Solving Constrained Variational Inequalities via an Interior Point Method

Tong Yang*
University of California, Berkeley
pptmiao@berkeley.edu

Michael I. Jordan*
University of California, Berkeley
jordan@cs.berkeley.edu

Tatjana Chavdarova*
University of California, Berkeley
tatjana.chavdarova@berkeley.edu

Abstract

We develop an interior-point approach to solve constrained variational inequality (cVI) problems. Inspired by the efficacy of the alternating direction method of multipliers (ADMM) method in the single-objective context, we generalize ADMM to derive a first-order method for cVIs, that we refer to as ADMM-based interior point method for constrained VIs (ACVI). We provide convergence guarantees for ACVI in two general classes of problems: (i) when the operator is $\xi$-monotone, and (ii) when it is monotone, the constraints are active and the game is not purely rotational. When the operator is in addition L-Lipschitz for the latter case, we match known lower bounds on rates for the gap function of $O(1/\sqrt{K})$ and $O(1/K)$ for the last and average iterate, respectively. To the best of our knowledge, this is the first presentation of a first-order interior-point method for the general cVI problem that has a global convergence guarantee. Moreover, unlike previous work in this setting, ACVI provides a means to solve cVIs when the constraints are nontrivial. Empirical analyses demonstrate clear advantages of ACVI over common first-order methods. In particular, (i) cyclical behavior is notably reduced as our methods approach the solution from the analytic center, and (ii) unlike projection-based methods that oscillate when near a constraint, ACVI efficiently handles the constraints.

1 Introduction

We are interested in the general constrained variational inequality problem [Stampacchia, 1964]:

\[
\begin{align*}
\text{find } x^* \in \mathcal{X} \quad \text{s.t.} \quad \langle x - x^*, F(x^*) \rangle &\geq 0, \quad \forall x \in \mathcal{X},
\end{align*}
\]

(cVI)

where $F : \mathcal{X} \to \mathbb{R}^n$ is continuous map, and $\mathcal{X}$ is a subset of the Euclidean $n$-dimensional space $\mathbb{R}^n$—defined by the constrains or $\mathcal{X} = \mathbb{R}^n$ for the unconstrained case; see § 3. We denote the set of solutions of the cVI by SOL($\mathcal{X}, F$). Finding such solutions is fundamental in many fields—including optimization, economics, and machine learning [see, e.g. Goodfellow et al., 2014, Omidshafiei et al., 2017]; indeed, the range of problems that can be formalized as cVIs is vast—cVIs generalize standard single-objective optimization, the broad class of complementarity problems [Cottle and Dantzig, 1968], zero-sum games [von Neumann and Morgenstern, 1947, Rockafellar, 1970] and multi-player games, as well as many other problems where solution concepts are expressed as equilibria. As an example, consider two agents $x_1 \in \mathcal{X}_1$, and $x_2 \in \mathcal{X}_2$ that share a loss/utility function, $f : \mathcal{X}_1 \times \mathcal{X}_2 \to \ldots$

* All authors contributed equally.
This corresponds to a cVI in which the development of efficient algorithms with provable convergence has recently been the focus of (IP) methods is the de facto family of iterative algorithms for constrained optimization. Interior point methods are the de facto family of iterative algorithms for constrained optimization.

As a result, we pose the following question. First-order methods for cVI that have global convergence guarantee, and that can be applied when general inequality and equality constraints are given, and that have global convergence guarantees for a broad class of cVI problems?
In this paper, due to the immense application of IP methods in constrained convex optimization, we address the above question via an IP approach. Inspired by the work of Lin et al. [2018], we replace the Newton method with the so-called modified alternating direction method of multipliers (ADMM) method, described in § 3.1. By generalizing ADMM’s technique, we derive a novel algorithm for solving monotone VIs with very general constraints. Our contributions can be summarized as follows:

• Primarily, based on the KKT system for the constrained VI problem and the ADMM technique described in § 3.1, we derive an algorithm named ADMM-based interior point method for constrained VIs (ACVI)—see § 4.1 and Algorithm 1.

• We then prove the global convergence of ACVI on two sets of assumptions: (i) when $F$ is $\xi$-monotone—see Def. 1 and (ii) when it is monotone, the constraints are active at the solution, and the game is not purely rotational. By further assuming $F$ is a Lipschitz operator, we upper bound the rate of decreasing the gap function (see Def. 2) and we reach the known lower bound rate for the gap function of $O(1/\sqrt{K})$ for the last iterate—see § 4.2.

• Empirically, we observe two notable advantages of ACVI over popular projection-based saddle point methods: (i) the ACVI iterates exhibit significantly reduced rotations, as they approach the solution from the analytic center—see Fig. 1, and (ii) while projection-based methods show extensive zigzagging when hitting a constraint, ACVI avoids this, resulting in more efficient updates—see § 5.

Our convergence guarantees are parameter-free, meaning these do not require knowing apriori the constants of the problem (such as the Lipschitzness constant), and interestingly, the convergence guarantee does not require that $F$ is Lipschitz—this assumption is solely used to express the rate in terms of decreasing the gap function (in contrast to the extragradient method [Korpelevich, 1976] where such assumption is necessary to show convergence—see Theorem 1 therein). To our knowledge, the proposed ACVI method is the first first-order IP algorithm for VIs with global convergence proof.

2 Related works

Unconstrained VIs: methods and guarantees. Apart from the standard gradient descent ascent (GDA) method, among the most commonly used methods for multi-objective optimization are the extragradient method [EG, Korpelevich, 1976], optimistic GDA [OGDA, Popov, 1980], and recently the lookahead [LA, Zhang et al., 2019, Chavdarova et al., 2021b] method was shown to have outperforming performances on challenging non-convex problems—see App. A for full description. In contrast to gradient fields (as in a single-objective setting), when $F$ is a general vector field, the last iterate can be significantly further apart from the solution although the average iterate (of convex $X$) converges to it [Daskalakis et al., 2018, Chavdarova et al., 2019]. This is problematic since it implies that the average convergence guarantee is weaker in the sense that it may not extend to more general setups where we can no longer rely on the convexity of $X$, and even for convex $X$, the convergence will be significantly slower when $X$ is large. In this direction, Golowich et al. [2020b,a] provided last iterate lower bound for the broad class of $p$-stationary canonical linear iterative ($p$-SCLI) first order methods [Arjevani et al., 2016] of $O(\frac{1}{p\sqrt{K}})$. Extensive line of works provide guarantees for the last iterate for varying classes of problems. For the general MVI class, we have guarantees that the following $p$-SCLI methods match the lower bound: (i) Golowich et al. [2020b] obtain the rate in terms of the gap function relying on first and second order smoothness of $F$, and Gorbunov et al. [2022] obtain a rate of $O(\frac{1}{k})$ in terms of reducing the squared norm of the operator relying on first order smoothness of $F$ (Assumption 1) using computer-assisted proof; and (ii) Golowich et al. [2020b] and Chavdarova et al. [2021a] provide the best iterate rate for OGDA.

Constrained zero-sum and VI classes of problems. Gidel et al. [2017] extend the Frank-Wolfe [Frank and Wolfe, 1956, Jaggi, 2013, Lacoste-Julien and Jaggi, 2015] method—also known as the conditional gradient [V.F. Demyanov, 1970]—to solve a subclass of the cVI problem—constrained zero-sum problems, under strong convex-concavity assumption and also when assuming the constraint set is strongly convex—that is, has sublevel sets of strongly convex functions [Vial, 1983]. Daskalakis and Panageas [2019] provide asymptotic last iterate proof for zero-sum convex-concave constrained problems for the optimistic multiplicative weights update (OMWU) method. Similarly, [Wei et al., 2021] focus on OGDA and OMWU in the constrained setting and provide the convergence rates for bilinear games over the simplex. In her seminal work, Korpelevich [1976] proposed the classical
(projected) extragradient method (EG)—see App. A and proved its convergence for monotone (c)VI with L-Lipschitz operator, and as mentioned above, recently [Cai et al., 2022] provided its rate with respect to the gap function using computer-aided proof. For the constrained setting, Tseng [1995] builds on [Pang, 1987] and provides linear convergence rate of EG for strongly monotone $F$ (see Def. 1), whereas Malitsky [2015] focuses on the same setting but on the therein proposed projected reflected gradient method. Diakonikolas [2020] obtains parameter-free guarantee for Halpern iteration [Halpern, 1967] for ccoercive operators (see definition in App. A).

**Interior point (IP) methods in single- and multi-objective settings.** Traditionally IP methods primarily express the inequality constraints by augmenting the objective with the log-barrier penalty (see § 3.1), and then use Newton’s method to solve the sub-problem [Boyd and Vandenberghe, 2004]. The latter involves computing either the inverse of a large matrix or Cholesky decomposition, however, it is yet highly efficient as it requires only a few iterations to converge. Nonetheless, when the dimension of the variable is large, such computation becomes intractable. Among other IP variants that address the above, Lin et al. [2018] replace the Newton step with the ADMM method (projected extragradient method (EG)—see App. A and proved its convergence for monotone (c)VI variants that address the above, Lin et al. [2018] replace the Newton step with the ADMM method (described in § 3.1) which is known to be highly scalable with the dimension [Boyd et al., 2011]. In the context of cVIs, several works apply IP methods, mostly Newton-based. Monteiro and Pang [1996] analyze path-following IP methods for complementarity problems which is a subclass of cVI, using local homeomorphic maps. Chen et al. [1998] provide a superlinear global convergence rate of the smoothing Newton method when $F$ is semismooth for box constrained VIs. Similarly, Qi and Sun [2002], Qi et al. [2000] focus on the smoothing Newton method and provide solely the rate of the outer loop. Ralph and Wright [2000] show superlinear convergence for MVI problems under solely inequality constraints, provided set of assumptions hold—including (i) existence of a strictly complementary solution, (ii) full rank of the Jacobian of the active constraints at the solution, (iii) twice differentiable constraints, among others, and provide a local convergence rate. Fan and Yan [2010] consider solely inequality constraints and propose second-order Newton-based method which has global convergence guarantees under some conditions, and a rate is not provided.

### 3 Preliminaries

**Notation.** Bold small and capital letters denote vectors and matrices, respectively. Sets are denoted with curly capital letters, e.g., $S$. The Euclidean norm of $v$ is denoted by $\|v\|$, and the inner product in Euclidean space with $\langle \cdot, \cdot \rangle$. With $\odot$ we denote element-wise product. We let $[n]$ denote $\{1, \ldots, n\}$ and let $\epsilon$ denote vector of all 1’s.

In the remainder of the paper, we consider a general setting in which the constraint set $C \subseteq X$ is defined as an intersection of finitely many inequalities and linear equalities:

\[
C = \{x \in \mathbb{R}^n | \varphi_i(x) \leq 0, i \in [m], Cx = d\}, \tag{CS}
\]

where each $\varphi_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $C \in \mathbb{R}^{p \times n}$, $d \in \mathbb{R}^p$, where we assume $\text{rank}(C) = p$. For brevity, with $\varphi$ we denote the concatenated $\varphi_i(\cdot), i \in [m]$, and in the remainder of the paper, each $\varphi_i \in \mathbb{R}^1(\mathbb{R}^n), i \in [m]$ and is convex. For convenience we denote:

\[
C_\leq \triangleq \{x \in \mathbb{R}^n | \varphi(x) \leq 0\}, \quad C_< \triangleq \{x \in \mathbb{R}^n | \varphi(x) < 0\}, \quad \text{and} \quad C_- \triangleq \{y \in \mathbb{R}^n | Cy = d\},
\]

thus the relative interior of $C$ is $\text{int} C \triangleq C_\leq \cap C_-$, and we consider $\text{int} C \neq \emptyset$ and $C$ is compact.

In the following, we list the definitions and assumptions we refer to later on.

**Definition 1 ((strong/ξ) monotonicity).** An operator $F : X \supseteq S \rightarrow \mathbb{R}^n$ is monotone on $S$ if:

\[
(x - x', F(x) - F(x')) \geq 0, \quad \forall x, x' \in S.
\]

$F$ is said to be $\xi$-monotone on $S$ iff: $\exists \xi > 0$, and $\xi > 1$ such that $(x - x', F(x) - F(x')) \geq c\|x - x'\|^\xi$ for all $x, x' \in S$. Finally, $F$ is $\mu$-strongly monotone on $S$ iff: $\exists \mu > 0$, s.t. $(x - x', F(x) - F(x')) \geq \mu\|x - x'\|^2$, for all $x, x' \in S$. Moreover, we say that an operator $F$ is star--monotone, star--$\xi$-monotone or star--strongly-monotone (on $S$) if the respective definition holds for $x^* \equiv x^*$, where $x^* \in \text{SOL}(S, F)$.

Note how “star--” definitions are weaker relative to their respective non-star counterparts. The above definition holds similarly for unconstrained VIs, by setting $S \equiv \mathbb{R}^n$. The analog of the function values—used as a performance measure for convergence rates in convex optimization—for cVI is the gap function (a.k.a., the optimality gap or primal gap), defined next.
Definition 2 (Gap function). Given a candidate point \( x' \in X \) in a parameter space and a map \( F : X \supseteq S \to \mathbb{R}^n \) where \( S \) is compact, the gap function \( \mathcal{G} : \mathbb{R}^n \to \mathbb{R} \) is defined as \( \mathcal{G}(x', S) = \max_{x \in S} \langle F(x'), x' - x \rangle \).

Note that the gap function requires \( S \) to be compact in order to be defined (as otherwise it can be infinite). We will rely on the following assumption to express our rates in terms of the gap function.

Assumption 1 (First-order Smoothness). Let \( F : X \supseteq S \to \mathbb{R}^n \) be an operator, we say that \( F \) satisfies \( \ell \)-first-order smoothness on \( S \), or \( L \)-smoothness, if \( F \) is a \( L \)-Lipschitz map, that is \( \exists L > 0 \) such that \( \|F(x) - F(x')\| \leq L \|x - x'\| \), for \( \forall x, x' \in S \).

As an informal summary, the solution existence guarantee follows when \( X \) is compact, see Chapter 2.2 of [Facchinei and Pang, 2003], and App. A.2.

3.1 Relevant path-following interior-point methods

In this section, we overview the interior-point approach to single-objective optimization, focusing on aspects that are most relevant to our proposed method. Consider the following problem:

\[
\min_{x} f(x) \quad \text{s.t.} \quad \varphi(x) \leq 0 \quad \text{and} \quad Cx = d, \tag{cCVX}
\]

where \( f, \varphi : \mathbb{R}^n \to \mathbb{R} \) are convex and continuously differentiable, \( x \in \mathbb{R}^n, C \in \mathbb{R}^{\nu \times n}, \) and \( d \in \mathbb{R}^\nu \).

IP methods solve problem (cCVX) by reducing it to a sequence of linear equality-constrained problems via the logarithmic barrier [see, for example, Boyd and Vandenberghe, 2004, Chapter 11]:

\[
\min_{x} f(x) - \mu \sum_{i=1}^m \log(-\varphi_i(x)) \quad \text{s.t.} \quad Cx = d, \quad \text{with} \quad \mu > 0. \tag{l-cCVX}
\]

Assume that (l-cCVX) has a solution for each \( \mu > 0 \), and let \( x^\mu \) denote the solution of (l-cCVX) for a given \( \mu \). The central path of (l-cCVX) is defined as the set of points \( x^\mu, \mu > 0 \). Note that \( x^\mu \in \mathbb{R}^n \) is a strictly feasible point of (cCVX) as it satisfies \( \varphi(x^\mu) < 0 \) and \( Cx^\mu = d \).

Alternating direction method of multipliers (ADMM) method. ADMM [Glowinski and Marroco, 1975, Gabay and Mercier, 1976, Lions and Mercier, 1979, Glowinski and Le Tallec, 1989] is a gradient-based algorithm for convex optimization problems that splits the objective into sub-problems each of which are easier to solve. Its popularity is due to its computational scalability [Boyd et al., 2011]. Consider a problem of the following form:

\[
\min_{x,y} f(x) + g(y) \quad \text{s.t.} \quad Ax + By = b, \tag{ADMM-Pr}
\]

where \( f, g : \mathbb{R}^n \to \mathbb{R} \) are convex, \( x, y \in \mathbb{R}^n, A, B \in \mathbb{R}^{\nu \times n}, \) and \( b \in \mathbb{R}^\nu \). The augmented Lagrangian function \( \mathcal{L}_\beta(\cdot) \) of the (ADMM-Pr) problem is:

\[
\mathcal{L}_\beta(x, y, \lambda) = f(x) + g(y) + \langle x - y, \lambda \rangle + \frac{\beta}{2} \|x - y\|^2 \tag{AL-CVX}
\]

where \( \beta > 0 \) is often called the penalty parameter. If the augmented Lagrangian method is used to solve (AL-CVX), at each step \( k \) we have: \( x_{k+1}, y_{k+1} = \arg\min_{x,y} \mathcal{L}_\beta(x, y, \lambda_k) \) and \( \lambda_{k+1} = \lambda_k + \beta(Ax_{k+1} + By_{k+1} - b) \), where the latter step is gradient ascent on the dual. In contrast, ADMM updates \( x \) and \( y \) in an alternating way as follows—as the sub-problems are often simpler:

\[
\begin{align*}
    x_{k+1} &= \arg\min_x \mathcal{L}_\beta(x, y_k, \lambda_k), \\
    y_{k+1} &= \arg\min_y \mathcal{L}_\beta(x_{k+1}, y_k, \lambda_k), \tag{ADMM}
\end{align*}
\]

4 ACVI: An Interior Point Method for Constrained Variational Inequalities

4.1 Deriving the ACVI algorithm

In this section, we derive an interior-point method for the cVI problem that we refer to as ACVI (ADMM-based Interior problem for constrained VIIs). Primarily, we restate the cVI problem in
a form that will allow for developing an interior point based method. By the definition of \( cVI \) it follows [see §1.3 in Facchinei and Pang, 2003] that:

\[
x \in SOL(C, F) \Leftrightarrow \begin{cases} w = z \\ z = \arg \min_x F(w)^\top x \\ \text{s.t. } \varphi(x) \leq 0 \\ Cx = d \end{cases} \Leftrightarrow \begin{cases} F(x) + \nabla \varphi^\top(x) \lambda + C^\top \nu = 0 \\ \lambda \perp \varphi(x) \leq 0 \end{cases}
\]

(KKT)

where \( \lambda \in \mathbb{R}^m \) and \( \nu \in \mathbb{R}^p \) are dual variables. Recall that we assume that \( intC \neq \emptyset \), thus, by the Slater condition (using the fact that \( \varphi_i(x), i \in [m] \) are convex) and the KKT conditions, the second equivalence holds, yielding the KKT system of \( cVI \). Note that the above equivalence also guarantees the two solutions coincide; see Facchinei and Pang [2003, Prop. 1.3.4 (b)]. Analogous to the method described in § 3, we add a log-barrier term to the objective to remove the inequality constraints and obtain the following modified version of (KKT):

\[
\begin{align*}
\min_x F(w)^\top x - \mu \sum_{i=1}^m \log \left(-\varphi_i(x)\right) & \quad \text{s.t. } Cx = d, \quad \text{ where we fix } w = w_k. \\
& \quad \text{(KKT-2)}
\end{align*}
\]

with \( \mu > 0, \ e \triangleq [1, \ldots, 1]^\top \in \mathbb{R}^m \). Again, the equivalence holds by the KKT and the Slater condition. We derive the update rule at step \( k \) via the following sub-problem: \( \min_x F(w_k)^\top x - \mu \sum_{i=1}^m \log \left(-\varphi_i(x)\right), \quad \text{s.t. } Cx = d, \) where we fix \( w = w_k \). Notice that (i) \( w_k \) is a constant vector in this sub-problem, and (ii) the objective is split, making ADMM a natural choice to solve the sub-problem. To apply ADMM-type of method, we introduce a new variable \( y \in \mathbb{R}^n \) yielding:

\[
\begin{align*}
\min_{x,y} F(w_k)^\top x + \mathbb{1}[Cx = d] - \mu \sum_{i=1}^m \log \left(-\varphi_i(y)\right), \quad \mathbb{1}[Cx = d] \triangleq \begin{cases} 0, & \text{if } Cx = d \\ +\infty, & \text{if } Cx \neq d. \end{cases} \\
\text{s.t. } x = y
\end{align*}
\]

Note that \( \mathbb{1}[Cx = d] \) is a generalized real-valued convex function of \( x \). We introduce the following:

\[
P_c \triangleq \mathbf{I} - C^\top (CC^\top)^{-1}C, \quad (P_c) \quad \text{and} \quad d_c \triangleq C^\top (CC^\top)^{-1}d, \quad (d_c\text{-EQ})
\]

where \( P_c \in \mathbb{R}^{n \times n} \) and \( d_c \in \mathbb{R}^n \). The augmented Lagrangian of (1) is thus:

\[
\mathcal{L}_\beta(x, y, \lambda) = F(w_k)^\top x + \mathbb{1}[Cx = d] - \mu \sum_{i=1}^m \log (-\varphi_i(y)) + \langle \lambda, x - y \rangle + \frac{\beta}{2} \| x - y \|^2, \quad (AL)
\]

where \( \beta > 0 \) is the penalty parameter. Finally, using ADMM, we have the following update rule for \( x \) at step \( k \):

\[
x_{k+1} = \arg \min_{x \in C_{\infty}} \mathcal{L}_\beta(x, y_k, \lambda_k) = \arg \min_{x \in C_{\infty}} \frac{\beta}{2} \| x - y_k + \frac{1}{\beta}(F(w_k) + \lambda_k) \|^2. \quad (2)
\]

This yields the following expression for \( x \):

\[
x_{k+1} = P_c \left( y_k - \frac{1}{\beta}(F(w_k) + \lambda_k) \right) + d_c. \quad (X\text{-EQ})
\]

For \( y \) and the dual variable \( \lambda \), we have:

\[
y_{k+1} = \arg \min_y \mathcal{L}_\beta(x_{k+1}, y, \lambda_k) = -\mu \sum_{i=1}^m \log (-\varphi_i(y)) + \frac{\beta}{2} \| y - x_{k+1} - \frac{1}{\beta} \lambda_k \|^2, \quad (Y\text{-EQ})
\]

\[
\lambda_{k+1} = \lambda_k + \beta(x_{k+1} - y_{k+1}). \quad (\lambda\text{-EQ})
\]

Next, we derive the update rule for \( w \). We set \( w_k \) to be the solution of the following eq. (w.r.t. \( w \)):

\[
w + \frac{1}{\beta} P_c F(w) - P_c y_k + \frac{1}{\beta} P_c \lambda_k - d_c = 0. \quad (W\text{-EQ})
\]

The following theorem ensures the solution of (W-EQ) exists and is unique, see App. B.1 for proof.
Further notice that which is a stronger assumption than monotonicity, yet weaker. Algorithm 1 ACVI pseudocode.

Theorem 1 (W-EQ: solution uniqueness). If \( F \) is monotone on \( \mathcal{C}_\omega \), the following statements hold true for the solution of (W-EQ): (i) it always exists, (ii) it is unique, and (iii) it is contained in \( \mathcal{C}_\omega \).

Remark 1. Note that when there are no equality constraints, \( \mathcal{C}_\omega \) becomes the entire space \( \mathbb{R}^n \). Further notice that \( w_k = x_{k+1} \), thus it is redundant to state it in the algorithm, and we remove \( w \).

We are now ready to state the full algorithm as Algorithm 1, see App. B for a discussion. In the remaining of the paper, where clear from the context, we drop the superscript from the iterate \( x_k^{(t)} \).

For completeness, App. C presents a variant of Algorithm 1 that follows similar derivation and discusses the advantages of Algorithm 1.

Algorithm 1 ACVI pseudocode.

1. **Input:** operator \( F : \mathcal{X} \rightarrow \mathbb{R}^n \), equality \( Cx = d \) and inequality constraints \( \varphi_i(x) \leq 0, i = [m] \), hyperparameters \( \mu_\perp, \beta > 0, \delta \in (0, 1) \), number of outer and inner loop iterations \( T \) and \( K \), resp.
2. **Initialize:** \( y_0^{(0)} \in \mathbb{R}^n, \lambda_0^{(0)} \in \mathbb{R}^n \) where \( P_c \in \mathbb{R}^{n \times n} \)
3. \( d_c \equiv C^\top (CC^\top)^{-1}d \)
4. \( d_e \equiv C^\top (CC^\top)^{-1}d \)
5. for \( t = 0, \ldots, T - 1 \) do
6. \( \mu_t = \delta \mu_{t-1} \)
7. for \( k = 0, \ldots, K - 1 \) do
8. Set \( x_{k+1}^{(t)} \) to be the solution of: \( x + \frac{1}{\beta} P_c F(x) - P_c y_k^{(t)} + \frac{1}{\beta} P_c \lambda_k^{(t)} - d_e = 0 \) (w.r.t. \( x \))
9. \( y_{k+1}^{(t)} = \arg \min_y - \mu_t \sum_{i=1}^m \log (-\varphi_i(y)) + \frac{\gamma}{2} \| y - x_{k+1}^{(t)} - \frac{1}{\beta} \lambda_k^{(t)} \|^2 \)
10. \( \lambda_{k+1}^{(t)} = \lambda_k^{(t)} + \beta(x_{k+1}^{(t)} - y_{k+1}^{(t)}) \)
11. end for
12. \( (y_0^{(T)}, \lambda_0^{(T)}) \equiv (y_K^{(T)}, \lambda_K^{(T)}) \)
13. end for

4.2 Convergence Analysis

We consider two broad classes of problems. The first class relies on that \( F \) is \( \xi \)-monotone on \( \mathcal{C}_\omega \)—which is a stronger assumption than monotonicity, yet weaker than strong-monotonicity. The second setup requires that (i) \( F \) is monotone, (ii) the constraints are active at the solution, and (iii) \( F \) is not purely rotational. Although may seem otherwise, (iii) is weaker than requiring that the active constraints at the solution form an acute angle with the operator—other words, given the latter, the former holds due to monotonicity of \( F \)—see App. B. Note that (iii) is not strong, as purely rotational games occur “almost never” in Baire category sense [Kupka, 1963, Smale, 1963, Balduzzi et al., 2018, Hsieh et al., 2021]. The proofs are given in App. B. The theoretical results use the following lemma.

Lemma 1 (Upper bound for \( \mathcal{G}(\cdot) \)). When \( F \) is \( \mathcal{L} \)-Lipschitz on \( \mathcal{C}_\omega \)—as per Assumption 1—we have that any iterate \( x_k \) produced by Algorithm 1 satisfies: \( \mathcal{G}(x_k, \mathcal{C}) \leq M_0 \| x_k - x^* \| \), where \( M_0 > 0 \) and \( x^* \in \text{SOL}(\mathcal{C}, F) \).

To state the results we define the following sets. For \( r, s > 0 \), let \( \mathcal{C}_r \equiv \{ x \in \mathbb{R}^n | Cx = d, \varphi(x) \leq r e \} \), and similarly let \( \mathcal{C}_s \equiv \{ x \in \mathbb{R}^n | Cx - d \| x \| \leq s, \varphi(x) \leq 0 \} \). We have the following.

Theorem 2 (Last and average iterate convergence for star-\( \xi \)-monotone operator). Given an operator \( F : \mathcal{X} \rightarrow \mathbb{R}^n \) monotone on \( \mathcal{C}_\omega \) (Def. 1), and either \( F \) is strictly monotone on \( \mathcal{C} \) or one of the \( \varphi_i \) is strictly convex. Assume there exists \( r > 0 \) or \( s > 0 \) such that \( F \) is star-\( \xi \)-monotone—as per Def. 1—on either \( \mathcal{C}_r \) or \( \mathcal{C}_s \), resp. Let \( x_k^{(t)} \) and \( \bar{x}_k^{(t)} \equiv \frac{1}{K} \sum_{i=1}^K x_k^{(t)} \) denote the last and average iterate of Algorithm 1, respectively, run with sufficiently small \( \mu_\perp \). Then for all \( t \in [T] \), we have that:

1. \( \| x_K^{(t)} - x^* \| \leq O\left( \frac{1}{K^{\tau-\gamma}} \right) \), and \( \| \bar{x}_K^{(t)} - x^* \| \leq O\left( \frac{1}{K^{\tau-\gamma}} \right) \).
2. Moreover, if \( F \) is \( \mathcal{L} \)-Lipschitz on \( \mathcal{C}_\omega \)—as per Assumption 1—the same corresponding upper bounds hold for \( \mathcal{G}(x_K^{(t)}, \mathcal{C}) \) and \( \mathcal{G}(\bar{x}_K^{(t)}, \mathcal{C}) \).
Remark 2. Note that the convergence guarantee does not rely on Assumption 1, and it is solely used to relate the rate to the gap function. Also note that \( \mu_{-1} \) does not impact the convergence rate. Moreover, for simplicity we state the result with sufficiently small \( \mu_{-1} \), however the proof extends to any \( \mu_{-1} > 0 \), that is, the above result can be made parameter-free, see App. B.

Theorem 3 (Last and average iterate convergence for monotone operator). Given an operator \( F : X \to \mathbb{R}^n \), assume: (i) \( F \) is monotone on \( C_\varphi \) as per Def. 1; (ii) either \( F \) is strictly monotone on \( C \) or one of \( \varphi_i \), is strictly convex; and (iii) \( \inf_{x \in S \setminus \{x^*\}} F(x)^* + \frac{x - x^*}{\|x - x^*\|} > 0 \), where \( S \equiv \hat{C}_r \) or \( \hat{C}_s \). Let \( \hat{x}^{(t)}_K \)

and \( \hat{x}^{(t)}_{\bar{K}} \) denote the last and average iterate of Algorithm 1, respectively, run with sufficiently small \( \mu_{-1} \). Then for all \( t \in [T] \), we have that:

1. \( \|\hat{x}^{(t)}_{\bar{K}} - x^*\| \leq O\left(\frac{1}{T}\right) \).

2. If in addition \( \inf_{x \in S \setminus \{x^*\}} F(x)^* + \frac{x - x^*}{\|x - x^*\|} > 0 \), then \( \|\hat{x}^{(t)}_{K} - x^*\| \leq O\left(\frac{1}{T}\right) \).

3. Moreover, if \( F \) is \( L \)-Lipschitz on \( C_\varphi \)—as per Assumption 1—the same corresponding upper bounds hold for \( G(x_K^{(t)}, C) \) and \( G(x^{(t)}_{\bar{K}}, C) \).

Assumption (iii) in Theorem 3 requires the angle of \( F(x) \) and \( x - x^* \) to be acute on \( S \setminus \{x^*\} \), where \( S \equiv \hat{C}_r \) or \( \hat{C}_s \). For example, when there are no equality constraints, Assumption (iii) becomes \( \inf_{x \in C \setminus \{x^*\}} F(x)^* + \frac{x - x^*}{\|x - x^*\|} > 0 \) (cVI) and by the monotonicity of \( F \), we can see that for any point \( x \in C \setminus \{x^*\} \), the angle between \( F(x) \) and \( x - x^* \) is always less than or equal to \( \pi/2 \). And assumption (iii) requires that \( F(x^*) \neq 0 \), which means the constraints are active at \( x^* \), and \( \forall \theta \in (0, \pi/2) \) s.t. for any \( x \in C \setminus \{x^*\} \), the angle between \( F(x) \) and \( x - x^* \) is upper bounded by \( \theta \).

Remark 3. Our proofs rely on the existence of the central path—see Appendix A. Note that since \( C \) is compact, it suffices that either: (i) \( F \) is strictly monotone on \( C \), or that (ii) one of the inequality constraints \( \varphi_i \) is strictly convex for the central path to exist [Facchinei and Pang, 2003, Corollary 11.4.24]. Thus, if \( F \) is \( \xi \)-monotone on \( C \), then the central path exists. However, to relax the former assumption, notice that—by the compactness of \( C \)—there exists a sufficiently large \( M \) such that for any \( x \in C \), \( x^\top x \leq M \). Thus, one can add a strictly convex inequality constraint \( \varphi_{m+1}(x) \) e.g., \( x^\top x - M \leq 0 \) and the solution set remains intact. That is, as \( \mu \) tends to 0 the original problem is recovered. This ensures the existence of the central path without changing the original problem.

Figure 2: Convergence of GDA, EG, OGDA, LA-GDA, and ACVI on three different 2d problems, for fixed number of total iterations, where markers denote the iterates of the respective method. See § 5 and App. D for discussion and implementation details, respectively.

5 Experiments

Methods. We compare ACVI with the projected variants of the most commonly used saddle point optimizers (fully described in App. A.4): (i) GDA. (ii) EG [Korpelevich, 1976], (iii) OGDA [Popov,
1000, and (iv) LA-\tilde{k}-GDA [Zhang et al., 2019, Chavdarova et al., 2021b], where \( \tilde{k} \) is the hyperparameter of LA, and we use GDA as a base optimizer. App. D lists implementation details.

Problems. For 2D problems we use: (i) \( \epsilon BG \): the common bilinear game, constrained on \( \mathbb{R}_+ \) for the two players, stated in Fig. 1, (ii) Von Neumann’s ratio game [Von Neumann, 1971, Daskalakis et al., 2020, Diakonikolas et al., 2021], (iii) Forsaken game [Hsieh et al., 2021]—which exhibits limit cycles, as well as (iv) toy GAN—used in [Daskalakis et al., 2018, Antonakopoulos et al., 2021]. Note that these are known to be challenging problems in the literature, and interestingly the latter three are non-monotone, going beyond the assumptions of our theoretical results. We also consider the following higher-dimension bilinear game on the probability simplex, with \( \eta \in (0, 1) \), \( n = 1000 \):

\[
\min_{\vec{x}_1} \max_{\vec{x}_2} \eta \vec{x}_1^T \vec{x}_1 + (1 - \eta) \vec{x}_1^T \vec{x}_2 - \eta \vec{x}_2^T \vec{x}_2 ; \quad \Delta = \{ \vec{x}_i \in \mathbb{R}^{500} | \vec{x}_i \geq 0, \text{ and } e^T \vec{x}_i = 1 \}. \quad (\text{HBG})
\]

While the herein numerical results aim to show the qualitative differences of Algorithm 1, App. E shows results on MNIST [Lecun and Cortes, 1998] using GANs, which we augment with constraints.

Results. The \( \epsilon BG \) problem is depicted in Fig. 1, where for clarity we omit (LA-)GDA and OGDA as these perform similarly to EG—see App. E. Fig. 2 depicts the results on the remaining 2D problems. We observe that a single step of ACVI significantly reduces the distance to the solution. Moreover, while projection-based methods zigzag when hitting a constraint, ACVI avoids this, see further App. E. Fig. 3 depicts the results on (HBG): (a) indicates that ACVI is time efficient, and (b) that relative to projection based methods, ACVI performs well for varying rotational intensity \( (1 - \eta) \).

6 Discussion

We derived a first-order interior point method for the constrained VI (cVI) problem—named ACVI, showed its convergence for two broad classes of problems, and provided the corresponding rates. ACVI allows for using general constraints and does not require a feasible point at initialization. Numerical experiments show that standard projection-based saddle point methods extensively zigzag when hitting a constraint due to the rotational component of the vector field. ACVI on the other hand, fully avoids this by incorporating the constraints in the update rule, resulting in a more efficient optimization.

Interestingly, our numerical analyses included problems that go beyond the herein considered assumptions, thus it remains an open problem to prove the convergence for wider problem classes. On the engineering side, we showed that ACVI performs well on MNIST, and it remains an interesting direction to further exploit the structure of the sub-problems of ACVI for higher computational efficiency.

Acknowledgements

TC is supported by the Swiss National Science Foundation (SNSF), grant P2ELP2_199740. The authors thank Tianyi Lin and Matteo Pagliardini for insightful discussions and feedback. TY thanks Zhouchen Lin for his collection and overview of Alternating Direction Method of Multipliers methods.
References

Kimon Antonakopoulos, E. Veronica Belmega, and Panayotis Mertikopoulos. Adaptive extra-gradient methods for min-max optimization and games. In ICLR, 2021.

Yossi Arjevani, Shai Shalev-Shwartz, and Ohad Shamir. On lower and upper bounds for smooth and strongly convex optimization problems. In JMLR, 2016.

Waïss Azizian, Ioannis Mitliagkas, Simon Lacoste-Julien, and Gauthier Gidel. A tight and unified analysis of gradient-based methods for a whole spectrum of differentiable games. In AISTATS, pages 2863–2873, 2020.

David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The mechanics of n-player differentiable games. In ICML, 2018.

Heinz H. Bauschke and Patrick L. Combettes. Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer Publishing Company, Incorporated, 2nd edition, 2017. ISBN 1441994661.

Axel Bohm. Solving nonconvex-nonconcave min-max problems exhibiting weak minty solutions. arXiv:2201.12247, 2022.

Radu Ioan Bot, Ernö Robert Csetnek, and Dang-Khoa Nguyen. Fast OGDA in continuous and discrete time. arXiv:2203.10947, 2022.

Stephen Boyd and Lieven Vandenberghe. Convex optimization. Cambridge university press, 2004.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 3, 2011. ISSN 1935-8237. doi: 10.1561/2200000016.

Yang Cai, Argyris Oikonomou, and Weiqiang Zheng. Tight last-iterate convergence of the extragradient method for constrained monotone variational inequalities. arXiv:2204.09228, 2022.

Tatjana Chavdarova, Gauthier Gidel, François Fleuret, and Simon Lacoste-Julien. Reducing noise in GAN training with variance reduced extragradient. In NeurIPS, 2019.

Tatjana Chavdarova, Michael I. Jordan, and Manolis Zampetakis. Last-iterate convergence of saddle point optimizers via high-resolution differential equations. NeurIPS Workshop on Optimization for Machine Learning, 2021a.

Tatjana Chavdarova, Matteo Pagliardini, Sebastian U Stich, François Fleuret, and Martin Jaggi. Taming GANs with Lookahead-Minmax. In ICLR, 2021b.

Xiaojun Chen, Liqun Qi, and Defeng Sun. Global and superlinear convergence of the smoothing newton method and its application to general box constrained variational inequalities. Mathematics of computation, 67(222):519–540, 1998.

Richard W. Cottle and George B. Dantzig. Complementary pivot theory of mathematical programming. Linear Algebra and its Applications, 1(1):103–125, 1968. ISSN 0024-3795.

Constantinos Daskalakis and Ioannis Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. In ITCS, 2019.

Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training GANs with optimism. In ICLR, 2018.

Constantinos Daskalakis, Dylan J. Foster, and Noah Golowich. Independent policy gradient methods for competitive reinforcement learning. In NeurIPS, 2020.

Damek Davis and Wotao Yin. Convergence rate analysis of several splitting schemes. In Splitting methods in communication, imaging, science, and engineering, pages 115–163. Springer, 2016.

Jelena Diakonikolas. Halpern iteration for near-optimal and parameter-free monotone inclusion and strong solutions to variational inequalities. COLT, 125, 2020.
Jelena Diakonikolas, Constantinos Daskalakis, and Michael I. Jordan. Efficient methods for structured nonconvex-nonconcave min-max optimization. In AISTATS, 2021.

Jonathan Eckstein and Dimitri P Bertsekas. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. Mathematical Programming, 55(1):293–318, 1992.

Francisco Facchinei and Jong-Shi Pang. Finite-dimensional variational inequalities and complementarity problems. Springer, 2003.

Xiaona Fan and Qinglun Yan. An interior point algorithm for variational inequality problems. International Journal of Contemporary Mathematical Sciences, 5(52):2595–2604, 2010.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. Naval Research Logistics Quarterly, 3(1-2):95–110, 1956. doi: https://doi.org/10.1002/nav.380030109.

Daniel Gabay. Chapter IX applications of the method of multipliers to variational inequalities. In Studies in mathematics and its applications, volume 15, pages 299–331. Elsevier, 1983.

Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Computers & Mathematics with Applications, 2:17–40, 1976. ISSN 0898-1221.

Gauthier Gidel, Tony Jebara, and Simon Lacoste-Julien. Frank-Wolfe algorithms for saddle point problems. In AISTATS, 2017.

R. Glowinski and A. Marroco. Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique, 9(R2):41–76, 1975.

Roland Glowinski and Patrick Le Tallec. Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics. Society for Industrial and Applied Mathematics, 1989. doi: 10.1137/1.9781611970838.

Noah Golowich, Sarath Pattathil, and Constantinos Daskalakis. Tight last-iterate convergence rates for no-regret learning in multi-player games. In NeurIPS, 2020a.

Noah Golowich, Sarath Pattathil, Constantinos Daskalakis, and Asman Ozdaglar. Last iterate is slower than averaged iterate in smooth convex-concave saddle point problems. In COLT, pages 1758–1784, 2020b.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In NIPS, 2014.

Eduard Gorbunov, Nicolas Loizou, and Gauthier Gidel. Extragradient method: $O(1/K)$ last-iterate convergence for monotone variational inequalities and connections with cocoercivity. In AISTATS, 2022.

Benjamin Halpern. Fixed points of nonexpanding maps. Bulletin of the American Mathematical Society, 73:957 – 961, 1967.

Bingsheng He and Xiaoming Yuan. On the $O(1/n)$ convergence rate of the Douglas–Rachford alternating direction method. SIAM Journal on Numerical Analysis, 50(2):700–709, 2012.

Bingsheng He and Xiaoming Yuan. On non-ergodic convergence rate of douglas–rachford alternating direction method of multipliers. Numerische Mathematik, 130(3):567–577, 2015.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In NIPS, 2017.

Ya-Ping Hsieh, Panayotis Mertikopoulos, and Volkan Cevher. The limits of min-max optimization algorithms: convergence to spurious non-critical sets. In ICML, 2021.
Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, volume 28, pages 427–435. PMLR, 2013.

Galina Michailovna Kornejeva. The extragradient method for finding saddle points and other problems. *Matecon*, 1976.

Ivan Kupka. Contributiona la théorie des champs génériques. *Contributions to differential equations*, 2:457–484, 1963.

Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *NIPS*, 2015.

Yann Lecun and Corinna Cortes. The MNIST database of handwritten digits. 1998. URL http://yann.lecun.com/exdb/mnist/.

Tianyi Lin, Shiqian Ma, Yinyu Ye, and Shuzhong Zhang. An admm-based interior-point method for large-scale linear programming. *arXiv:1805.12344*, 2018.

P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

Yu. Malitsky. Projected reflected gradient methods for monotone variational inequalities. *SIAM Journal on Optimization*, 25:502–520, 2015.

Nimrod Megiddo. *Progress in Mathematical Programming: Interior-Point and Related Methods*. Springer-Verlag, 1989. ISBN 0387968474.

Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. *arXiv:1901.08511*, 2019.

Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. Convergence rate of $O(1/k)$ for optimistic gradient and extra-gradient methods in smooth convex-concave saddle point problems. In *SIAM Journal on Optimization*, 2020.

Renato DC Monteiro and Jong-Shi Pang. Properties of an interior-point mapping for mixed complementarity problems. *Mathematics of Operations Research*, 21(3):629–654, 1996.

Yurii Nesterov and Arkadi Nemirovski. Interior-point polynomial algorithms in convex programming. In *Siam studies in applied mathematics*, 1994.

Yurii Nesterov and Michael J. Todd. Primal-dual interior-point methods for self-scaled cones. *SIAM Journal on Optimization*, 8:324–360, 1998.

Shayegan Omidshafiei, Jason Pazis, Chris Amato, Jonathan P. How, and John Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *ICML*, 2017.

Jong-Shi Pang. A posteriori error bounds for the linearly-constrained variational inequality problem. *Mathematics of Operations Research*, 12:474–484, 1987.

Leonid Denisovich Popov. A modification of the arrow–hurwicz method for search of saddle points. *Mathematical Notes of the Academy of Sciences of the USSR*, 28(5):845–848, 1980.

Liqun Qi and Defeng Sun. Smoothing functions and smoothing newton method for complementarity and variational inequality problems. *Journal of Optimization Theory and Applications*, 113(1):121–147, 2002.

Liqun Qi, Defeng Sun, and Guanglu Zhou. A new look at smoothing newton methods for non-linear complementarity problems and box constrained variational inequalities. *Mathematical programming*, 87(1):1–35, 2000.
Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In ICLR, 2016.

Daniel Ralph and Stephen J Wright. Superlinear convergence of an interior-point method despite dependent constraints. Mathematics of Operations Research, 25(2):179–194, 2000.

James Renegar. A Mathematical View of Interior-Point Methods in Convex Optimization. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2001. ISBN 9780898715026.

Ralph Tyrrell Rockafellar. Monotone operators associated with saddle-functions and minimax problems. Nonlinear functional analysis, 18(part 1):397–407, 1970.

Ernest K. Ryu and Wotao Yin. Large-Scale Convex Optimization via Monotone Operators. Springer Publishing Company, Incorporated, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In NIPS, 2016.

Stephen Smale. Stable manifolds for differential equations and diffeomorphisms. Annali della Scuola Normale Superiore di Pisa-Classe di Scienze, 17(1-2):97–116, 1963.

Guido Stampacchia. Formes bilineaires coercitives sur les ensembles convexes. Académie des Sciences de Paris, 258:4413–4416, 1964.

Paul Tseng. A path-following algorithm for linear programming using quadratic and logarithmic penalty functions. SIAM Journal on Control and Optimization, 31:1578–1598, 1993.

Paul Tseng. On linear convergence of iterative methods for the variational inequality problem. Journal of Computational and Applied Mathematics, 60:237–252, 1995. ISSN 0377-0427.

A.M Rubinov V.F Demyanov. Approximate Methods in Optimization Problems, volume 9. American Elsevier, New York, 1970.

Jean-Philippe Vial. Strong and weak convexity of sets and functions. Mathematics of Operations Research, 8:231–259, 1983.

John Von Neumann. A Model of General Economic Equilibrium. Palgrave Macmillan UK, 1971. ISBN 978-1-349-15430-2.

John von Neumann and Oskar Morgenstern. Theory of games and economic behavior. Princeton University Press, 1947.

Chen-Yu Wei, Chung-Wei Lee, Mengxiao Zhang, and Haipeng Luo. Linear last-iterate convergence in constrained saddle-point optimization. In ICLR, 2021.

Stephen J. Wright. Primal-Dual Interior-Points Methods. SIAM, 1997.

Stephen J. Wright. Lectures on Modern Convex Optimization. Analysis, Algorithms, and Engineering Applications. SIAM, 2001.

Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. In NeurIPS, 2019.
A Background: Additional Details

This section lists additional background relevant to some discussions in the main part, stated for completeness.

A.1 Additional VI definitions & equivalent formulations

Seeing an operator \( F : \mathcal{X} \to \mathbb{R}^n \) as the graph \( GrF = \{(x, y) | x \in \mathcal{X}, y = F(x)\} \), its inverse \( F^{-1} \) is defined as \( GrF^{-1} = \{(y, x) | (y, x) \in GrF\} \). See for example [Ryu and Yin, 2022] for further discussion. We denote the projection to the set \( \mathcal{X} \) with \( \Pi_{\mathcal{X}} \).

**Definition 3 \((\frac{1}{\mu\text{-cocoercive operator}})\).** An operator \( F : \mathcal{X} \supseteq S \to \mathbb{R}^n \) is \( \frac{1}{\mu\text{-cocoercive (or }\frac{1}{\mu\text{-inverse}}\text{ strongly monotone) on } S \text{ if its inverse (graph) } F^{-1} \text{ is } \mu\text{-strongly monotone on } S, \text{ that is,} \)

\[ \exists \mu > 0, \quad \text{s.t.} \quad \langle x - x', F(x) - F(x') \rangle \geq \mu \|F(x) - F(x')\|^2, \forall x, x' \in S. \]

It is star \( \frac{1}{\mu\text{-cocoercive if the above holds when setting } x' \equiv x^* \text{ where } x^* \text{ denotes a solution, that is:} \)

\[ \exists \mu > 0, \quad \text{s.t.} \quad \langle x - x^*, F(x) - F(x^*) \rangle \geq \mu \|F(x) - F(x^*)\|^2, \forall x \in S, x^* \in SOL(\mathcal{X}, F). \]

Note from Def. 3 that cocoercivity is a strict subclass of monotone and L-Lipschitz operators, thus it is a stronger assumption. See Chapter 4.2 of [Bauschke and Combettes, 2017] for further relations of cocoercivity with other properties of operators.

In the following, we will make use of the natural and normal mappings of an operator \( F : \mathcal{X} \to \mathbb{R}^n \), where \( \mathcal{X} \subset \mathbb{R}^n \). Following the notation of [Facchinei and Pang, 2003], the natural map \( F^{NAT}_{\mathcal{X}} : \mathcal{X} \to \mathbb{R}^n \) is:

\[ F^{NAT}_{\mathcal{X}} \triangleq x - \Pi_{\mathcal{X}}(x - F(x)), \quad \forall x \in \mathcal{X}, \quad \text{(F-NAT)} \]

whereas the normal map \( F^{NOR}_{\mathcal{X}} : \mathbb{R}^n \to \mathbb{R}^n \) is:

\[ F^{NOR}_{\mathcal{X}} \triangleq F(\Pi_{\mathcal{X}}(x)) + x - \Pi_{\mathcal{X}}(x), \quad \forall x \in \mathbb{R}^n. \quad \text{(F-NOR)} \]

Moreover, we have the following solution characterizations:

(i) \( x^* \in SOL(\mathcal{X}, F) \) iff \( F^{NAT}_{\mathcal{X}}(x^*) = 0 \), and

(ii) \( x^* \in SOL(\mathcal{X}, F) \) iff \( \exists x' \in \mathbb{R}^n \text{ s.t. } x^* = \Pi_{\mathcal{X}}(x') \) and \( F^{NOR}_{\mathcal{X}}(x') = 0 \).

A.2 Existence of solution

We provide brief (informal) summary of some sufficient conditions for solution existence, that \( SOL(\mathcal{X}, F) \neq \emptyset \). See Chapter 2 of [Facchinei and Pang, 2003] for full treatment of the topic.

The common underlying tool to establish that a solution to the VI(\( \mathcal{X}, F \)) problem exists is using *topological degree*. The topological degree tool is designed so as to satisfy the so called *homotopy invariance* axiom, what in turn allows for reducing a solution existence question of a complicated map to a simpler one (which is homotopy invariant to the original one) which we can more easily show (e.g., the identity map on a closed domain). It can be used (as one way) to prove the celebrated *Brouwer fixed-point theorem*, which states that any continuous map \( \Phi : S \to S \), where \( S \) is nonempty convex compact set, has a fixed point in \( S \).

We have the following sufficient condition, see [Cor. 2.2.5 Facchinei and Pang, 2003]

**Theorem 4** (sufficient condition for existence of the solution, Cor. 2.2.5 [Facchinei and Pang, 2003]). If \( \mathcal{X} \subseteq \mathbb{R}^n \) is compact and convex, and \( F : \mathcal{X} \to \mathbb{R}^n \) is continuous, then the solution set is nonempty and compact.

It can also be shown that when \( \mathcal{X} \) is closed convex (and \( F \) continuous), if one can find \( x' \in \mathcal{X} \) s.t. \( \langle F(x), x - x' \rangle \geq 0, \forall x \in \mathcal{X} \), then \( SOL(\mathcal{X}, F) \neq \emptyset \). The same conclusion follows if one can show that \( \exists x' \in \mathcal{X} \) and the set \( \{x \in \mathcal{X} \mid \langle F(x), x - x' \rangle < 0\} \) is bounded (possibly empty, see Prop. 2.2.3 in [Facchinei and Pang, 2003]).

Sufficient conditions can also be established via the natural and the normal map due to the above solution characterizations. In this case, we require that \( F \) is continuous on an open set \( S \) and we are
interested if $SOL(\mathcal{X}, F) \neq \emptyset$, where $\mathcal{X}$ is assumed closed and convex and subset of $\mathcal{S}$, $\mathcal{X} \subseteq \mathcal{S}$. If one establishes that a solution exists for $F_{\mathcal{X}}^{\text{NAT}}$ on a bounded open set $\mathcal{U}$, and if $\text{cl} \mathcal{U} \subseteq \mathcal{S}$, then it follows that $SOL(\mathcal{X}, F) \neq \emptyset$. Similar implication holds when we have guarantee for $F_{\mathcal{X}}^{\text{NOR}}$. See Theorem 2.2.1 of [Facchinei and Pang, 2003].

In summary, the guarantee follows from the boundness of some set which includes $\mathcal{X}$, the boundness of the set of “potential solutions” (if we can construct such set), or the compactness of $\mathcal{X}$ itself.

### A.3 Existence of Central path

In this section, we discuss the results that establish guarantees of the existence of the central path. Let $L(x, \lambda, \nu) \triangleq F(x) + \nabla \varphi^T(x)\lambda + C^T\nu$. For $(\lambda, w, x, \nu) \in \mathbb{R}^{2m+n+p}$, let

$$G(\lambda, w, x, \nu) \triangleq \begin{pmatrix} w \circ \lambda \\ w + g(x) \\ L(x, \lambda, \nu) \\ h(x) \end{pmatrix} \in \mathbb{R}^{2m+n+p},$$

and

$$H(\lambda, w, x, \nu) \triangleq \begin{pmatrix} w + g(x) \\ L(x, \lambda, \nu) \\ h(x) \end{pmatrix} \in \mathbb{R}^{m+n+p}.$$

We write $H_{++} \triangleq H(\mathbb{R}^{2m}_+ \times \mathbb{R}^n \times \mathbb{R}^p)$. By Facchinei and Pang [Corollary 11.4.24 2003] we have the following proposition.

**Proposition 1** (Sufficient condition for the existence of the central path.) If $F$ is monotone, either $F$ is strictly monotone or one of $\varphi_i$ is strictly convex, and $C$ is bounded. The following four statements hold for the functions $G$ and $H$:

1. $G$ maps $\mathbb{R}^{2m}_+ \times \mathbb{R}^{n+p}$ homeomorphically onto $\mathbb{R}^m_+ \times H_{++}$;
2. $\mathbb{R}^m_+ \times H_{++} \subseteq G(\mathbb{R}^{2m}_+ \times \mathbb{R}^{n+p})$;
3. for every vector $a \in \mathbb{R}^m_+$, the system $H(\lambda, w, x, \nu) = 0$, $w \circ \lambda = a$ has a solution $(\lambda, w, x, \nu) \in \mathbb{R}^{2m}_+ \times \mathbb{R}^{n+p}$; and
4. the set $H_{++}$ is convex.

### A.4 Saddle-point optimization methods

In this section, we describe in detail the saddle point methods that we compare with in the main paper in § 5. We denote the projection to the set $\mathcal{X}$ with $\Pi_{\mathcal{X}}$, and when the method is applied in the unconstrained setting $\Pi_{\mathcal{X}} \equiv I$.

For instructive purposes, consider the following constrained zero sum game:

$$\min_{x_1 \in \mathcal{X}_1} \max_{x_2 \in \mathcal{X}_2} f(x_1, x_2), \quad (\text{ZS-G})$$

where $f : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}$ is smooth and convex in $x_1$ and concave in $x_2$. As in the main paper, we write $x \triangleq (x_1, x_2) \in \mathbb{R}^n$.

**(Projected) Gradient Descent Ascent (GDA).** The extension of gradient descent for the cVI problem is gradient descent ascent (GDA). The GDA operator $F : \mathcal{X} \to \mathbb{R}^n$ and its Jacobian $J$ are defined as:

$$F(x) = \begin{bmatrix} \nabla_{x_1} f(x) \\ -\nabla_{x_2} f(x) \end{bmatrix}, \quad J(x) = \begin{bmatrix} \nabla_{x_1}^2 f(x) & \nabla_{x_2} \nabla_{x_1} f(x) \\ -\nabla_{x_1} \nabla_{x_2} f(x) & -\nabla_{x_2}^2 f(x) \end{bmatrix}.$$  

In the remaining, we will only refer to the joint variable $x$ and (with abuse of notation) the subscript will denote the step. Letting $\gamma \in [0, 1]$ denote the step size, the GDA update at step $k$ is then:

$$x_{k+1} = \Pi_{\mathcal{X}}(x_k - \gamma F(x_k)). \quad (\text{GDA})$$
(Projected) Extragradient (EG). EG [Korpelevich, 1976] uses a “prediction” step to obtain an extrapolated point $x_{k+\frac{1}{2}}$ using GDA: 

$$x_{k+\frac{1}{2}} = \Pi_X \left( x_k - \gamma F(x_k) \right).$$

The gradients at the extrapolated point are then applied to the current iterate $x_t$:

$$x_{k+1} = \Pi_X \left( x_k - \gamma F(\Pi_X (x_k - \gamma F(x_k))) \right).$$

(EG)

In the original EG paper, [Korpelevich, 1976] proved that the EG method converges for monotone VIs, as follows.

**Theorem 5 (Korpelevich [1976]).** Given a map $F : \mathcal{X} \mapsto \mathbb{R}^n$, if the following is satisfied:

1. the set $\mathcal{X}$ is closed and convex,
2. $F$ is single-valued, definite, and monotone on $\mathcal{X}$—as per Def. 1,
3. $F$ is L-Lipschitz— as per Asm. 1.

then there exists a solution $x^* \in \mathcal{X}$, such that the iterates $x_k$ produced by the EG update rule converge to it, that is $x_k \rightarrow x^*$, as $k \rightarrow \infty$.

Facchinei and Pang [2003] also show that for any convex-concave function $f$ and any closed convex sets $x_1 \in \mathcal{X}_1$ and $x_2 \in \mathcal{X}_2$, the EG method converges [Facchinei and Pang, 2003, Theorem 12.1.11].

(Projected) Optimistic Gradient Descent Ascent (OGDA). The update rule of Optimistic Gradient Descent Ascent OGDA [(OGDA) Popov, 1980] is:

$$x_{n+1} = \Pi_X \left( x_n - 2\gamma F(x_n) + \gamma F(x_{n-1}) \right).$$

(OGDA)

(Projected) Lookahead–Minmax (LA). The LA algorithm for min-max optimization [Chavdarova et al., 2021b], originally proposed for minimization by Zhang et al. [2019], is a general wrapper of a “base” optimizer where, at every step $t$: (i) a copy of the current iterate $\bar{x}_n$ is made: $\bar{x}_n \leftarrow x_n$. (ii) $\bar{x}_n$ is updated $k \geq 1$ times, yielding $\bar{x}_{n+k}$, and finally (iii) the actual update $x_{n+1}$ is obtained as a point that lies on a line between the current $x_n$ iterate and the predicted one $\bar{x}_{n+k}$:

$$x_{n+1} \leftarrow x_n + \alpha (\bar{x}_{n+k} - x_n), \quad \alpha \in [0, 1].$$

(LA)

In this work, we use solely GDA as a base optimizer for LA, and denote it with $LkGDA$.  

16
B Omitted Proofs and Discussions concerning Algorithm 1

This section provides the proofs of the theoretical results in § 4.2.

B.1 Proof of Theorem 1: uniqueness of the solution of (W-EQ)

Proof of Theorem 1: uniqueness of the solution of (W-EQ). Let \( G(x) \) denote the LHS of (W-EQ), that is:

\[
G(x) \triangleq x + \frac{1}{\beta} P_c F(x) - P_c y_k + \frac{1}{\beta} P_c \lambda_k - d_c
\]

(3)

We claim that \( G(x) \) is strongly monotone on \( \mathbb{R}^n \). In fact, \( \forall x, y \in \mathbb{R}^n, P_c(x - y) = x - y \). Note that \( P_c \) is symmetric, thus we have:

\[
\langle G(x) - G(y), x - y \rangle = \|x - y\|^2 + \frac{1}{\beta} \langle P_c F(x) - P_c F(y), x - y \rangle
\]

\[
= \|x - y\|^2 + \frac{1}{\beta} \langle x - y, F(x) - F(y) \rangle
\]

\[
\geq \|x - y\|^2.
\]

Therefore, according to Theorem 2.3.3 (b) in [Facchinei and Pang, 2003], \( SOL(\mathbb{C}_\infty, G) \) has a unique solution \( \bar{x} \in \mathbb{C}_\infty \). Thus, we have:

\[
G(\bar{x})^T (x - \bar{x}) = 0, \ \forall x \in \mathbb{C}_\infty
\]

From the above, we deduce that \( G(x) \in \text{Span} \{ c_1, \cdots, c_p \} \), where \( c_i \) is the row vectors of \( C \), \( i \in [p] \).

Suppose \( G(\bar{x}) = \sum_{i=1}^p \alpha_i c_i \). Note that \( CG(x) = 0, \ \forall x \in \mathbb{C}_\infty \), we have that:

\[
e_i^T G(\bar{x}) = e_j^T \sum_{i=1}^p \alpha_i c_i = 0, \ \forall j \in [p].
\]

Thus,

\[
\langle G(\bar{x}), G(\bar{x}) \rangle = \sum_{i=1}^p \alpha_i c_i \sum_{i=1}^p \alpha_i c_i = 0,
\]

which indicates that \( G(\bar{x}) = 0 \). Hence, \( \bar{x} \) is a solution of (W-EQ) in \( \mathbb{C}_\infty \).

On the other hand, \( \forall x \in \mathbb{R}^n \), if \( x \) is a solution of (W-EQ), i.e. \( G(x) = 0 \), then \( x \in \mathbb{C}_\infty \). By the uniqueness of \( \bar{x} \) in \( \mathbb{C}_\infty \) we have that \( x = \bar{x} \), which means \( \bar{x} \) is unique in \( \mathbb{R}^n \).

\[ \square \]

B.2 Proof of Lemma 1: Upper bound on the gap function

Proof of Lemma 1: Upper bound on the gap function. Let \( x_k \) denote an iterate produced by Algorithm 1, and let \( x \in C \). Note that we always have \( x_k \in \mathbb{C}_\infty \). We have that:

\[
\langle F(x_k), x_k - x \rangle = \langle F(x^*), x^* - x \rangle + \langle F(x_k), x_k - x \rangle - \langle F(x^*), x^* - x \rangle
\]

\[
= \langle F(x^*), x^* - x \rangle + \langle F(x_k), x_k - x^* + x^* - x \rangle - \langle F(x^*), x^* - x \rangle.
\]

From the proof of Theorem 6 we know that \( x_k \) is bounded, which gives: \( \langle F(x_k), x_k - x^* \rangle \leq M \|x_k - x^*\|, \) with \( M > 0 \), as well as that \( \langle F(x^*), x^* - x \rangle \leq 0 \). Thus, for the above we get:

\[
\langle F(x_k), x_k - x \rangle \leq \langle F(x_k) - F(x^*), x^* - x \rangle + M \|x_k - x^*\|
\]

\[
\leq D \|F(x_k) - F(x^*)\| + M \|x_k - x^*\|
\]

\[
\leq (DL + M) \|x_k - x^*\|
\]

where for the second row we used \( x^* - x < D^* \), where \( D^* \triangleq \max_{x \in C} \|x^* - x\| \) is the largest distance between any point in \( C \) and \( x^* \). For the last row we used that \( F \) is L-Lipschitz—Assumption 1—which concludes the proof. \[ \square \]

The proof is analogous for the \( y_k \in \mathbb{C}_\infty \) iterates produced by Algorithm 1.
B.3 Proofs of the convergence rate: Theorems 2 and 3

Let
\[
\begin{align*}
  f^{(t)}(x) & \triangleq F(x^{\mu})^T x + \mathbb{1}(C x = d), \\
  f_k^{(t)}(x) & \triangleq F(x_k^{(t)})^T x + \mathbb{1}(C x = d), \\
  g^{(t)}(y) & \triangleq -\mu \sum_{i=1}^{m} \log(-\varphi_i(y)),
\end{align*}
\]
where \(x^{\mu}\) is a solution of (KKT-2) when \(\mu = \mu_t\). Note that the existence of \(x^{\mu}\) is guaranteed by the existence of the central path—see App. A, and that \(f^{(t)}, f_k^{(t)}\) and \(g^{(t)}\) are all convex.

In the following proofs, unless causing confusion, we drop the subscript \(t\) to simplify notations.

Let \(y^{\mu} = x^{\mu}\), from (KKT-2) we can see that \((x^{\mu}, y^{\mu})\) is an optimal solution of
\[
\begin{align*}
  \min_{x, y} & f(x) + g(y) \\
  \text{s.t.} & x = y.
\end{align*}
\]  

(4)

There exists \(\lambda^{\mu} \in \mathbb{R}^n\) such that \((x^{\mu}, y^{\mu}, \lambda^{\mu})\) is a KKT point of (4). We give the following proposition which we will repeatedly use in the proofs:

**Proposition 2.** If \(F\) is monotone, then \(\forall k \in \mathbb{N},\)
\[
f_k(x_{k+1}) - f_k(x^{\mu}) \geq f(x_{k+1}) - f(x^{\mu}).
\]

Furthermore, if \(F\) is \(\xi\)-monotone, as per Def. 1
\[
f_k(x_{k+1}) - f_k(x^{\mu}) \geq f(x_{k+1}) - f(x^{\mu}) + c\|x_{k+1} - x^{\mu}\|_2^\xi.
\]

**Proof of Proposition 2.** It suffices to note that:
\[
f_k(x_{k+1}) - f_k(x^{\mu}) - (f(x_{k+1}) - f(x^{\mu})) = (F(x_{k+1}) - F(x^{\mu}))^T (x_{k+1} - x^{\mu}).
\]

Some of our proofs that follow are inspired by some convergence proofs in ADMM [Gabay, 1983, Eckstein and Bertsekas, 1992, Davis and Yin, 2016, He and Yuan, 2012, 2015]. However, although Algorithm 1 adopts the high level idea of ADMM, we can not directly refer to the convergence proofs of ADMM, but need to substantially modify these.

We will use the following lemma.

**Lemma 2.** \(f(x) + g(y) - f(x^{\mu}) - g(y^{\mu}) + \langle \lambda^{\mu}, x - y \rangle \geq 0, \forall x, y.\)

**Proof.** The Lagrange function of (4) is
\[
L(x, y, \lambda) = f(x) + g(y) + \lambda^T (x - y).
\]

And by the property of KKT point we have
\[
L(x^{\mu}, y^{\mu}, \lambda) \leq L(x^{\mu}, y^{\mu}, \lambda^{\mu}) \leq L(x, y, \lambda^{\mu}), \forall (x, y, \lambda),
\]
from which the conclusion follows.

The next lemma is easy to verify.

**Lemma 3.** If
\[
f(x) + g(y) - f(x^{\mu}) - g(y^{\mu}) + \langle \lambda^{\mu}, x - y \rangle \leq \alpha_1, \\
\|x - y\| \leq \alpha_2
\]
then we have
\[
-\|\lambda^{\mu}\| \alpha_2 \leq f(x) + g(y) - f(x^{\mu}) - g(y^{\mu}) \leq \|\lambda^{\mu}\| \alpha_2 + \alpha_1.
\]
The following lemma lists some simple but useful facts that we will use in the following proofs.

**Lemma 4.** For (4) and Algorithm 1, we have

\begin{align*}
0 & \in \partial f_k(x_{k+1}) + \lambda_k + \beta(x_{k+1} - y_{k+1}) \tag{5} \\
0 & \in \partial g(y_{k+1}) - \lambda_k - \beta(x_{k+1} - y_{k+1}) \tag{6} \\
\lambda_{k+1} - \lambda_k & = \beta(x_{k+1} - y_{k+1}) \tag{7} \\
0 & \in \partial f(x^\mu) + \lambda^\mu \tag{8} \\
0 & \in \partial g(y^\mu) - \lambda^\mu \tag{9} \\
x^\mu & = y^\mu \tag{10}
\end{align*}

We define:

\[ \nabla f_k(x_{k+1}) \triangleq -\lambda_k - \beta(x_{k+1} - y_k), \tag{11} \]

\[ \nabla g(y_{k+1}) \triangleq \lambda_k + \beta(x_{k+1} - y_{k+1}). \tag{12} \]

Then from (5) and (6) we can see that

\[ \nabla f_k(x_{k+1}) \in \partial f_k(x_{k+1}) \text{ and } \nabla g(y_{k+1}) \in \partial g(y_{k+1}). \tag{13} \]

**Lemma 5.** For Algorithm 1, we have

\[ \langle \nabla g(y_{k+1}), y_{k+1} - y \rangle = -\langle \lambda_{k+1}, y - y_{k+1} \rangle, \tag{14} \]

and

\[ \langle \nabla f_k(x_{k+1}), x_{k+1} - x \rangle + \langle \nabla g(y_{k+1}), y_{k+1} - y \rangle = -\langle \lambda_{k+1}, x_{k+1} - y_{k+1} - x + y \rangle + \beta(-y_{k+1} + y_k, x_{k+1} - x). \tag{15} \]

**Proof of Lemma 5.** From (7), (11) and (12) we have:

\[ \langle \nabla f_k(x_{k+1}), x_{k+1} - x \rangle = -\langle \lambda_k + \beta(x_{k+1} - y_k), x_{k+1} - x \rangle = -\langle \lambda_{k+1}, x_{k+1} - x \rangle + \beta(-y_{k+1} + y_k, x_{k+1} - x), \]

and

\[ \langle \nabla g(y_{k+1}), y_{k+1} - y \rangle = -\langle \lambda_{k+1}, y - y_{k+1} \rangle. \]

Adding these together yields (15). \hfill \square

**Lemma 6.** For Algorithm 1, we have

\[
\langle \nabla f_k(x_{k+1}), x_{k+1} - x^\mu \rangle + \langle \nabla g(y_{k+1}), y_{k+1} - y^\mu \rangle + \langle \lambda^\mu, x_{k+1} - y_{k+1} \rangle \\
\leq \frac{1}{2\beta} \| \lambda_k - \lambda^\mu \|^2 - \frac{1}{2\beta} \| \lambda_{k+1} - \lambda^\mu \|^2 \\
+ \beta \| y^\mu - y_k \|^2 - \beta \| y^\mu - y_{k+1} \|^2 \\
- \frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 - \frac{\beta}{2} \| y_k - y_{k+1} \|^2
\]
Proof of Lemma 6. Letting \((x, y, \lambda) = (x^\mu, y^\mu, \lambda^\mu)\) in (15), adding \(\langle \lambda^\mu, x_{k+1} - y_{k+1} \rangle\) to both sides, and using (7) and (10), we have:

\[
\langle \nabla f_k(x_{k+1}), x_{k+1} - x^\mu \rangle + \langle \nabla g(y_{k+1}), y_{k+1} - y^\mu \rangle + \langle \lambda^\mu, x_{k+1} - y_{k+1} \rangle = -\langle \lambda_{k+1} - \lambda^\mu, x_{k+1} - y_{k+1} \rangle + \beta \langle -y_{k+1} + y_k, x_{k+1} - x^\mu \rangle
\]

\[
= \frac{1}{\beta} \langle \lambda_{k+1} - \lambda^\mu, \lambda_{k+1} - \lambda_k \rangle + \langle -y_{k+1} + y_k, \lambda_{k+1} - \lambda_k \rangle
\]

\[
- \beta \langle -y_{k+1} + y_k, y_{k+1} - y^\mu \rangle
\]

\[
= \frac{1}{2\beta} \|\lambda_k - \lambda^\mu\|^2 - \frac{1}{2\beta} \|\lambda_{k+1} - \lambda^\mu\|^2 - \frac{1}{2\beta} \|\lambda_{k+1} - \lambda_k\|^2
\]

\[
+ \frac{1}{2} \|y_k + y^\mu\|^2 - \frac{1}{2} \|y_{k+1} + y^\mu\|^2 - \frac{1}{2} \|y_{k+1} + y_k\|^2
\]

\[
+ \langle -y_{k+1} + y_k, \lambda_{k+1} - \lambda_k \rangle.
\]

On the other hand, (14) gives

\[
\langle \nabla g(y_k), y_k - y \rangle + \langle \lambda_k, -y_k + y \rangle = 0.
\]

Letting \(y = y_k\) in (14) and \(y = y_{k+1}\) in (18), and adding them together, we have:

\[
\langle \nabla g(y_{k+1}) - \nabla g(y_k), y_{k+1} - y_k \rangle + \langle \lambda_{k+1} - \lambda_k, -y_{k+1} + y_k \rangle = 0.
\]

By the monotonicity of \(\partial g\) we know that the first term of the above equality is non-negative. Thus, we have:

\[
\langle \lambda_{k+1} - \lambda_k, -y_{k+1} + y_k \rangle \leq 0.
\]

Plugging it into (17), we have the conclusion. □

Lemma 7. For Algorithm 1, we have

\[
f(x_{k+1}) + g(y_{k+1}) - f(x^\mu) - g(y^\mu) + \langle \lambda^\mu, x_{k+1} - y_{k+1} \rangle
\]

\[
\leq \frac{1}{2\beta} \|\lambda_k - \lambda^\mu\|^2 - \frac{1}{2\beta} \|\lambda_{k+1} - \lambda^\mu\|^2
\]

\[
+ \frac{1}{2} \|y_k + y^\mu\|^2 - \frac{1}{2} \|y_{k+1} + y^\mu\|^2
\]

\[
- \frac{1}{2\beta} \|\lambda_{k+1} - \lambda_k\|^2 - \frac{1}{2} \|y_{k+1} + y_k\|^2
\]

(20)

Furthermore, if \(F\) is \(\xi\)-monotone on \(C\), we have

\[
c \|x_{k+1} - x^\mu\|^2 + f(x_{k+1}) + g(y_{k+1}) - f(x^\mu) - g(y^\mu) + \langle \lambda^\mu, x_{k+1} - y_{k+1} \rangle
\]

\[
\leq \frac{1}{2\beta} \|\lambda_k - \lambda^\mu\|^2 - \frac{1}{2\beta} \|\lambda_{k+1} - \lambda^\mu\|^2
\]

\[
+ \frac{1}{2} \|y_k + y^\mu\|^2 - \frac{1}{2} \|y_{k+1} + y^\mu\|^2
\]

\[
- \frac{1}{2\beta} \|\lambda_{k+1} - \lambda_k\|^2 - \frac{1}{2} \|y_{k+1} + y_k\|^2.
\]

(21)
Proof. From the convexity of \( f_k(x) \) and \( g(y) \) and using Proposition 2 and (13), we have
\[
f(x_{k+1}) + g(y_{k+1}) - f(x^\mu) - g(y^\mu) + \langle \lambda^\mu, x_{k+1} - y_{k+1} \rangle \\
\leq f_k(x_{k+1}) + g(y_{k+1}) - f_k(x^\mu) - g(y^\mu) + \langle \lambda^\mu, x_{k+1} - y_{k+1} \rangle \\
\leq \langle \nabla f_k(x_{k+1}), x_{k+1} - x^\mu \rangle + \langle \nabla g(y_{k+1}), y_{k+1} - y^\mu \rangle + \langle \lambda^\mu, x_{k+1} - y_{k+1} \rangle \\
\leq \frac{1}{2\beta} \| \lambda_k - \lambda^\mu \|^2 - \frac{1}{2\beta} \| \lambda_{k+1} - \lambda^\mu \|^2 \\
+ \frac{\beta}{2} \| -y_k + y^\mu \|^2 - \frac{\beta}{2} \| -y_{k+1} + y^\mu \|^2 \\
- \frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 - \frac{\beta}{2} \| -y_{k+1} + y_k \|^2.
\]
(22)

If \( F \) is \( \xi \)-monotone on \( C \), again by Proposition 2, we can add the term \( c \| x_{k+1} - x^\mu \|^2 \) to the first line and the inequality still holds.

\begin{proof}
\text{Theorem 6.} For Algorithm 1, we have
\[
f(x_{k+1}) + g(y_{k+1}) - f(x^\mu) - g(y^\mu) \to 0, \\
f_k(x_{k+1}) - f_k(x^\mu) + g(y_{k+1}) - g(y^\mu) \to 0, \\
x_{k+1} - y_{k+1} \to 0,
\]
as \( k \to \infty \). Furthermore, if \( F \) is \( \xi \)-monotone on \( C \), we have
\[
x_{k+1} \to x^\mu, \ k \to \infty
\]
\end{proof}

Proof of Theorem 6. Proof From Lemma 2 and (20), we have
\[
\frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 + \frac{\beta}{2} \| -y_{k+1} + y_k \|^2 \\
\leq \frac{1}{2\beta} \| \lambda_k - \lambda^\mu \|^2 - \frac{1}{2\beta} \| \lambda_{k+1} - \lambda^\mu \|^2 \\
+ \frac{\beta}{2} \| -y_k + y^\mu \|^2 - \frac{\beta}{2} \| -y_{k+1} + y^\mu \|^2.
\]
(23)

Summing over \( k = 0, \ldots, \infty \), we have
\[
\sum_{k=0}^{\infty} \left( \frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 + \frac{\beta}{2} \| -y_{k+1} + y_k \|^2 \right) \\
\leq \frac{1}{2\beta} \| \lambda_0 - \lambda^\mu \|^2 + \frac{\beta}{2} \| -y_0 + y^\mu \|^2.
\]
from which we deduce that \( \lambda_{k+1} - \lambda_k \to 0 \) and \( -y_{k+1} + y_k \to 0 \). Moreover, \( \| \lambda_k - \lambda^\mu \|^2 \) and \( \| -y_k + y^\mu \|^2 \) are bounded for all \( k \), as well as \( \| \lambda_k \| \). Since
\[
\lambda_{k+1} - \lambda_k = \beta(x_{k+1} - y_{k+1}) = \beta(x_{k+1} - x^\mu) + \beta(-y_{k+1} + y^\mu)
\]
we deduce that \( x_{k+1} - y_{k+1} \to 0 \) and \( x_{k+1} - x^\mu \) is also bounded.

From (15) and the convexity of \( f \) and \( g \), and using Proposition 2, we have:
\[
f(x_{k+1}) - f(x^\mu) + g(y_{k+1}) - g(y^\mu) \\
\leq f_k(x_{k+1}) - f_k(x^\mu) + g(y_{k+1}) - g(y^\mu) \\
\leq -\langle \lambda_{k+1} - x_{k+1} - y_{k+1} \rangle + \beta \langle -y_{k+1} + y_k, x_{k+1} - x^\mu \rangle \to 0.
\]
On the other hand, from (8), (9), and (10), we have:
\[
f_k(x_{k+1}) - f_k(x^\mu) + g(y_{k+1}) - g(y^\mu) \\
\geq f(x_{k+1}) - f(x^\mu) + g(y_{k+1}) - g(y^\mu) \\
\geq -\langle \lambda^\mu, x_{k+1} - x^\mu \rangle + \langle \lambda^\mu, y_{k+1} - y^\mu \rangle \\
= -\langle \lambda^\mu, x_{k+1} - y_{k+1} \rangle \to 0.
\]
Thus, we have $f(x_{k+1}) - f(x^\mu) + g(y_{k+1}) - g(y^\mu) \to 0$ and $f_k(x_{k+1}) - f_k(x^\mu) + g(y_{k+1}) - g(y^\mu) \to 0, k \to \infty$.

If $F$ is $\xi$-monotone on $C_\infty$, from Lemma 2 and (21) we have

$$c \| x_{k+1} - x^\mu \|_\xi^2 + \frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 + \frac{\beta}{2} \| -y_{k+1} + y_k \|^2 \leq \frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 - \frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 + \frac{\beta}{2} \| -y_k + y^\mu \|^2 + \beta \| -y_k + y^\mu \|^2.$$  \hfill (24)

From which we deduce that:

$$c \| x_{k+1} - x^\mu \|_2^2 + \sum_{k=0}^\infty \left( \frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 + \frac{\beta}{2} \| -y_{k+1} + y_k \|^2 \right) \leq \frac{1}{2\beta} \| \lambda^0 - \lambda^\mu \|^2 + \frac{\beta}{2} \| y^0 - y^\mu \|^2.$$  \hfill (25)

Therefore, $\| x_{k+1} - x^\mu \|_2 \to 0, k \to \infty$. \hfill \Box

**Lemma 8.** For Algorithm 1, we have

$$\frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 + \frac{\beta}{2} \| -y_{k+1} + y_k \|^2 \leq \frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 - \frac{\beta}{2} \| -y_{k+1} + y_k \|^2.$$  \hfill (25)

Furthermore, if $F$ is $\xi$-monotone on $C_\infty$, we have

$$c \| x_{k+1} - x_k \|^2 + \frac{1}{2\beta} \| \lambda_{k+1} - \lambda_k \|^2 + \frac{\beta}{2} \| -y_{k+1} + y_k \|^2 \leq \frac{1}{2\beta} \| \lambda_k - \lambda_{k-1} \|^2 + \frac{\beta}{2} \| -y_k + y_{k-1} \|^2.$$  \hfill (26)

**Proof of Lemma 8.** (15) gives:

$$\langle \nabla f_{k-1}(x_k), x_k - x \rangle + \langle \nabla g(y_k), y_k - y \rangle = -\langle \lambda_k, x_k - y_k - x + y \rangle + \beta(-y_k + y_{k-1}, x_k - x).$$  \hfill (27)
Letting \((x, y, \lambda) = (x_k, y_k, \lambda_k)\) in (15) and \((x, y, \lambda) = (x_{k+1}, y_{k+1}, \lambda_{k+1})\) in (27), and adding them together, and using (7), we have
\[
\langle \hat{\nabla} f_k (x_{k+1}) - \hat{\nabla} f_{k-1}(x_k), x_{k+1} - x_k \rangle + \langle \hat{\nabla} g (y_{k+1}) - \hat{\nabla} g (y_k), y_{k+1} - y_k \rangle \\
= - \langle \lambda_{k+1} - \lambda_k, x_{k+1} - y_{k+1} - x_k + y_k \rangle + \beta (-y_{k+1} + y_k - (-y_k + y_{k-1}), x_{k+1} - x_k \rangle \\
= - \frac{1}{\beta} \langle \lambda_{k+1} - \lambda_k, \lambda_{k+1} - \lambda_k \rangle \\
+ (-y_{k+1} + y_k + (y_k - y_{k-1}), \lambda_{k+1} - \lambda_k + \beta (y_{k+1} - (\lambda_{k+1} - \lambda_k - y_k)) \rangle \\
= \frac{1}{2\beta} \left[ \| \lambda_{k+1} - \lambda_k \|^2 - \| \lambda_{k+1} - \lambda_k \|^2 - \| \lambda_{k+1} - \lambda_k \|^2 \right] \\
+ \frac{\beta}{\sqrt{\sqrt{K + 1}}} \left[ \| -y_{k+1} + y_k - (-y_k + y_{k-1}) \|^2 - \| -y_{k+1} + y_k - (-y_k + y_{k-1}) \|^2 \right] \\
+ (-y_{k+1} + y_k + (y_k - y_{k-1}), \lambda_{k+1} - \lambda_k + (\lambda_{k+1} - \lambda_k - y_k) \rangle \\
\leq \frac{1}{2\beta} \left[ \| \lambda_{k+1} - \lambda_k \|^2 - \| \lambda_{k+1} - \lambda_k \|^2 \right] + \frac{\beta}{\sqrt{\sqrt{K + 1}}} \left[ \| -y_{k+1} + y_k - (-y_k + y_{k-1}) \|^2 - \| -y_{k+1} + y_k - (-y_k + y_{k-1}) \|^2 \right]
\]

Note that
\[
\langle \hat{\nabla} f_k (x_{k+1}) - \hat{\nabla} f_{k-1}(x_k), x_{k+1} - x_k \rangle \\
= \langle \hat{\nabla} f_k (x_{k+1}) - \hat{\nabla} f_{k}(x_k), x_{k+1} - x_k \rangle + \langle \hat{\nabla} f_k (x_k) - \hat{\nabla} f_{k-1}(x_k), x_{k+1} - x_k \rangle \\
= \langle \hat{\nabla} f_k (x_{k+1}) - \hat{\nabla} f_k (x_k), x_{k+1} - x_k \rangle + \langle F(x_{k+1}) - F(x_k), x_{k+1} - x_k \rangle
\]

Using the monotonicity of \(\partial f_k, \partial g\) and \(F\), we draw the conclusion.

\textbf{Theorem 7.} If \(F\) is monotone on \(C_m\), then for Algorithm 1, we have
\[
-\|\lambda^\mu\| \sqrt{\frac{\Delta}{\beta(K + 1)}} \leq f(x_{K+1}) + g(y_{K+1}) - f(x^\mu) - g(y^\mu) \\
\leq f(x_{K+1}) + g(y_{K+1}) - f(x^\mu) - g(y^\mu) \\
\leq \frac{\Delta}{K + 1} + \frac{2\Delta}{\sqrt{K + 1}} + \|\lambda^\mu\| \sqrt{\frac{\Delta}{\beta(K + 1)}},
\]

\[
\|x_{K+1} - y_{K+1}\| \leq \sqrt{\frac{\Delta}{\beta(K + 1)}},
\]

where \(\Delta \triangleq \frac{1}{\beta} \|\lambda_0 - \lambda_\mu\|^2 + \beta \|y_0 - y^\mu\|^2\).

Furthermore, if \(F\) is \(\xi\)-monotone on \(C_m\), we have
\[
c\|\hat{x}_{K+1} - x^\mu\| \leq \frac{\Delta}{K + 1},
\]
\[
c\|x_{K+1} - x^\mu\| \leq \frac{\Delta}{K + 1} + \frac{2\Delta}{\sqrt{K + 1}}.
\]

where \(\hat{x}_{K+1} = \frac{1}{K + 1} \sum_{k=1}^{K+1} x_k\).

\textbf{Proof of Theorem 7.} Summing (23) over \(k = 0, 1, \cdots, K\) and using the monotonicity of \(\frac{1}{\beta} \|\lambda_{k+1} - \lambda_k\|^2 + \frac{\beta}{\sqrt{K + 1}}\|y_{k+1} - y_k\|^2\) from Lemma 8, we have:
\[
\frac{1}{\beta} \|\lambda_{K+1} - \lambda_k\|^2 + \beta \|y_{K+1} - y_k\|^2 \leq \frac{1}{K + 1} \left( \frac{1}{\beta} \|\lambda_0 - \lambda_\mu\|^2 + \beta \|y_0 - y^\mu\|^2 \right)
\]
From which we deduce that
\[ \beta \| x_{K+1} - y_{K+1} \| = \| \lambda_{K+1} - \lambda_K \| \leq \sqrt{\frac{\beta \Delta}{K+1}}, \]
\[ \| - y_{K+1} + y_K \| \leq \sqrt{\frac{\Delta}{\beta (K+1)}}. \]

On the other hand, (23) gives:
\[ \frac{1}{2\beta} \| \lambda_{K+1} - \lambda_K \|^2 + \frac{\beta}{2} \| - y_{K+1} + y_K \|^2 \leq \frac{1}{2\beta} \| \lambda_0 - \lambda_K \|^2 + \frac{\beta}{2} \| - y_0 + y_K \|^2 = \frac{1}{2} \Delta. \]

Hence, we have:
\[ \| \lambda_{K+1} - \lambda_K \| \leq \sqrt{\beta \Delta}, \]
\[ \| - y_{K+1} + y_K \| \leq \sqrt{\frac{\Delta}{\beta}}. \]

Then from (16) and the convexity of \( f \) and \( g \), we have:
\[
\begin{align*}
&f (x_{K+1}) - f (x_K) + g (y_{K+1}) - g (y_K) + \langle \lambda, x_{K+1} - y_{K+1} \rangle \\
&\leq f (x_{K+1}) - f (x_K) + g (y_{K+1}) - g (y_K) + \langle \lambda, x_{K+1} - y_{K+1} \rangle \\
&\leq \frac{1}{\beta} \| \lambda_{K+1} - \lambda_K \| \| x_{K+1} - y_{K+1} \| + \| y_{K+1} + y_K \| \| \lambda_{K+1} - \lambda_K \| \\
&+ \beta \| - y_{K+1} + y_K \| \| - y_{K+1} + y_K \| \leq \frac{\Delta}{K+1} + \frac{2\Delta}{\sqrt{K+1}}.
\end{align*}
\]

From Lemma 3, we have (28).

If in addition \( F \) is \( \xi \)-monotone on \( C_\infty \), using (24), we can obtain the following inequality similar to (30):
\[
\begin{align*}
&c \sum_{k=0}^{K} \| x_{k+1} - x_K \|_2^2 + \frac{1}{\beta} \| \lambda_{K+1} - \lambda_K \|^2 + \beta \| - y_{K+1} + y_K \|^2 \\
&\leq \frac{1}{K+1} \left( \frac{1}{\beta} \| \lambda_0 - \lambda_K \|^2 + \beta \| - y_0 + y_K \|^2 \right).
\end{align*}
\]

By the convexity of \( \| \cdot \|_2^2 \) we have \( c \| \hat{x}_{K+1} - x_K \|_2^2 \leq \frac{\Delta}{K+1} \). And from (32) we can see that \( c \| \hat{x}_{K+1} - x_K \|_2^2 \leq \frac{\Delta}{K+1} + \frac{2\Delta}{\sqrt{K+1}}. \]

**Theorem 8.** If \( F \) is monotone on \( C_\infty \), then for Algorithm 1, we have
\[
\begin{align*}
&| f (\hat{x}_{K+1}) + g (\hat{y}_{K+1}) - f (x_K) - g (y_K) | \leq \frac{\Delta}{2(K+1)} + \frac{2\sqrt{\beta \Delta} \| \lambda_K \|}{\beta (K+1)}, \\
&\| \hat{x}_{K+1} - \hat{y}_{K+1} \| \leq \frac{2\sqrt{\beta \Delta}}{\beta (K+1)}.
\end{align*}
\]

where \( \hat{x}_{K+1} = \frac{1}{K+1} \sum_{k=1}^{K+1} x_k \), \( \hat{y}_{K+1} = \frac{1}{K+1} \sum_{k=1}^{K+1} y_k \), and \( \Delta \) is defined in Theorem 7.
Proof of Theorem 8. Summing (20) over \( k = 0, 1, \cdots, K \), dividing both sides with \( K + 1 \), and using the definitions of \( \hat{x}_{K+1} \) and \( \hat{y}_{K+1} \) and the convexity of \( f \) and \( g \), we have
\[
f(\hat{x}_{K+1}) + g(\hat{y}_{K+1}) - f(x^\mu) - g(y^\mu) + \langle \lambda^\mu, \hat{x}_{K+1} - \hat{y}_{K+1} \rangle \leq \frac{\Delta}{2(K+1)}.
\]

From (7) and (31), we have:
\[
\|\hat{x}_{K+1} - \hat{y}_{K+1}\| = \frac{1}{\beta (K+1)} \left\| \sum_{k=0}^{K} (\lambda_{k+1} - \lambda_k) \right\|
\leq \frac{1}{\beta (K+1)} \|\lambda_{K+1} - \lambda_0\|
\leq \frac{1}{\beta (K+1)} (\|\lambda_0 - \lambda^\mu\| + \|\lambda_{K+1} - \lambda^\mu\|)
\leq \frac{2\sqrt{\beta \Delta}}{\beta (K+1)}.
\]

Finally, from Lemma 3, the conclusion follows. \(\square\)

From Proposition 1 and the fact that \( C \) is compact we can see that \( \lim_{\mu \to 0} x^\mu \) exists and is finite. Let \( x^* = \lim_{\mu \to 0} x^\mu \), then \( x^* \in SOL(C, F) \). Note that when \( F \) is monotone on \( C_\infty \), by (29) we have that \( \forall t \geq 0 \) and \( \forall r,s > 0 \), there exists \( K_0 \in \mathbb{N} \), s.t. \( \forall k \geq K_0, x^{(t)}_k \in \hat{C}_r \) and \( y^{(t)}_k \in \hat{C}_s \). Knowing this fact, it is not hard to see that Theorem 2 and Theorem 3 is a direct consequence of the following theorem:

Theorem 9. \( \exists \mu > 0 \), s.t. if \( \mu < \mu \), then \( |F(x^{(t)}_{K+1})^T(x^{(t)}_{K+1} - x^*)| + |F(x^*)^T(x^{(t)}_{K+1} - x^*)| \leq O\left(\frac{1}{\sqrt{K+1}}\right) \), and \( |F(x^*)^T(\hat{x}^{(t)}_{K+1} - x^*)| \leq O\left(\frac{1}{\sqrt{K+1}}\right) \), \( \forall K \geq 0 \).

Proof of Theorem 9. For simplicity we let \( B(x) \) denote the \( \log \)-barrier term \( \sum_{i=1}^{m} \log(-\varphi_i(x)) \). From Theorem 7 and 8 we have
\[
\left|F(x^\mu)^T(x^{(t)}_{K+1} - x^\mu) + \mu(B(y^{(t)}_{K+1}) - B(y^\mu))\right|
= \left|f(x^{(t)}_{K+1}) - f(x^\mu) + g(y^{(t)}_{K+1}) - g(y^\mu)\right|
\leq O\left(\frac{1}{\sqrt{K+1}}\right),
\]
\[
\left|F(x^{(t)}_{K+1})^T(x^{(t)}_{K+1} - x^\mu) + \mu(B(y^{(t)}_{K+1}) - B(y^\mu))\right|
= \left|f(x^{(t)}_{K+1}) - f(x^\mu) + g(y^{(t)}_{K+1}) - g(y^\mu)\right|
\leq O\left(\frac{1}{\sqrt{K+1}}\right),
\]
\[
\left|F(x^\mu)^T(\hat{x}^{(t)}_{K+1} - x^\mu) + \mu(B(\hat{y}^{(t)}_{K+1}) - B(y^\mu))\right|
= \left|f(\hat{x}^{(t)}_{K+1}) - f(x^\mu) + g(\hat{y}^{(t)}_{K+1}) - g(y^\mu)\right|
\leq O\left(\frac{1}{K+1}\right).
\]

25
From which we can see that $\exists \tilde{\mu} > 0$, s.t. if $\mu_t < \tilde{\mu}$, then we have
\[
\left| F(x^{*})^\top (x^{(t)}_{K+1} - x^{*}) \right| \leq O\left( \frac{1}{\sqrt{K+1}} \right),
\]
\[
\left| F(x^{(t)}_{K+1})^\top (x^{(t)}_{K+1} - x^{*}) \right| \leq O\left( \frac{1}{\sqrt{K+1}} \right),
\]
\[
\left| F(x^{*})^\top (\hat{x}^{(t)}_{K+1} - x^{*}) \right| \leq O\left( \frac{1}{K+1} \right).
\]

From Theorem 9 we can see that by setting $\mu - 1$ small enough, Theorem 2 and 3 hold true. But since we do not know exactly how small $\mu - 1$ should be, in practice, we can set a small $\mu - 1$, and $\mu_t$ could be smaller than any fixed positive number after a very small number of outer loops, as $\mu_t$ decays exponentially. Thus, Algorithm 1 is actually parameter-free.

Note that the assumption (iii) in Theorem 3 is the weakening of the assumption $\inf_{x \in \mathcal{S} \setminus \{x^{*}\}} F(x^{*})^\top (x - x^{*}) \parallel x - x^{*} \parallel > 0$, where $\mathcal{S} = \hat{\mathcal{C}}_r$ or $\tilde{\mathcal{C}}_s$. In fact, by the monotonicity of $F$, we have $F(x)^\top (x - x^{*}) \geq F(x^{*})^\top (x - x^{*})$, so if $\inf_{x \in \mathcal{S} \setminus \{x^{*}\}} F(x^{*})^\top (x - x^{*}) \parallel x - x^{*} \parallel > 0$, there must be $\inf_{x \in \mathcal{S} \setminus \{x^{*}\}} F(x)^\top (x - x^{*}) \parallel x - x^{*} \parallel > 0$.

### B.4 Discussion on Algorithm 1

**Advantages and disadvantages of Algorithm 1.** In Algorithm 1, the updating of $x$ requires solving (W-EQ). Our method is very suitable for problems where (W-EQ) is easy to solve analytically. For example, the affine variational problems or problems with low dimension. For problems where (W-EQ) is hard to solve—for example, min-max optimization problems in GANs—one could use other unconstrained solvers like GDA or EG methods to solve $VI(\mathbb{R}^n, G)$, where $G$ is defined in (3). See App. E for an example applying our method to the MNIST dataset using EG and GDA to solve (W-EQ), where we observe that our method is competitive with standard methods (given fixed execution time).

Note that when there are constraints, the projection-based methods such as GDA, EG, OEGDA etc. require solving a quadratic programming problem in each iteration (or twice per iteration for EG). This problem is often nontrivial and solving it may require using an interior point method. On the other hand, Algorithm 1 can be applied directly when the problem is simpler, or can be combined with unconstrained solvers.

**When given different type of constraints.** If there are no equality constraints, then $\mathcal{C}_e$ becomes $\mathbb{R}^n$. In this case, we have that $x^{(t)}_{k+1}$ is the solution of:
\[
x_k + \frac{1}{\beta} (F(x) + \lambda_k^{(t)}) - y_k^{(t)} = 0.
\]

When there are no inequality constraints, we let $y_k = x_k$ and $\lambda_k = 0$ for every $k$, and we can remove the outer loop. Thus Algorithm 1 can be simplified to update only one variable $x$ each iteration with the following updating rule:
\[
x_{k+1} \text{ is the unique solution of } x + \frac{1}{\beta} P_c F(x) - P_c x_k - d_c = 0.
\]
C  Variant of the ACVI Algorithm (v-ACVI)

Observe from (1) that we could also consider different splitting than that in § 4. For completeness, in this section we present a variant of Algorithm 1, and discuss its advantages and disadvantages relative to Algorithm 1.

C.1 Introduction of the variant ACVI

By considering a different splitting in (1) we get:

\[
\begin{align*}
\min_{x,y} & \quad F(w) \top x - \mu \sum_{i=1}^{m} \log(-\varphi_i(x)) + 1(Cy = d) \\
\text{s.t.} & \quad x = y
\end{align*}
\]

where: \(1(Cy = d) = \begin{cases} 0, & \text{if } Cy = d \\ +\infty, & \text{if } Cy \neq d \end{cases} \).

The augmented Lagrangian of (33) is thus:

\[
\mathcal{L}_\beta(x, y, \lambda) = F(w) \top x - \mu \sum_{i=1}^{m} \log(-\varphi_i(x)) + 1(Cy = d) + \langle \lambda, x - y \rangle + \frac{\beta}{2} \|x - y\|^2
\]

where \(\beta > 0\) is the penalty parameter. We have that for \(x\) at step \(k + 1\):

\[
x_{k+1} = \arg\min_{x} \mathcal{L}_\beta(x, y_k, \lambda_k)
\]

\[
= \arg\min_{x} \frac{1}{2} \left\| x - y_k + \frac{1}{\beta} (F(w) + \lambda_k) \right\|^2 - \frac{\mu}{\beta} \sum_{i=1}^{m} \log(-\varphi_i(x))
\]

The following proposition ensures the existence and uniqueness of \(x_{k+1}\) in \(C_\prec\). i.e. We show that \(x_{k+1}\) is the unique solution in \(C_\prec\) of the following closed form equation (see App. C.2 for its proof):

\[
x - y_k + \frac{1}{\beta} (F(w) + \lambda_k) - \frac{\mu}{\beta} \sum_{i=1}^{m} \nabla \varphi_i(x) \varphi_i(x) = 0.
\]

**Proposition 3** (unique solution). The problem (X-CF) has a solution in \(C_\prec\) and the solution is unique.

For \(y\), the updating rule is

\[
y_{k+1} = \arg\min_{y} \mathcal{L}_\beta(x_{k+1}, y, \lambda_k)
\]

\[
= \arg\min_{y \in C_\prec} -\frac{1}{\beta} (\lambda_k) \top y + \frac{1}{2} \|y - x_{k+1}\|^2
\]

\[
= \arg\min_{y \in C_\prec} \frac{1}{2} \|y - x_{k+1} - \frac{1}{\beta} \lambda_k\|^2
\]

\[
= P_c(x_{k+1} + \frac{\lambda_k}{\beta}) + d_c
\]

And the updating rule for dual variable \(\lambda\) is

\[
\lambda_{k+1} = \lambda_k + \beta(x_{k+1} - y_{k+1})
\]

As in § 4, we’d like to choose \(w_k\) so that \(w_k = x_{k+1}\). To this end, we need the following assumption:

**Assumption 2.** \(\forall b \in \mathbb{R}^n \text{ and } \mu > 0, x + \frac{1}{\beta} F(x) - \frac{\mu}{\beta} \sum_{i=1}^{m} \nabla \varphi_i(x) + b = 0\) has a solution in \(C_\prec\).

If Assumption 2 holds true, we can let \(w_k\) be a solution of

\[
w - y_{k+1} + \frac{1}{\beta} (F(w) + \lambda_{k+1}) - \frac{\mu}{\beta} \sum_{i=1}^{m} \nabla \varphi_i(w) \varphi_i(w) = 0
\]
in $C_\prec$. And by Proposition 3, we can let $x_{k+1}$ be the unique solution of

$$x - y_k + \frac{1}{\beta} (F(w_k) + \lambda_k) - \frac{\mu}{\beta} \sum_{i=1}^{m} \nabla \varphi_i(x) = 0 \quad (x)$$

in $C_\prec$.

Note that $w_k$ is also a solution of $(x)$. By the uniqueness of the solution of $(x)$ (See Prop. 3) we know that $w_k = x_{k+1}$.

So in the $(k+1)$-th step, we can compute $x, y, \lambda$ and $w$ by $(x), (y), (\lambda)$ and (35), respectively. Since $w_k = x_{k+1}$, we can simplify our algorithm by removing variable $w$ and only keep $x, y$ and $\lambda$, see Algorithm 2.

Algorithm 2 v-ACVI pseudocode.

1: Input: operator $F : \mathcal{X} \rightarrow \mathbb{R}^n$, equality $Cx = d$ and inequality constraints $\varphi_i(x) \leq 0, i = [m]$, hyperparameters $\mu, \beta > 0, \delta \in (0, 1)$, number of outer and inner loop iterations $T$ and $K$, resp.

2: Initialize: $y_0^{(0)} \in \mathbb{R}^n, \lambda_0^{(0)} \in \mathbb{R}^n$

3: $P_c \triangleq I - C^\top (CC^\top)^{-1} C$

4: $d_c \triangleq C^\top (CC^\top)^{-1} d$

5: for $t = 0, \ldots, T - 1$ do

6: $\mu_t = \delta \mu_t$

7: Denote $\hat{\varphi}(\lambda, y)$ as the solution of $\frac{1}{\beta} \left( \mu_t \sum_{i=1}^{m} \nabla \varphi_i(x) - F(x) - \lambda \right) + y - x = 0$ with respect to $x$, where $y, \lambda$ are variables

8: for $k = 0, \ldots, K - 1$ do

9: $x_{k+1}^{(t)} = \hat{\varphi}(\lambda_k^{(t)}, y_k^{(t)})$

10: $y_{k+1}^{(t)} = P_c(x_{k+1}^{(t)} + \frac{\lambda_k^{(t)}}{\beta}) + d_c$

11: $\lambda_{k+1}^{(t)} = \lambda_k^{(t)} + \beta(x_{k+1}^{(t)} - y_{k+1}^{(t)})$

12: end for

13: $(y_0^{(t+1)}, \lambda_0^{(t+1)}) \triangleq (y_K^{(t)}, \lambda_K^{(t)})$

14: end for

Compared to Algorithm 1, the updating of $x$ in Algorithm 2 becomes more complicated and the updating of $y$ becomes simpler.

By proofs analogous to appendix B, we can get similar convergence results as Algorithm 1. Specifically, Theorem 2 and 3 are still valid for Algorithm 2, except that we need to replace the assumption “$F$ is monotone on $C_\prec$” with “$F$ is monotone on $C_\prec$” in those two Theorems.

C.2 Proof of Proposition 3

First, we give the following lemma:

Lemma 9. $\forall b \in \mathbb{R}^n, \forall x \in C_\prec \frac{1}{2} \| x - b \|^2_2 - \frac{\mu}{\beta} \sum_{i=1}^{m} \log(-\varphi_i(x)) \rightarrow +\infty, \| x \|_2 \rightarrow +\infty$

Proof of Lemma 9. We denote $\phi : C_\prec \rightarrow \mathbb{R}$ by

$$\phi(x) = \frac{1}{2} \| x - b \|^2_2 - \frac{\mu}{\beta} \sum_{i=1}^{m} \log(-\varphi_i(x))$$

let $B(x) = -\frac{\mu}{\beta} \sum_{i=1}^{m} \log(-\varphi_i(x))$. We choose an arbitrary $x_0 \in C_\prec$. Then by the convexity of $B(x)$ we deduce that

$$\forall x \in C_\prec, \phi(x) \geq \frac{1}{2} \| x - b \|^2_2 + B(x_0) + \nabla B(x_0)^\top (x - x_0) \rightarrow +\infty, \| x \|_2 \rightarrow +\infty$$

□
Next, we prove Proposition 3.

**Proof of Proposition 3.** We denote $\phi : C_\prec \to \mathbb{R}$ by:

$$
\phi(x) = \frac{1}{2} \|x - y_k + \frac{1}{\beta}(F(w) + \lambda_k)\|^2_2 - \frac{\mu}{\beta} \sum_{i=1}^{m} \log(-\varphi_i(x))
$$

We choose $x_0 \in C_\prec$. By Lemma 9, $\forall x \in C_\prec$, $\phi(x) \to +\infty$, $\|x\|_2 \to +\infty$.

So there exists $M > 0$ such that $x_0 \in B(0, M)$ and $\forall x \in S$, $\phi(x) \leq \phi(x_0)$, $x$ must belong to $B(0, M)$, where

$$
B(0, M) = \{ x \in \mathbb{R}^n | \|x\| \leq M \}
$$

It’s clear that there exists $t > 0$ such that for every $x \in C_\prec$ that satisfies $\phi(x) \leq \phi(x_0)$, $x$ must belong to $C_t$, where

$$
C_t = \{ x \in B(0, M) | \varphi_i(x) \leq -t \} \subset C_\prec \quad (36)
$$

And we can make $t$ small enough so that $x_0 \in C_t$. $C_t$ is a nonempty compact set and $\phi$ is continuous, so there exists $x^* \in C_t$ such that $\phi(x^*) \leq \phi(x)$, $\forall x \in C_\prec$.

Note that $\forall x \in C_\prec \setminus C_t$, $\phi(x) \geq \phi(x_0) \geq \phi(x^*)$. Therefore, $x^*$ is a global minimizer of $\phi$. $\phi$ is strongly-convex thus $x_{k+1} = x^*$ is its unique minimizer. So $x = x_{k+1}$ if and only if $\nabla \phi(x) = 0$.

Therefore, $x_{k+1}$ is the unique solution of $x - y_k + \frac{1}{\beta}(F(w) + \lambda_k) - \frac{\mu}{\beta} \sum_{i=1}^{m} \nabla \varphi_i(x) = 0$. \qed
D Details on the implementation

In this section, we provide the details on the implementation of the experiments shown in the main part in 2D and higher dimension bilinear game, see § D.1 and § D.2, respectively. We also provide here in § D.3 the details of the MNIST experiments presented later in App. E. In addition, we provide the source code through the following link: https://github.com/Chavdarova/ACVI

D.1 Experiments in 2D

We first state the considered problem fully, then describe the setup what includes the hyperparameters.

Problems. We consider the following constrained bilinear game (for the same experiment shown in Fig. 1 and 4):

\[ \min_{x_1 \in \mathbb{R}^+} \max_{x_2 \in \mathbb{R}^+} 0.05 x_1^2 + x_1 x_2 - 0.05 x_2^2. \]  

(cBG)

The Von Neumann’s ratio game [Von Neumann, 1971] (results in Fig. (a)) is as follows:

\[ \min_{x} \max_{y} \langle x, R y \rangle, \]  

\[ \text{s.t.} \quad x, y \in \Delta^2, \]  

\[ R = \begin{pmatrix} -0.6 & -0.3 \\ 0.6 & -0.3 \end{pmatrix} \quad \text{and} \quad S = \begin{pmatrix} 0.9 & 0.5 \\ 0.8 & 0.4 \end{pmatrix}. \]

The so called Forsaken [Hsieh et al., 2021] game—used in Fig. 2(b) and in App. E—is as follows:

\[ \min_{x_1 \in X_1} \max_{x_2 \in X_2} x_1 (x_2 - 0.45) + h(x_1) - h(x_2), \]  

(Forsaken)

where \( h(z) = \frac{1}{4} z^2 - \frac{1}{2} z^4 + \frac{1}{6} z^6 \). The original version is unconstrained \( X = \mathbb{R}^2 \). In Fig. 2(b) we use the constraint \( x_1^2 + x_2^2 \leq 4 \), and in App. E we use two other constraints: \( x_1 \geq 0.08 \) and \( x_2 \geq 0.4 \).

For the toy GAN experiments shown in Fig. 2(c), the problem is as follows:

\[ \min_{\theta} \max_{\varphi} \mathbb{E}_{x \sim \mathcal{N}(0,2)} (\varphi x^2) - \mathbb{E}_{z \sim \mathcal{N}(0,1)} (\varphi \theta^2 z^2), \]  

\[ \text{s.t.} \quad \varphi^2 + \theta^2 \leq 4. \]  

(toy-GAN)

In the experiment, we look at a finite sum (sample average) approximation, which we then solve deterministically in a full batch fashion.

Setup. For all the 2D problems, we set the step size of GDA, EG and OGDA to 0.1, we use \( k = 5 \) and \( \alpha = 0.5 \) for LA-GDA, we set \( \beta = 0.08, \mu_{-1} = 10^{-5}, \delta = 0.5 \) and \( \lambda_0 = 0 \) for ACVI; and run for 50 iterations. For ACVI, we set the number of outer loop \( T = 20 \). In the first 19 outer loop iterations, we only run one inner loop iteration, and in the last outer loop iteration, we run 30 inner loop iterations (for total of 50 updates). In Fig. 1 and Fig. 2(b), we set the starting point to be \( (0.5, 0.5)^T \) for all algorithms. In Fig. 2(a) and (b), we set the starting point to be \( (0.5, 0.5)^T \) for all algorithms.

D.2 High-dimension Bilinear Game

We set the step size of GDA, EG and OGDA to 0.1, use \( k = 4 \) and \( \alpha = 0.5 \) for LA-GDA. For ACVI, we set \( \beta = 0.5, \mu_{-1} = 10^{-6}, \delta = 0.5 \) and \( \lambda_0 = 0 \) for ACVI.

The solution of (HBG) is \( x^* = \frac{1}{5000} e \), where \( e \in \mathbb{R}^{1000} \). As a metric of the experiments on this problem we use the relative error: \( \varepsilon_r(x_k) = \frac{\|x_k - x^*\|}{\|x^*\|} \). In Fig. 3(b), we set \( \varepsilon = 0.02 \) and compute number of iterations of ACVI needed to reach the relative error given different rotation “strength” \( \eta \), \( \eta \in (0, 1) \). Here we set the maximum numbers of iterations to be 50 for all algorithms. In Fig. 3(a), we set \( \eta = 0.05 \) and compute CPU times needed for ACVI, EG, OGDA and LA4-GDA to reach different relative errors. Here we set the maximum run time to be 1500 seconds for all algorithms. In Fig. 6 in App. E on the other hand, we fix \( \eta = 0.05 \), and for varying CPU time limits, we compute the relative error of the last iterates of ACVI, GDA, EG, OGDA and LA4-GDA.
Table 1: DCGAN architectures [Radford et al., 2016] used for experiments on MNIST. With “conv.” we denote a convolutional layer and “transposed conv” a transposed convolution layer [Radford et al., 2016]. We use ker and pad to denote kernel and padding for the (transposed) convolution layers, respectively. With $h \times w$ we denote the kernel size. With $c_{in} \rightarrow c_{out}$ we denote the number of channels of the input and output, for (transposed) convolution layers. The models use Batch Normalization [Ioffe and Szegedy, 2015] layers.

### D.3 MNIST experiments

For the experiments on the MNIST dataset, we use the source code of Chavdarova et al. [2021b] for the baselines and we build on it to implement ACVI. For completeness, we provide an overview of the implementation.

#### Models

We used the DCGAN architectures [Radford et al., 2016], listed in Table 1, and the parameters of the models are initialized using PyTorch default initialization. For experiments on this dataset, we used the non saturating GAN loss as proposed in [Goodfellow et al., 2014]:

$$
\mathcal{L}_D = \mathbb{E}_{\tilde{x}_d \sim p_d} \log (D(\tilde{x}_d)) + \mathbb{E}_{\tilde{z} \sim p_z} \log (1 - D(G(\tilde{z}))) \tag{L-D}
$$

$$
\mathcal{L}_G = \mathbb{E}_{\tilde{z} \sim p_z} \log (D(G(\til{z}))), \tag{L-G}
$$

where $G(\cdot), D(\cdot)$ denote the generator and discriminator, resp., and $p_d$ and $p_z$ denote the data and the latent distributions (the latter predefined as normal distribution).

#### Details on the ACVI implementation

When implementing ACVI on MNIST, we remove the outer loop of Algorithm 1 and fix $\mu$ to be a small number, in particular, we select $\mu = 10^{-9}$. We add two constraints for the MNIST experiment: we set the squared sum of all parameters of the Generator and the Discriminator (separately) to be less than or equal to a hyperparameter $M$. We select some large number for $M$, in particular, we set $M = 50$. We randomly initialize $x$ and $y$ and initialize $\lambda$ to zero, $\lambda = 0$. For line 8 and 9 of Algorithm 1, we run $l$ steps of (unconstrained) EG and GDA, respectively. For the update of $x$ (using EG), we use step-size $\eta_x = 0.001$, whereas for $y$ (using GDA), we use step-size $\eta_y = 0.2$. We present results when $l \in \{1, 10\}$. At every iteration, we update $\lambda$ using the expression in line 11 of Algorithm 1, with $\beta = 0.5$.

#### Metrics

We describe the metrics for the MNIST experiments shown later in App. E. We use the two standard GAN metrics, Inception Score (IS, Salimans et al., 2016) and Fréchet Inception Distance (FID, Heusel et al., 2017). Both FID and IS rely on a pre-trained classifier, and take finite set of $\tilde{m}$ samples from the generator to compute these. Since MNIST has greyscale images, we used a classifier trained on this dataset and used $\tilde{m} = 5000$.

#### Metrics: IS

Given a sample from the generator $\tilde{x}_g \sim p_g$—where $p_g$ denotes the data distribution of the generator—IS uses the softmax output of the pre-trained network $p(y|\tilde{x}_g)$ which represents the probability that $\tilde{x}_g$ is of class $c_i, i \in 1 \ldots C$, i.e., $p(y|\tilde{x}_g) \in [0, 1]^C$. It then computes the marginal

---

*Provided under Creative Commons Attribution-Share Alike 3.0.
*Provided at: [https://github.com/Chavdarova/LAGAN-Lookahead_Minimax](https://github.com/Chavdarova/LAGAN-Lookahead_Minimax)
class distribution \( p(\tilde{y}) = \int_{\tilde{x}} p(\tilde{y}|\tilde{x}) p(\tilde{x}) \). IS measures the Kullback–Leibler divergence \( \mathbb{D}_{KL} \) between the predicted conditional label distribution \( p(\tilde{y}|\tilde{x}) \) and the marginal class distribution \( p(\tilde{y}) \). More precisely, it is computed as follows:

\[
IS(G) = \exp \left( \mathbb{E}_{\tilde{x}_g \sim p_g} \left[ \mathbb{D}_{KL}(p(\tilde{y}|\tilde{x}_g)||p(\tilde{y})) \right] \right) = \exp \left( \frac{1}{m} \sum_{i=1}^{m} \sum_{c=1}^{C} p(y_c|\tilde{x}_i) \log \frac{p(y_c|\tilde{x}_i)}{p(y)} \right). \tag{IS}
\]

It aims at estimating (i) if the samples look realistic i.e., \( p(\tilde{y}|\tilde{x}_g) \) should have low entropy, and (ii) if the samples are diverse (from different ImageNet classes) i.e., \( p(\tilde{y}) \) should have high entropy. As these are combined using the Kullback–Leibler divergence, the higher the score is, the better the performance.

**Metrics: FID.** Contrary to IS, FID compares the synthetic samples \( \tilde{x}_g \sim p_g \) with those of the training dataset \( \tilde{x}_d \sim p_d \) in a feature space. The samples are embedded using the first several layers of a pretrained classifier. It assumes \( p_g \) and \( p_d \) are multivariate normal distributions, and estimates the means \( m_g \) and \( m_d \) and covariances \( C_g \) and \( C_d \), respectively for \( p_g \) and \( p_d \) in that feature space. Finally, FID is computed as:

\[
\mathbb{D}_{FID}(p_d, p_g) \approx \mathcal{D}_2((m_d, C_d), (m_g, C_g)) = \|m_d - m_g\|^2 + Tr(C_d + C_g - 2(C_dC_g)^{\frac{1}{2}}), \tag{FID}
\]

where \( \mathcal{D}_2 \) denotes the Fréchet Distance. Note that as this metric is a distance, the lower it is, the better the performance.

**Hardware.** We used the Colab platform (https://colab.research.google.com/) and Tesla P100 GPUs. The running times are reported in App. E.
E Additional Empirical Analysis

In this section, we provide omitted plots of the main paper—including the remaining methods of Fig. 1, different analysis on the HBG problem (see App. E.1), as well as additional experiments on MNIST (see App. E.2).

E.1 Additional experiments in 2D and on HBG

While Fig. 1 lists solely EG and ACVI, Fig. 4 depicts all the considered baselines on the cBG problem for completeness.

Figure 4: In addition to Fig. 1, here we depict all the considered baselines on the cBG problem. The solution at $(0, 0)$ is denoted with red star (⋆), and the vector field of this problem with gray arrows. See App. D.1 for details on the implementation.

Figure 5: Forsaken game with different constraints: we consider two additional (to that in Fig. 2) constraints: (a) that $x_1 \geq 0.08$, and (b) that $x_2 \geq 0.4$. See App. D.1 for details on the implementation, and App. E.1 for discussion.

Fig. 5 depicts the baseline methods and ACVI on the Forsaken problem with two different constraints than that considered in Fig. 2 (that $x_1^2 + x_2^2 \leq 4$). Since this game is non-monotone, we observe that for some constraints the baseline methods—GDA, EG, OGDA, LA4-GDA—stay near the constraint (and do not converge). This indicates that ACVI may have better chances of converging for broader classes of problems than monotone VIs, relative to baseline methods whose convergence may depend on the constraints, and when hitting a constraint may be significantly slower (as Fig. 1 illustrates).
Similar to Fig. 3, in Fig. 6 we run experiments on the HBG problem. However, here for a given fixed CPU time, we depict the relative error of the considered baselines and ACVI.

Figure 6: Given varying CPU time (in seconds), depicting the relative error (see App. D.2) of GDA, EG, OGDA, LA4-GDA, and ACVI (Algorithm 1) on the HBG problem. In addition to Fig. 3 in the main paper. See App. D.2 for details on the implementation.

E.2 Experiments on MNIST

In this section, we consider the MNIST dataset, which is an unconstrained problem. We augment the problem with a mild constraint which requires that the norm of the per-player parameters does not exceed certain value (see App. D.3). The primary purpose of these experiments is to observe if Algorithm 1 is competitive computationally-wise when lines 8 and 9 are non-trivial and require (unconstrained) solver. However note that since MNIST is a relatively easy problem, it may not answer the natural question if ACVI has advantages on “augmentedly-constrained” problems over standard methods. Thus, we leave this for future work. The implementation and the used metrics are described in App. D.3.

Fig. 7 summarizes the experiments in terms of the obtained FID score over time. We observe that ACVI (although it uses two solvers at each iteration) is yet performing competitively to GDA and EG. Figures 8–13 provide in addition samples of the Generator and IS scores, separately for each method. We believe that further exploring the type of constraints to be added, or the implementation options (e.g., l, step-size) may be proven fruitful even for problems that are originally unconstrained.
Figure 7: Summary of the experiments on MNIST, using FID (lower is better). 7(a): GDA and ACVI with GDA, and 7(b): EG and ACVI with EG, using $\ell = \{1, 10\}$ for ACVI. Using step size of 0.001. See App. D.3 and E.2 for details on the implementation and discussion, resp.

Figure 8: GDA on MNIST. Left: samples $\tilde{x}_g \sim p_g$ of the last iterate of the Generator. Right: FID and IS of GDA, depicted in blue and red, respectively. Using step size of 0.001.

Figure 9: ACVI (Algorithm 1) with 10 GDA steps on MNIST. Left: samples $\tilde{x}_g \sim p_g$ of the last iterate of the Generator. Right: FID and IS of GDA, depicted in blue and red, respectively. Using step size of 0.001 for $x$ and 0.2 for $y$, and $\ell = 10$ both for $x$ and $y$. 
Figure 10: ACVI (Algorithm 1) with 1 GDA step on MNIST. Left: samples \( \tilde{x}_g \sim p_g \) of the last iterate of the Generator. Right: FID and IS of GDA, depicted in blue and red, respectively. Using step size of 0.001 for \( x \) and 0.2 for \( y \), and \( l = 1 \) both for \( x \) and \( y \).

Figure 11: EG on MNIST. Left: samples \( \tilde{x}_g \sim p_g \) of the last iterate of the Generator. Right: FID and IS of GDA, depicted in blue and red, respectively. Using step size of 0.001.

Figure 12: ACVI (Algorithm 1) with 10 EG steps on MNIST. Left: samples \( \tilde{x}_g \sim p_g \) of the last iterate of the Generator. Right: FID and IS of GDA, depicted in blue and red, respectively. Using step size of 0.001 for \( x \) and 0.2 for \( y \), and \( l = 10 \) both for \( x \) and \( y \).
Figure 13: ACVI (Algorithm 1) with 1 EG step on MNIST. Left: samples $\tilde{x}_g \sim p_g$ of the last iterate of the Generator. Right: FID and IS of GDA, depicted in blue and red, respectively. Using step size of 0.001 for $x$ and 0.2 for $y$, and $l = 1$ both for $x$ and $y$. 