B(0, \varepsilon)} \langle \mathbf{g}, \mathbf{z} + \mathbf{w} \rangle = \langle \mathbf{g}, \mathbf{z} \rangle + \varepsilon, \tag{13}$$

proving the claim. Finally, we claim that $\langle \mathbf{g}, \bar{\mathbf{z}} \rangle \le \langle \hat{\mathbf{g}}, \bar{\mathbf{z}} \rangle + \frac{\varepsilon}{2}$:

$$\langle \mathbf{g}, \bar{\mathbf{z}} \rangle - \langle \hat{\mathbf{g}}, \bar{\mathbf{z}} \rangle = \langle \mathbf{g} - \hat{\mathbf{g}}, \bar{\mathbf{z}} \rangle \le \|\mathbf{g} - \hat{\mathbf{g}}\|_\infty \cdot \|\bar{\mathbf{z}}\|_1 \le \varepsilon' R(n+d) = \frac{\varepsilon}{2}. \tag{14}$$

Combining inequalities (12)-(14) proves that the cut $\langle \hat{\mathbf{g}}, \mathbf{z} \rangle \le \langle \hat{\mathbf{g}}, \bar{\mathbf{z}} \rangle$ is valid for $C_{-\varepsilon}$.

To obtain the ε-approximate value comparison oracle, since f takes values in $[-U, U]$, it suffices to probe the sign plus $\log \frac{2U}{\varepsilon}$ bits of $f(\bar{\mathbf{z}})$ and $f(\bar{\mathbf{z}}')$ to approximate each of the values within $\pm\frac{\varepsilon}{2}$, in which case we can decide whether $f(\bar{\mathbf{z}}) \le f(\bar{\mathbf{z}}') + \varepsilon$ or not. This concludes the proof when using bit queries.

Obtaining approximate oracles using inner product sign queries. This proof largely follows the same steps as for the first part, except for the need for the following result, which may be of independent interest.

Lemma 30. *For any $\varepsilon \in (0,1)$ and any vector $\mathbf{g} \in \mathbb{R}^d$, using $O(d \log \frac{d}{\varepsilon})$ inner product sign queries one can obtain a unit-length vector $\hat{\mathbf{g}} \in \mathbb{R}^d$ such that $\big\|\hat{\mathbf{g}} - \frac{\mathbf{g}}{\|\mathbf{g}\|}\big\| \le \varepsilon$.*

Proof. We prove by induction on the dimension d that for every $\delta \in (0,2)$, with $d \log \frac{8}{\delta}$ inner product sign queries we can obtain a vector $\hat{\mathbf{g}}$ such that $\big\|\hat{\mathbf{g}} - \frac{\mathbf{g}}{\|\mathbf{g}\|}\big\| \le 2d\delta$; the lemma then follows by setting $\delta = \frac{\varepsilon}{2d}$.

Just one query suffices when $d = 1$, so consider the base case $d = 2$. Perform a binary search as follows: Start with the cone $K_0 = \mathbb{R}^2$, with corresponding angle 2π. In iteration t, we maintain a cone K_t containing $\mathbf{g}$ whose angle is half that of K_{t-1} as follows. For each iteration, find a line $\{\mathbf{x} : \langle \mathbf{a}, \mathbf{x}\rangle = 0\}$ that cuts K_t into two cones $K_t^L = K_t \cap \{\mathbf{x} : \langle \mathbf{a}, \mathbf{x}\rangle \le 0\}$ and $K_t^R = K_t \cap \{\mathbf{x} : \langle \mathbf{a}, \mathbf{x}\rangle \ge 0\}$ each with half the angle of K_t, i.e. bisecting K_t. Ask the query "Is $\langle \mathbf{a}, \mathbf{g}\rangle \ge 0$?", and if so set $K_{t+1} = K_t^R$, otherwise set to $K_{t+1} = K_t^L$, and repeat the procedure. By construction all the cones K_t contain $\mathbf{g}$, and after $\log \frac{8}{\delta}$ iterations we obtain a cone K with angle $\frac{\delta\pi}{4}$. Let $\hat{\mathbf{g}}$ be any vector in this cone with unit ℓ_2-norm. For any other $\mathbf{x} \in K$ also of unit norm, we have

$$\|\hat{\mathbf{g}} - \mathbf{x}\|_2^2 = 2 - 2\langle \hat{\mathbf{g}}, \mathbf{x}\rangle \le 2 - 2\cos(\delta\pi/4) \le (\delta\pi/4)^2 \le \delta^2,$$

where the second inequality uses that fact that $\cos(\theta) \ge 1 - \frac{\theta^2}{2}$ for all $\theta \in (0, \pi/2)$. So $\|\hat{\mathbf{g}} - \mathbf{x}\|_2 \le \delta$ for all unit-norm vectors in K, and in particular $\hat{\mathbf{g}}$ gives the desired approximation of $\frac{\mathbf{g}}{\|\mathbf{g}\|_2}$, proving the desired result when $d = 2$.

Now consider the general case $d > 2$. Consider any 2-dimensional subspace A of $\mathbb{R}^d$, and let Π_A denote the projection onto this subspace. Using the 2-dimensional case on the subspace A, we see that by using $\log \frac{8}{\delta}$ queries of the form "Is $\langle \mathbf{a}, \Pi_A \mathbf{g}\rangle \ge 0$?", we can obtain a unit length vector $\tilde{\mathbf{g}} \in A$ such that $\|\lambda_A \cdot \tilde{\mathbf{g}} - \Pi_A \mathbf{g}\| \le \delta \|\Pi_A \mathbf{g}\|$, where $\lambda_A := \|\Pi_A \mathbf{g}\|$. We note that since $\langle \mathbf{a}, \Pi_A \mathbf{g}\rangle = \langle \Pi_A^* \mathbf{a}, \mathbf{g}\rangle$, the required queries can be obtained by inner product sign queries (here Π_A^* denotes the adjoint linear operator for the projection operator Π_A, whose matrix representation is given by the transpose of the matrix representing the projection Π_A).

Now consider the $(d-1)$-dimensional subspace $B := \operatorname{span}\{\tilde{\mathbf{g}}, A^\perp\}$, and notice that $\operatorname{dist}(\mathbf{g}, B) \le \delta\|\mathbf{g}\|$: the vector $\mathbf{b} := \lambda_A \cdot \tilde{\mathbf{g}} + (\mathbf{g} - \Pi_A \mathbf{g})$ belongs to B and $\|\mathbf{g} - \mathbf{b}\| = \|\lambda_A \cdot \tilde{\mathbf{g}} - \Pi_A \mathbf{g}\| \le \delta \|\Pi_A \mathbf{g}\| \le \delta \|\mathbf{g}\|$. Since $\mathbf{g}$ is close to this subspace, we project it there and recurse on dimension. More precisely, consider the projection $\Pi_B \mathbf{g}$ of $\mathbf{g}$ onto B, and inductively obtain a vector $\hat{\mathbf{g}} \in B$ such that $\|\lambda_B \cdot \hat{\mathbf{g}} - \Pi_B \mathbf{g}\|_2 \le 2(d-1)\delta \cdot \|\Pi_B \mathbf{g}\|$ (letting $\lambda_B := \|\Pi_B \mathbf{g}\|$), by using additional $(d-1)\log \frac{8}{\delta}$ queries (for a total of $d \log \frac{8}{\delta}$ queries).

We claim that $\hat{\mathbf{g}}$ is the desired approximation of $\mathbf{g}$, namely $\|\frac{\mathbf{g}}{\|\mathbf{g}\|} - \hat{\mathbf{g}}\|_2 \le 2d\delta$. To see this, from triangle inequality we have

$$\big\|\mathbf{g} - \|\mathbf{g}\| \cdot \hat{\mathbf{g}}\big\| \le \|\mathbf{g} - \Pi_B \mathbf{g}\| + \|\Pi_B \mathbf{g} - \lambda_B \cdot \hat{\mathbf{g}}\| + \big\|\lambda_B \cdot \hat{\mathbf{g}} - \|\mathbf{g}\| \cdot \hat{\mathbf{g}}\big\|. \tag{15}$$

The first term of the right-hand side equals $\operatorname{dist}(\mathbf{g}, B)$, which is at most $\delta\|\mathbf{g}\|$ as argued above. For the second term, by induction we have

$$\|\Pi_B \mathbf{g} - \lambda_B \cdot \hat{\mathbf{g}}\|_2 \le 2(d-1)\delta \cdot \|\Pi_B \mathbf{g}\| \le 2(d-1)\delta\|\mathbf{g}\|.$$

Finally, we claim that the last term of (15) is at most $\delta\|\mathbf{g}\|$: since $\hat{\mathbf{g}}$ has unit norm, it equals $|\lambda_B - \|\mathbf{g}\|| = |\|\Pi_B\mathbf{g}\| - \|\mathbf{g}\|| = \|\mathbf{g}\| - \|\Pi_B\mathbf{g}\|$, and by triangle inequality we have

$$\|\mathbf{g}\| \leq \|\Pi_B\mathbf{g}\| + \|\mathbf{g} - \Pi_B\mathbf{g}\| \leq \|\Pi_B\mathbf{g}\| + \delta\|\mathbf{g}\|,$$

giving the claim.

Applying all these bounds to (15), we get that $\big\|\mathbf{g} - \|\mathbf{g}\| \cdot \hat{\mathbf{g}}\big\| \leq 2d\delta\|\mathbf{g}\|$, as desired. This concludes the proof of the lemma. $\square$

Now we are ready to finish the proof. To obtain an ε-approximate value comparison oracle using $\mathcal{H}^{\mathrm{dir}}$, we can do a binary search on the function values using the queries $h^{\mathrm{val}}_{u,c}$ with $u = 1$ and different values of c (as the midpoint of the interval in the binary search) – see Definition 5. Thus, with $O(\log \frac{U}{\varepsilon})$ queries, we can implement an ε-approximate value comparison oracle.

For the ε-approximate separation, we apply Lemma 30 above to the separation oracle $\mathbf{g}^{\mathrm{sep}}_{\bar{\mathbf{z}}}(f, C)$, with $O\big((n+d)\log\frac{(n+d)R}{\varepsilon}\big)$ inner product sign queries to obtain a vector $\hat{\mathbf{g}}^{\mathrm{sep}}_{\bar{\mathbf{z}}}(f, C)$ such that $\|\mathbf{g}^{\mathrm{sep}}_{\bar{\mathbf{z}}}(f, C) - \hat{\mathbf{g}}^{\mathrm{sep}}_{\bar{\mathbf{z}}}(f, C)\| \leq \frac{\varepsilon}{2R\sqrt{n+d}}$ (recall the non-zero separation oracles are assumed to have unit length). Using the same arguments as in inequalities (12)-(14), we see that $\hat{\mathbf{g}}^{\mathrm{sep}}_{\bar{\mathbf{z}}}(f, C)$ gives a cut valid for $C_{-\varepsilon}$, and hence is an ε-approximate separation oracle.

For the ε-approximate value cut oracle, we do the same thing, but apply Lemma 30 to $\mathbf{g}^{\mathrm{sub}}_{\bar{\mathbf{z}}}(f, C)$, with $O\left((n+d)\log\left(\frac{(n+d)MR}{\varepsilon}\right)\right)$ oracle calls to obtain an approximation $\|\mathbf{g}^{\mathrm{sub}}_{\bar{\mathbf{z}}}(f, C) - \hat{\mathbf{g}}^{\mathrm{sub}}_{\bar{\mathbf{z}}}(f, C)\| \leq \frac{\varepsilon}{2MR\sqrt{n+d}}$ and then use the argument from (11).

Remark 31. *Notice that the main ingredient for implementing approximate oracles using inner product sign queries is to use such queries for approximating a given vector* $\mathbf{g}$ *(Lemma 30). We remark that this task is related to (actively) learning a linear classifier, namely that whose normal is given by* $\mathbf{g}$*. While there are existing procedures for doing this, they only guarantee the desired approximation with high probability (albeit on a slightly weaker query model), instead of with probability 1 as we want here; see for example [1]. While it is possible that these methods can be adapted to our setting, we present a different, self-contained statement and proof.*

5 Proofs of Theorem 12 and Corollary 13

Proof of Theorem 12. Suppose we have an algorithm $\mathcal{A}$ that reports an ε-solution to any instance in $\mathcal{I}_{n,d,R,\rho,M}$ after u queries to a full-information first-order oracle based on the first-order chart $\mathcal{G}$, a finite set of instances $\mathcal{I} \subseteq \mathcal{I}_{n,d,R,\rho,M}$, and a true (unknown) instance $I \in \mathcal{I}$. Our goal is to report a feasible ε-solution using few queries to $\mathcal{O}(\mathcal{G}, \mathcal{H})$, where $\mathcal{H}$ contains all binary queries. For this, we design a procedure that maintains a family $\mathcal{U} \subseteq \mathcal{I}$ of the instances, which always includes the true instance I, and possibly determines exact information to pass to $\mathcal{A}$. We will show that we can always either reduce $|\mathcal{U}|$ by a constant factor, or determine exact information to use with $\mathcal{A}$.

Denote by D the query strategy of $\mathcal{A}$. Initialize $\mathcal{U} = \mathcal{I}$, and (ordered) lists $Q = \emptyset, H = \emptyset$, which will serve as query-response pairs for the algorithm $\mathcal{A}$. In particular, H will contain full first-order information about the true instance, and Q will be the sequence of queries $\mathcal{A}$ makes. While $|\mathcal{U}| > 1$ and $|Q| \leq u$, do the following:

- Set $\mathbf{z} = D(Q, H)$, and query whether $\mathbf{z}$ is feasible. Let us simply write $\mathbf{g}_{\mathbf{z}}(I)$ to mean $\mathbf{g}^{\mathrm{sep}}_{\mathbf{z}}(I)$ if $\mathbf{z}$ is infeasible for the instance I, and $\mathbf{g}^{\mathrm{val}}_{\mathbf{z}}(I)$ or $\mathbf{g}^{\mathrm{sub}}_{\mathbf{z}}(I)$ if feasible, where $\mathbf{g}^{\mathrm{sep}}_{\mathbf{z}}$, $\mathbf{g}^{\mathrm{val}}_{\mathbf{z}}$ and $\mathbf{g}^{\mathrm{sub}}_{\mathbf{z}}$ are the first-order maps for separating hyperplane, function value and subgradient, respectively, used by the general binary oracle at $\mathbf{z}$. Write $V_{\mathbf{z}}$ to be the appropriate codomain.

- **Case 1:** For every $\mathbf{v} \in V_{\mathbf{z}}$, at most half of the instances $I' \in \mathcal{U}$ give $\mathbf{g}_{\mathbf{z}}(I') = \mathbf{v}$. Then there exists a set $A \subseteq V_{\mathbf{z}}$ such that the number of instances $I' \in \mathcal{U}$ with $\mathbf{g}_{\mathbf{z}}(I') \in A$ is between $\frac{1}{4}|\mathcal{U}|$ and $\frac{3}{4}|\mathcal{U}|$. Let $\mathcal{U}_0 := \{I \in \mathcal{U} : \mathbf{g}_{\mathbf{z}}(I') \notin A\}$, $\mathcal{U}_1 := \{I \in \mathcal{U} : \mathbf{g}_{\mathbf{z}}(I') \in A\}$; thus, $|\mathcal{U}_i| \leq \frac{3}{4}|\mathcal{U}|$ for $i = 0, 1$. Query whether the true instance I has $\mathbf{g}_{\mathbf{z}}(I) \in A$, using the binary query $h_A : h_A(\mathbf{v}) = 1$ iff $\mathbf{v} \in A$, so that $h_A(\mathbf{g}_{\mathbf{z}}(I)) = 0$ if $I \in \mathcal{U}_0$ and $h_A(\mathbf{g}_{\mathbf{z}}(I)) = 1$ if $I \in \mathcal{U}_1$. Update $\mathcal{U} \leftarrow \mathcal{U}_q$, where q is the answer to the query $(\mathbf{z}, h_A)$ given by the oracle.

- **Case 2:** There exists $\bar{\mathbf{v}} \in V_{\mathbf{z}}$ such that more than half of the instances $I' \in \mathcal{U}$ have $\mathbf{g}_{\mathbf{z}}(I') = \bar{\mathbf{v}}$. Query whether the true instance I has $\mathbf{g}_{\mathbf{z}}(I) = \bar{\mathbf{v}}$, using the binary query $h : h(\bar{\mathbf{v}}) = 1$ and $h(\mathbf{x}) = 0$ for all other inputs $\mathbf{x} \neq \bar{\mathbf{v}}$, so that $h(\mathbf{g}_{\mathbf{z}}(I)) = 1$ iff $\mathbf{g}_{\mathbf{z}}(I) = \bar{\mathbf{v}}$. If $\mathbf{g}_{\mathbf{z}}(I) \neq \bar{\mathbf{v}}$, then update $\mathcal{U}$ by removing from it all instances I' such that $\mathbf{g}_{\mathbf{z}}(I') = \bar{\mathbf{v}}$, reducing the size of $\mathcal{U}$ by at least half. Otherwise, if $\mathbf{z}$ was infeasible, we then know the exact separating hyperplane (or function value or subgradient, in the case $\mathbf{z}$ is feasible) for the true instance I and the first-order chart $\mathcal{G}$, namely $\mathbf{g}_{\mathbf{z}}^{\text{sep}}(I) = \bar{\mathbf{v}}$, and so employ it to update Q and H by appending $\mathbf{z}$ and $\mathbf{v}$ to them, respectively, which will serve as information for the algorithm $\mathcal{A}$.

In each step, either the size of $\mathcal{U}$ decreases by at least 1/4, or full (exact) first-order information at the query point determined by the query strategy of $\mathcal{A}$ is obtained and Q, H are updated. The former can only happen $O(\log|\mathcal{I}|)$ times until $\mathcal{U}$ becomes a singleton, in which case we know the true instance and can report its optimal solution, while if the latter happens u times, one can run the algorithm $\mathcal{A}$ with the information (Q, H) to report an ε-approximate solution to the true instance I, noting that since the points in Q were determined according to the query strategy of the algorithm, the information in (Q, H) is indeed sufficient to run the $\mathcal{A}$ on for u iterations. Hence, after at most $\log|\mathcal{I}| + u$ queries to the general binary oracle, one can report an ε-solution to the true instance. □

Corollary 13 follows immediately when one uses the centerpoint-based algorithm of [13, 3] as $\mathcal{A}$, which is the exact oracle version of Algorithm 1 above and needs at most

$$O\left(2^n\, d\,(n+d) \log\left(\frac{dMR}{\min\{\rho, 1\}\varepsilon}\right)\right)$$

queries in the mixed-integer case, or

$$O\left(d \log\left(\frac{MR}{\min\{\rho, 1\}\varepsilon}\right)\right)$$

queries in the continuous ($n = 0$) case to produce an ε-approximate solution to any instance in $\mathcal{I}_{n,d,\rho,M,R}$.

6 Statements and Declarations

Competing Interests. There are no financial or non-financial interests that are directly or indirectly related to this work.

References

[1] Maria-Florina Balcan and Phil Long. Active and passive learning of linear separators under log-concave distributions. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 288–316, Princeton, NJ, USA, 12–14 Jun 2013. PMLR.

[2] Amitabh Basu. Complexity of optimizing over the integers. *Mathematical Programming, Series B*, 200:739–780, 2023.

[3] Amitabh Basu and Timm Oertel. Centerpoints: A link between optimization and convex geometry. *SIAM Journal on Optimization*, 27(2):866–889, 2017.

[4] A Yu Chirkov, Dmitry V Gribanov, Dmitriy S Malyshev, Panos M Pardalos, Sergey I Veselov, and N Yu Zolotykh. On the complexity of quasiconvex integer minimization problem. *Journal of Global Optimization*, 73:761–788, 2019.

[5] Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization.* SIAM, 2009.

[6] Dmitriy V Gribanov and Dmitriy S Malyshev. Integer conic function minimization based on the comparison oracle. In *International Conference on Mathematical Optimization Theory and Operations Research*, pages 218–231. Springer, 2019.

[7] Dmitriy Vladimirovich Gribanov and Dmitriy Sergeevich Malyshev. Minimization of even conic functions on the two-dimensional integral lattice. *Journal of Applied and Industrial Mathematics*, 14:56–72, 2020.

[8] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics: Study and Research Texts.* Springer-Verlag, Berlin, 1988.

[9] Branko Grünbaum. Partitions of mass-distributions and of convex bodies by hyperplanes. *Pacific J. Math.*, 10:1257–1261, 1960.

[10] Kevin G. Jamieson, Robert D. Nowak, and Benjamin Recht. Query complexity of derivative-free optimization. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, page 2672–2680, Red Hook, NY, USA, 2012. Curran Associates Inc.

[11] Annie Marsden, Vatsal Sharan, Aaron Sidford, and Gregory Valiant. Efficient convex optimization requires superlinear memory. *arXiv preprint arXiv:2203.15260*, 2022.

[12] Arkadii S. Nemirovski and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization.* John Wiley, 1983.

[13] Timm Oertel. *Integer Convex Minimization in Low Dimensions.* PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 22288, 2014.

[14] V. Yu. Protasov. Algorithms for approximate calculation of the minimum of a convex function from its values. *Mathematical Notes*, 59(1):69–74, Jan 1996.

[15] Sergey I Veselov, Dmitry V Gribanov, N Yu Zolotykh, and A Yu Chirkov. A polynomial algorithm for minimizing discrete convic functions in fixed dimension. *Discrete Applied Mathematics*, 283:11–19, 2020.

[16] Blake Woodworth and Nathan Srebro. Open problem: The oracle complexity of convex optimization with limited memory. In *Conference on Learning Theory*, pages 3202–3210. PMLR, 2019.

A Information Games

In this section we define the game-theoretic perspective for information complexity and prove Lemma 15, which we restate for convenience.

Lemma 15. *Consider a class of instances $\mathcal{I}$ and an oracle $\mathcal{O}$. Then* $\operatorname{icomp}_\varepsilon(\mathcal{I}, \mathcal{O}) > \ell$ *if and only if there exists an adversary under $\mathcal{O}$ using $\mathcal{I}$ that is ε-hard for ℓ rounds.*

Let $\mathcal{I}$ be a family of optimization instances in $\mathbb{R}^n \times \mathbb{R}^d$. Let $\mathcal{O}$ be an oracle for $\mathcal{I}$. Let $\varepsilon > 0$. We define the *information game for* $(\mathcal{I}, \mathcal{O}, \varepsilon)$ as a two player game between Alg and Adv. The players make moves alternatingly with Alg moving first. For Alg, a move is a choice of query q from the oracle $\mathcal{O}$. For Adv, a move is a choice of $r \in H$, where H is the response set of q from the immediately preceding move by Alg. The only constraint on Adv is that there should exist at least one instance in $\mathcal{I}$ that will give the same responses as the moves made by Adv in the game so far to queries corresponding to the moves made by Alg so far. More precisely, at any stage of the game, there must exist an instance I such that for every response r given by Adv at any round of the game played so far to an immediately preceding query q made by Alg, we have $r = q(I)$.

Notice that a game strategy for Alg is equivalent to a query strategy from Definition 2. Also, any instance $\bar{I} \in \mathcal{I}$ gives a game strategy for Adv, by simply reporting the response $q(\bar{I})$ for all queries q made by Alg.

In the game tree defined by the above game, a node is said to be *ε-unambiguous* if the instances that are consistent with Adv's moves all have a common ε-approximate solution; otherwise, the node is *ε-ambiguous.* The game stops when the game reaches an ε-unambiguous node; thus, all leaf nodes of the game tree are ε-unambiguous. When the game stops at an ε-unambigious node L, the *loss (or payoff)* $\mathcal{L}_\varepsilon(L)$ is defined as the number of moves made by Alg to arrive at L in the game tree, i.e., it is exactly half of the depth of L in the game tree.

A subtree T of the game tree is said to be a *full subgame tree* if it is precisely the set of all descendants of a node in the game tree (including itself). A subtree T is said to have *finite horizon* if it has bounded depth, i.e., there exists $D > 0$ such that all nodes of T have depth (in T) bounded by D. Otherwise, T is said to have *infinite horizon* (note this includes the case where T has an infinite length path and the case where all paths are finite length, but there is no upper bound on the lengths).

We define the value $v(T)$ of a finite horizon subtree T by induction on the depth of T, which we call the *value of the subgame defined by* T.

- If T consists of a single node N, $v(T) := \mathcal{L}_\varepsilon(N)$ if N is an ε-unambiguous node, else $v(T) := \infty$.
- If T is rooted at a node N corresponding to Alg, then $v(T) := \inf\{v(T') : T' \text{ child subtree in } T \text{ at } N\}$.
- If T is rooted at a node N corresponding to Adv, then $v(T) := \sup\{v(T') : T' \text{ child subtree in } T \text{ at } N\}$.

Let T and T' be subtrees of the game tree. T' is said to be a *(finite horizon) truncation of* T if there exists $D > 0$ such that T' consists of all nodes of T with depth bounded by D. Observe that if T, T' are finite horizon subtrees and T' is a truncation of T, then $v(T') \geq v(T)$. We can thus naturally extend the notion of value to an infinite horizon subtree T, as a limit of its finite horizon truncations:

$$v(T) := \inf\{v(T') : T' \text{ finite horizon truncation of } T\}.$$

A *game strategy* for Alg (resp. Adv) is a choice of a specific move at every non leaf node corresponding to Alg (resp. Adv). Thus, any game strategy Q for Alg (resp. game strategy

A for Adv) corresponds to a subtree of the entire game tree, where we select a single outgoing edge at every node corresponding to Alg (resp. Adv). Such a subtree is called a *decision tree for* Alg (resp. Adv). With a slight abuse of notation, we will use $v(Q)$ (resp. $v(A)$) to denote the values of these decision trees. A simultaneous choice of strategies Q, A yields a single path in the original game tree; $v(Q, A)$ will denote the value of this path (subtree).

We note that we can express the notion of an ε-hard adversary for ℓ using this notation; it follows directly from the definition of the value function $v(\cdot)$.

Observation 32. *An adversary A is ε-hard for ℓ-rounds iff $\inf_Q v(Q, A) > \ell$, where the infimum is taken over all strategies for* Alg.

We now focus on proving Lemma 15, which requires a few preparatory observations. The first is the following, which is a useful extension of the recursive nature of the value of a subtree to infinite horizon trees.

Lemma 33. *Let T be any subtree of the game tree (not necessarily of finite horizon). Then,*

- *If T consists of a single node N, $v(T) = \mathcal{L}_\varepsilon(N)$ if N is an ε-unambiguous node, else $v(N) = \infty$.*
- *If T is rooted at a node N corresponding to* Alg, *then $v(T) = \inf\{v(T') : T'$ child subtree in T at $N\}$.*
- *If T is rooted at a node N corresponding to* Adv, *then $v(T) = \sup\{v(T') : T'$ child subtree in T at $N\}$.*

Proof. The result follows from the fact that for any finite horizon truncation T' of T, the subtrees of T' rooted at the children of the root of T' (and T) are finite horizon truncations of the subtrees of T rooted at the children of the root. □

The following technical property will be used several times in the sequel.

Lemma 34. *The value of the full game is finite if and only if every full subgame has finite value.*

Proof. Since the full game is a subgame of itself, if every subgame has finite value so does the full game. For the other direction, suppose now that the full game tree has finite value. Then, there is a finite horizon truncation T with finite value. The next claim shows that for every full subgame tree, its value can be upper bounded using the value of T, so in particular is finite as desired.

Claim 35. *Let S be a full subgame tree, and let N denote its root node. Then $v(S) \leq v(T) + depth(N) + 1$.*

Proof. Let D be the depth of T. Consider the truncation S' of S which has depth D if N is a node corresponding to Alg, and depth $D + 1$ if N is a node corresponding to Adv. Since S' is a truncation of S, it follows that $v(S) \leq v(S')$, and so we only need to show $v(S') \leq v(T) + depth(N) + 1$.

To prove this bound, consider first the case when N is a node corresponding to Alg. In this case, T is isomorphic to a subtree of S', i.e., each node of T corresponds to a node in S' in the natural way: the root of T corresponds to the root of S', the children of the root of T corresponds to (some of) the children of the root of S', etc. Observe that the nodes corresponding to Alg have the same possibilities for moves in both trees S' and T (in fact, every node of Alg in the full game tree has the same choice of moves by definition). However, nodes corresponding to Adv have a smaller set of choices in S', i.e., children, compared to T, because they are deeper in the game tree. Thus, in the inductive definition of $v(S')$, a supremum computed at a node in S' for Adv is over a smaller set of choices compared to the corresponding supremum when computing $v(T)$. Finally, the depth (in the full game tree) of any node in S is precisely $\mathit{depth}(N)$ more than the depth of the corresponding node in T. Thus, $v(S') \leq v(T) + \mathit{depth}(N) \leq v(T) + \mathit{depth}(N) + 1$.

Next, consider the case when N is a node corresponding to Adv. All the subtrees of S' rooted at the children of N are again in correspondence with T, since S is of depth $D+1$. The previous argument applies to these subtrees and thus, their values are all bounded by $v(T)+\mathit{depth}(N)+1$. Taking a supremum at node N shows that $v(S') \leq v(T) + \mathit{depth}(N) + 1$. □

□

We first prove the existence of a saddle-point in this game (in fact, the same argument gives that there is a subgame perfect equilibrium, which essentially means that this saddle point property holds for every subtree).

Lemma 36. *Suppose the value of the full game is finite (in other words, the optimization problem is solvable with finitely many queries using the given oracles). Then, there exists a saddle-point $Q^\star, A^\star$ for* Alg *and* Adv *respectively, i.e., strategies $Q^\star$ and $A^\star$ satisfying*

$$\inf_Q v(Q, A^\star) = v(Q^\star, A^\star) = \sup_A\, v(Q^\star, A) = v(\text{full game}), \tag{16}$$

where full game *denotes the full game tree, the infimum on the left-hand side is over all possible strategies for* Alg*, and the supremum on the right-hand side is over all strategies for* Adv*.*

Proof. Since we assume the full game has finite value, by Lemma 34 we know that the value $v(S)$ of any full subgame tree S is also finite. Combined with the fact that $v(S)$ is integer valued, the supremum or infimum in Lemma 33 is attained. A saddle-point $Q^\star$ and $A^\star$ can now be defined as follows. At any node N corresponding to a move by Alg, select the move that attains the infimum giving the value of the full subgame tree rooted at N by Lemma 33. At any node N corresponding to a move by Adv, select the move that attains the supremum giving the value of the full subgame tree rooted at N by Lemma 33.

We now prove that $Q^\star, A^\star$ satisfy the desired properties. When the full tree is finite, this follows directly from the definition of $Q^\star, A^\star$; the case where it is not finite is that requires more subtle arguments based on finite horizon truncations. Let T denote the full game tree, and for a node u let T_u denote its full subtree rooted at u.

We start by proving that $v(Q^\star, A^\star) = v(T)$. Let $p_1, p_2, \ldots$ be the nodes in the path $P^\star$ induced by $Q^\star, A^\star$ (p_1 being the root of the full game tree). By the second item of Lemma 33, we have that $v(T_{p_1})$ is the smallest value of the subtrees rooted at the children of p_1; since the strategy $Q^\star$ chooses precisely such child p_2 of smallest value, we get $v(T_{p_1}) = v(T_{p_2})$. Similarly, using the same lemma and the definition of the adversarial strategy $A^\star$, we get $v(T_{p_2}) = v(T_{p_3})$. Thus, subtrees T_{p_i}'s have the same value, equal to $v(T_{p_1}) = v(T)$ (which is finite). Moreover, the value of a subtree T_u is at least half of the depth of u: $v(T_u)$ is either ∞ or is the (finite) payoff of a node w in T_u, which is at least half the depth of w, and so at least half the depth of u. Therefore, all nodes in the path $P^\star$ are at depth at most $2 \cdot v(T)$; in particular, $P^\star$ has finite length. Now consider a finite horizon truncation T' of T with value $v(T') = v(T)$; we can further increase the depth of this truncation and assume that the path $P^\star$ is contained in T'. Since T' is finite, its value $v(T')$ is defined by taking the child that has subtree of smallest/largest value on the nodes of Alg/Adv, until a leaf (of finite value) is reached. But this is by construction the choices that the path $P^\star$ makes, namely they reach the same leaf (or leaves of the same value). The values of $v(T')$ and $v(P^\star)$ are then the value of this leaf/leaves, and hence $v(T') = v(P^\star)$, or equivalently $v(Q^\star, A^\star) = v(T)$ as desired.

We now consider the first desired equation, namely $\inf_Q v(Q, A^\star) = v(Q^\star, A^\star)$, and notice that it suffices to show

$$\inf_Q v(Q, A^\star) \geq v(Q^\star, A^\star) \tag{17}$$

(the equality follows by taking $Q = Q^\star$). Assume that the paths induced by $Q, A^\star$ and $Q^\star, A^\star$ are different, else there is nothing to show. Let u be the first node of Alg in these paths where Q and $Q^\star$ make different decisions; let w and $w^\star$ be the children of u on the paths induced by $Q, A^\star$ and $Q^\star, A^\star$ respectively. Notice that by construction of the choice $Q^\star$ makes at u, we have $v(T_{w^\star}) \leq v(T_w)$.

Now we claim that $v(Q, A^\star) \geq v(T_w)$. If the path P induced by $Q, A^\star$ has infinite length, then all of its nodes are ε-ambiguous and so by definition $v(Q, A^\star) = \infty$ and the inequality holds. So suppose P has finite length. Since the value $v(T_w)$ is finite, again due to our assumption that $v(T)$ is finite, there is a finite horizon truncation T' of this subtree with the same value; again, we can further increase the depth of this truncation and assume that the path $P \cap T_w$ (the suffix of P starting at w) is contained in T' (and hence $P \cap T_w = P \cap T'$). By definition, the nodes of Adv in the path $P \cap T'$ always select the action that lead to the child whose subtree has largest value (being based on $A^\star$), while the nodes of Alg may not necessarily select the children with lowest value subtrees (being based on Q). Since in the definition of the value $v(T')$, both the nodes of Adv and Alg make the optimal decisions, we see that $v(T') \leq v(P \cap T')$. Since the value of the path $v(P)$ and of its suffix $v(P \cap T')$ are the same (they are the value of the leaf of the path), we obtain $v(T_w) = v(T') \leq v(P \cap T') = v(P) = v(Q, A^\star)$, as claimed.

Next, we claim that $v(T_{w^\star}) = v(Q^\star, A^\star)$. As proved above, the path $P^\star$ induced by $Q^\star, A^\star$ has finite length. Since the value $v(T_{w^\star})$ is finite, as above, consider a truncation T' of $T_{w^\star}$ with same value $v(T') = v(T_{w^\star})$ and that contains the suffix $P^\star \cap T_{w^\star}$. By the same argument as in the previous paragraph, but now using that both $Q^\star$ and $A^\star$ make optimal decisions, we have $v(T') = v(P^\star \cap T_{w^\star})$, which is also equal to $v(P^\star)$; this implies $v(T_{w^\star}) = v(P^\star) = v(Q^\star, A^\star)$ as desired.

Putting together the bounds from the three previous paragraphs yields $v(Q^\star, A^\star) = v(T_{w^\star}) \leq v(T_w) \leq v(Q, A^\star)$ for every strategy Q for the algorithm. Taking an infimum over all such Q's finally proves (17), and hence that $\inf_Q v(Q, A^\star) = v(Q^\star, A^\star)$.

The proof that $\sup_A v(Q^\star, A) = v(Q^\star, A^\star)$ uses the exact same arguments, but exchanging the roles of the players Alg and Adv. □

By standard arguments, the existence of a saddle-point implies a minimax result, which is the first element required to prove Lemma 15.

Lemma 37. *Suppose the value of the full game is finite, and let $Q^\star, A^\star$ be a subgame perfect equilibrium guaranteed to exist by Lemma 36. The the following minimax holds;*

$$\sup_A \inf_Q \; v(Q, A) = v(Q^\star, A^\star) = \inf_Q \; \sup_A \; v(Q, A),$$

where the infima are over all possible strategies for Alg*, and the suprema are over all strategies for* Adv*.*

Proof. From the guarantees of $Q^\star, A^\star$, for all strategies Q, A for Alg and Adv respectively, we have

$$v(Q^\star, A^\star) \leq \inf_Q v(Q, A^\star) \leq \sup_A \inf_Q v(Q, A)$$

and

$$\inf_Q \sup_A v(Q, A) \leq \sup_A v(Q^\star, A) \leq v(Q^\star, A^\star).$$

Since we always have $\sup_A \inf_Q v(Q, A) \leq \inf_Q \sup_A v(Q, A)$, we get equalities throughout. $\square$

The last element required for proving Lemma 15 is the equivalence between the value of this game and the information complexity of the underlying optimization problem.

Lemma 38. *The value of the full game* $v(\text{full game})$ *equals* $\text{icomp}_\varepsilon(\mathcal{I}, \mathcal{O})$*.*

Proof. Combining Lemmas 36 and 37, $v(\text{full game}) = \inf_Q \sup_A v(Q, A)$, so it suffices to show

$$\inf_Q \sup_A v(Q, A) = \inf_Q \sup_{I \in \mathcal{I}} \text{icomp}_{\,\varepsilon}(Q, I, \mathcal{O}).$$

To see the "$\geq$" direction: Consider any strategy Q for Alg and instance $I \in \mathcal{I}$. Consider the adversary A based on the instance I, namely that reports $q(I)$ whenever Q asks a query q. Notice that $v(Q, A) = \text{icomp}_\varepsilon(Q, I, \mathcal{O})$ (if $\text{icomp}_\varepsilon(Q, I, \mathcal{O})$ is finite, after exactly $\text{icomp}_\varepsilon(Q, I, \mathcal{O})$ many moves of the algorithm Q we reach an ε-unambiguous node in the path induced by Q, A, so $v(Q, A) = \text{icomp}_\varepsilon(Q, I, \mathcal{O})$; if $\text{icomp}_\varepsilon(Q, I, \mathcal{O})$ is infinite, every finite horizon truncation of the path induced by Q, A must end on an ε-ambiguous node, so $v(Q, A)$ is also infinite). That is, for every I we can find a strategy A for the adversary with the "same value", which then implies $\sup_A v(Q, A) \geq \sup_{I \in \mathcal{I}} \text{icomp}_\varepsilon(Q, I, \mathcal{O})$. Taking an infimum over all Q's on both sides gives the desired inequality $\inf_Q \sup_A v(Q, A) \geq \inf_Q \sup_{i \in \mathcal{I}} \text{icomp}_\varepsilon(Q, I, \mathcal{O})$.

The proof for the "$\leq$" direction is analogous: Consider any strategy Q for Alg and strategy A for Adv. If $v(Q, A) = \infty$, then for every finite horizon truncation of length N (i.e. with $\frac{N}{2}$ queries by the algorithm) of the path induced by Q, A ends on an ε-ambiguous node; let $I \in \mathcal{I}$ be an instance consistent with A given up to this node. Then we see that $\text{icomp}_\varepsilon(Q, I, \mathcal{O}) \geq \frac{N}{2}$. Since this holds for every even number N, we get $\sup_{I \in \mathcal{I}} \text{icomp}_\varepsilon(Q, I, \mathcal{O}) = \infty$. Similarly, if $v(Q, A)$ is finite, then there is a node L in the path induced by Q, A with $\mathcal{L}_\varepsilon(L) = v(Q, A)$; any instance I associated to this node L (i.e., compatible with the answers given by A until reaching L) then has $\text{icomp}_\varepsilon(Q, I, \mathcal{O}) \geq \mathcal{L}_\varepsilon(L) = v(Q, A)$. These observations imply $\sup_{I \in \mathcal{I}} \text{icomp}_\varepsilon(Q, I, \mathcal{O}) \geq \sup_A v(Q, A)$. Taking an infimum over Q gives $\inf_Q \sup_{I \in \mathcal{I}} \text{icomp}_\varepsilon(Q, I, \mathcal{O}) \geq \inf_Q \sup_A v(Q, A)$ as desired.

This concludes the proof of the lemma. $\square$

Now we can finally prove Lemma 15, stated in the beginning of the section.

Proof of Lemma 15. From Observation 32, there is an adversary A that is ε-hard for ℓ-rounds iff $\sup_A \inf_Q v(Q, A) > \ell$. Combining Lemmas 36, 37, and 38, the latter is equivalent to $\text{icomp}_\varepsilon(\mathcal{I}, \mathcal{O}) > \ell$, which concludes the proof. $\square$