Linearly Constrained Bilevel Optimization: A Smoothed Implicit Gradient Approach

Prashant Khanduri [*1] **Ioannis Tsaknakis** [*2] **Yihua Zhang** [3] **Jia Liu** [4] **Sijia Liu** [3] **Jiawei Zhang** [5] **Mingyi Hong** [2]

Abstract

This work develops analysis and algorithms for solving a class of bilevel optimization problems where the lower-level (LL) problems have linear constraints. Most of the existing approaches for constrained bilevel problems rely on value function-based approximate reformulations, which suffer from issues such as non-convex and non-differentiable constraints. In contrast, in this work, we develop an implicit gradient-based approach, which is easy to implement, and is suitable for machine learning applications. We first provide an in-depth understanding of the problem, by showing that the implicit objective for such problems is in general non-differentiable. However, if we add some small (linear) perturbation to the LL objective, the resulting implicit objective becomes differentiable almost surely. This key observation opens the door for developing (deterministic and stochastic) gradient-based algorithms similar to the state-of-the-art ones for unconstrained bi-level problems. We show that when the implicit function is assumed to be strongly-convex, convex, and weakly-convex, the resulting algorithms converge with guaranteed rate. Finally, we experimentally corroborate the theoretical findings and evaluate the performance of the proposed framework on numerical and adversarial learning problems.

[*]Equal contribution [1]Department of CS, Wayne State University, Detroit, MI 48202, USA [2]Department of ECE, University of Minnesota, Minneapolis, MN 55455, USA [3]Department of CSE, Michigan State University, East Lansing, MI 48824, USA [4]Department of ECE, The Ohio State University, Columbus, OH 43210, USA [5]Laboratory for Information & Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Correspondence to: Ioannis Tsaknakis <tsakn001@umn.edu>, Prashant Khanduri <khanduri.prashant@wayne.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

1. Introduction

Bilevel optimization problems (Colson et al., 2005; Dempe & Zemkoho, 2020) can be used to model an important class of hierarchical optimization tasks with two levels of hierarchy, the *upper-level (UL)* and the *lower-level (LL)*. The key characteristics of bilevel problems are: 1) the solution of the UL problem requires access to the solution of the LL problem and, 2) the LL problem is parametrized by the UL variable. Bilevel optimization problems arise in a wide range of machine learning applications, such as meta-learning (Rajeswaran et al., 2019; Franceschi et al., 2018), data hypercleaning (Shaban et al., 2019), hyperparameter optimization (Sinha et al., 2020; Franceschi et al., 2018; 2017; Pedregosa, 2016), adversarial learning (Li et al., 2019; Liu et al., 2021a; Zhang et al., 2022), as well as in other application domains such as network optimization (Migdalas, 1995), economics (Cecchini et al., 2013), and transport research (Didi-Biha et al., 2006; Kalashnikov et al., 2010). In this work, we focus on a special class of stochastic bilevel optimization problems, where the LL problem involves the minimization of a *strongly convex* objective over a set of *linear inequality constraints*. More precisely, we consider the following formulation:

$$\min_{\mathbf{x}\in\mathcal{X}}\Big\{G(\mathbf{x}) \coloneqq f(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x})) \coloneqq \mathbb{E}_\xi[\widetilde{f}(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x});\xi)]\Big\}, \tag{1a}$$

$$\text{s.t. } \overline{\mathbf{y}}^*(\mathbf{x}) \in \arg\min_{\mathbf{y}\in\mathbb{R}^{d_\ell}}\Big\{h(\mathbf{x},\mathbf{y}) \;\big|\; A\mathbf{y}\le\mathbf{b}\Big\}, \tag{1b}$$

where $\xi\sim\mathcal{D}$ represents a stochastic sample of the objective $f(\cdot,\cdot)$, $\mathcal{X}\subseteq\mathbb{R}^{d_u}$ is a convex and closed set, $f:\mathcal{X}\times\mathbb{R}^{d_\ell}\to\mathbb{R}$ is the UL objective, $h:\mathcal{X}\times\mathbb{R}^{d_\ell}\to\mathbb{R}$ is the LL objective, and f,h are smooth functions; note that the UL objective is stochastic, while the LL one is not. We focus on the problems where $h(\mathbf{x},\mathbf{y})$ is strongly convex with respect to $\mathbf{y}$. The matrices $A\in\mathbb{R}^{k\times d_\ell}$, $B\in\mathbb{R}^{k\times d_u}$ and vector $\mathbf{b}\in\mathbb{R}^k$ define the linear constraints. In the following, we refer to (1a) as the UL problem, and to (1b) as the LL one.

The success of the bilevel formulation and its algorithms in many machine learning applications can be attributed to the use of the efficient (stochastic) gradient-based methods (Liu et al., 2021a). These methods take the following form, in which an (approximate) gradient direction of the UL

problem is computed (using chain rule), and then the UL variable is updated using gradient descent (GD):

$$\widehat{\nabla} G(\mathbf{x}) \approx \nabla_x f(\mathbf{x}, \overline{\mathbf{y}}^*(\mathbf{x})) + [\nabla \overline{\mathbf{y}}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \overline{\mathbf{y}}^*(\mathbf{x}))$$
$$\texttt{GD Update}: \mathbf{x}^+ = \mathbf{x} - \beta \widehat{\nabla} G(\mathbf{x}).$$

The gradient of $G(\mathbf{x})$ is often referred to as the *implicit gradient*. However, computing this implicit gradient not only requires access to the optimal $\overline{\mathbf{y}}^*(\mathbf{x})$, but also assumes differentiability of the mapping $\overline{\mathbf{y}}^*(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^{d_\ell}$. One can potentially solve the LL problem approximately and obtain an approximation $\widehat{\mathbf{y}}(\mathbf{x})$ such that $\widehat{\mathbf{y}}(\mathbf{x}) \approx \overline{\mathbf{y}}^*(\mathbf{x})$, and use it to compute the implicit gradient (Ghadimi & Wang, 2018). Unfortunately, not all solutions $\mathbf{y}^*(\mathbf{x})$ are differentiable, and when they are not the above approach cannot be applied.

It is known that when the LL problem is strongly convex and *unconstrained*, then $\nabla \overline{\mathbf{y}}^*(\mathbf{x})$ can be easily evaluated using the implicit function theorem (Ghadimi & Wang, 2018). This is the reason that the majority of recent works have focused on developing algorithms for the class of unconstrained bilevel problems (Ghadimi & Wang, 2018; Hong et al., 2023; Ji et al., 2021; Khanduri et al., 2021b; Chen et al., 2021a). However, when the LL problem is constrained, $\nabla \overline{\mathbf{y}}^*(\mathbf{x})$ might not even exist. In that case, most works adopt a value function-based approach to solve problems with LL constraints (Liu et al., 2021b; Sow et al., 2022; Liu et al., 2021c). Value-function-based methods typically transform the original problem into a single-level problem with non-convex and non-differentiable constraints. To resolve the latter issue these approaches regularize the problem by adding a strongly-convex penalty term, altering the problem's structure. In contrast, we introduce a perturbation-based smoothing technique, which at any given $\mathbf{x} \in \mathcal{X}$ makes $\overline{\mathbf{y}}^*(\mathbf{x})$ differentiable almost surely, without practically changing the landscape of the original problem (see Lu et al. (2020, pg. 5)). It is important to note that the value function-based approaches are more suited for deterministic implementations, and therefore it is difficult to use such algorithms for large scale applications and/or when the data sizes are large. On the other hand, the gradient-based algorithms developed in our work can easily handle stochastic problems. Finally, there is a line of work (Amos & Kolter, 2017; Agrawal et al., 2019; Donti et al., 2017; Gould et al., 2021) about implicit differentiation in deep learning literature. However, in these works the setting (e.g. layers of neural network described by optimization tasks) and the focus (e.g., on gradient computation and implementation, rather than on algorithms and analysis) is different. For more details see Appendix A. Below, we list the major contributions of our work.

Contributions. In this work, we study a class of bilevel optimization problems with strongly convex objective and linear constraints in the LL. Major challenges for solving such problems are the following: 1) How to ensure that the implicit function $G(\mathbf{x})$ is differentiable? and 2) Even if the implicit function is differentiable, how to compute its (approximate) gradient in order to develop first-order methods? Our work addresses these challenges and develops first-order methods to tackle such constrained bilevel problems. Specifically, our contributions are the following:

- We provide an in-depth understanding of bilevel problems with strongly convex linearly constrained LL problems. Specifically, we first show with an example that the implicit objective $G(\mathbf{x})$ is in general non-differentiable. To address the non-differentiability, we propose a perturbation-based smoothing technique that makes the implicit objective $G(\mathbf{x})$ differentiable in an almost sure sense, and we provide a way to compute the (approximate) implicit gradient that involves a closed-form expression and an (approximate) solution of the LL problem.
- The smoothed problem we obtain is challenging, since its implicit objective does not have Lipschitz continuous gradients. Therefore, conventional gradient based algorithms may no longer work. To address this issue, we propose the Deterministic Smoothed Implicit Gradient ([D]SIGD) method that utilizes an (approximate) line search-based algorithm and establish asymptotic convergence guarantees. We also analyze [S]SIGD for the stochastic version of problem (1) and establish finite-time convergence guarantees in a neighborhood around a stationary solution for the cases when the implicit function is weakly-convex, strongly-convex, and convex (but not Lipschitz smooth).
- Finally, we evaluate the performance of the proposed algorithmic framework via experiments on quadratic bilevel and adversarial learning problems.

The bilevel problem (1) captures several important applications. Below we provide two such applications.

Adversarial Training. The problem of robustly training a model $\phi(\mathbf{x}; \mathbf{c})$, where $\mathbf{x}$ denotes the model parameters and $\mathbf{c}$ the input to the model; let $\{(\mathbf{c}_i, \mathbf{d}_i)\}_{i=1}^N$ with $\mathbf{c}_i \in \mathbb{R}^{d_{\ell_i}}, \mathbf{d}_i \in \mathbb{R}$ be the training set (Zhang et al., 2022; Goodfellow et al., 2014). It can be formulated as the following bilevel problem:

$$\min_{\mathbf{x} \in \mathbb{R}^{d_u}} \left\{ \sum_{i=1}^N f_i(\phi(\mathbf{x}; \mathbf{c}_i + \overline{\mathbf{y}}_i^*(\mathbf{x})), \mathbf{d}_i) \right\} \tag{2}$$
$$s.t.\ \overline{\mathbf{y}}^*(\mathbf{x}) \in \begin{cases} \arg\min_{\mathbf{y}_i \in \mathbb{R}^{d_{\ell_i}}} \sum_{i=1}^N h_i(\phi(\mathbf{x}; \mathbf{c}_i + \mathbf{y}_i), \mathbf{d}_i) \\ \text{s.t.} \quad -\mathbf{b} \le \mathbf{y} \le \mathbf{b} \end{cases}$$

where $\mathbf{y} = [\mathbf{y}_1^T, \ldots, \mathbf{y}_N^T]^T \in \mathbb{R}^{d_\ell}$; with $\mathbf{y}_i \in \mathbb{R}^{d_{\ell_i}}$ denotes the attack on the ith example and we have $\sum_{i=1}^N d_{\ell_i} = d_\ell$.

Moreover, $f_i : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ denotes the loss function for learning the model parameter $\mathbf{x}$, while $h_i : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ denotes the adversarial objective used to design the optimal attack $\mathbf{y}$. Note that the linear constraints in the LL problem $-\mathbf{b} \leq \mathbf{y} \leq \mathbf{b}$ models the attack budget.

Distributed Optimization. In distributed optimization (Chang et al., 2020; Yang et al., 2019), a set of N agents aim to jointly minimize an objective function $G(\mathbf{x})$ over an undirected graph $\mathcal{G} = (V, E)$. We consider the following distributed bilevel problem

$$\min_{\{\mathbf{x}_i \in \mathcal{X} | A\mathbf{x}=0\}} \Big\{ G(\mathbf{x}) := \sum_{i=1}^{N} f_i(\mathbf{x}_i, \overline{\mathbf{y}}_i^*(\mathbf{x}_i)) \Big\}$$
$$\text{s.t. } \overline{\mathbf{y}}^*(\mathbf{x}) \in \arg\min_{\mathbf{y} \in \mathbb{R}^{d_\ell}} \Big\{ \sum_{i=1}^{N} h_i(\mathbf{x}_i, \mathbf{y}_i) \text{ s.t. } A\mathbf{y} = \mathbf{0} \Big\},$$

where $\mathbf{x} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and $\mathbf{y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$. Each agent $i \in [N]$ has access to f_i and h_i. The constraint $A\mathbf{y} = 0$ (resp. $A\mathbf{x} = 0$) is introduced to ensure the consensus of LL (resp. UL) variables. Such problems arise in signal processing and sensor networks (Yousefian, 2021). This formulation also models a decentralized meta learning problem where the training and validation data is distributed among agents while each agent aims to solve the meta learning problem globally (Ji et al., 2021).

2. Properties and Implicit Gradient of (1)

2.1. Preliminaries

In this section we study the properties of problem 1. First, let us define the necessary notations. Let $\overline{A}(\mathbf{y})$ be the matrix that contains the rows $S(\mathbf{y}) \subseteq \{1, \ldots, k\}$ of A that correspond to the active constraints of inequality $A\mathbf{y} \leq \mathbf{b}$ in the LL problem, that is we have $\overline{A}(\mathbf{y})\mathbf{y} = \overline{\mathbf{b}}(\mathbf{y})$, where $\overline{\mathbf{b}}(\mathbf{y})$ contains the elements of $\mathbf{b}$ with indices in $S(\mathbf{y})$. Also, we denote with $\overline{\boldsymbol{\lambda}}^*(\mathbf{x})$ the Lagrange multipliers vector that corresponds to the active constraints at $\overline{\mathbf{y}}^*(\mathbf{x})$. Next, we introduce some basic assumptions.

Assumption 2.1. We assume that the following conditions hold for problem (1):

(a) $f(\mathbf{x}, \mathbf{y})$ is continuously differentiable, and $h(\mathbf{x}, \mathbf{y})$ is twice continuously differentiable.

(b) $\mathcal{X}$ is closed and convex; $\mathcal{Y} = \{\mathbf{y} \in \mathbb{R}^{d_\ell} \mid A\mathbf{y} \leq \mathbf{b}\}$ is a compact set.

(c) $h(\mathbf{x}, \mathbf{y})$ is μ_h-strongly convex in $\mathbf{y}$, for every $\mathbf{x} \in \mathcal{X}$.

(d) There exists $\mathbf{y} \in \mathbb{R}^{d_\ell}$ such that $A\mathbf{y} < \mathbf{b}$.

(e) $\overline{A}(\overline{\mathbf{y}}^*(\mathbf{x}))$ is full row rank, for every $\mathbf{x} \in \mathcal{X}$[1].

[1] This is the LICQ condition (Bertsekas, 1998) of the LL problem. It is used to ensure that the optimal solutions satisfy the KKT conditions.

The Assumptions 2.1(a), (b) and (c) are standard assumptions in bilevel optimization literature and are required to ensure the continuity of the implicit function (Proposition 2.2). Assumption 2.1(c) ensures that the implicit function $G(\mathbf{x})$ is well defined as the LL problem returns a single point. Assumption 2.1(d) ensures strict feasibility of the LL problem, while Assumption 2.1(e) implies that the rows of A corresponding to the active constraints are linearly independent. Note that this assumption is necessary to ensure the differentiability of the implicit function (Lemma 2.3, 2.4). Also note that there are some special cases in which Assumption 2.1(e) is automatically satisfied. For instance, consider a problem where the LL problem has box constraints, i.e., $\mathbf{a} \leq \mathbf{y} \leq \mathbf{b}$. Then for any $\mathbf{y} \in \mathcal{Y}$ the only possible non-zero values in the matrix $\overline{A}(\mathbf{y})$ are $+1, -1$, and there is only one non-zero value at each column. Therefore, $\overline{A}(\mathbf{y})$ is full row rank. Next, we utilize the above assumptions to analyze the properties of mapping $\overline{\mathbf{y}}^*(\mathbf{x})$.

Proposition 2.2 (Appendix D.1.1)**.** *Under Assumption 2.1, the mapping $\overline{\mathbf{y}}^*(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^{d_\ell}$ and the implicit function $G(\mathbf{x})$ are both continuous.*

Proposition 2.2 ensures that $\overline{\mathbf{y}}^*(\mathbf{x})$ and $G(\mathbf{x})$ are both continuous. Now if we can ensure differentiability of $\overline{\mathbf{y}}^*(\mathbf{x})$, then we can implement a gradient-based update rule to solve (1). However, as the following example illustrates, $\overline{\mathbf{y}}^*(\mathbf{x})$ and thus $G(\mathbf{x})$ are not differentiable in general.

Example. Consider the following problem

$$\min_{x \in [0,1]} x + \overline{y}^*(x) \tag{3}$$
$$\text{s.t. } \overline{y}^*(x) \in \arg\min_{y \in \mathbb{R}} \{(y^2 - x^2)^2 \mid \sqrt{3/5} \leq y \leq 1\}. \tag{4}$$

The mapping $\overline{y}^*(x)$ is $\overline{y}^*(x) = x$, if $x \in [\sqrt{3/5}, 1]$, and $\overline{y}^*(x) = \sqrt{3/5}$, if $x \in [0, \sqrt{3/5})$. In Figure 1, we plot this mapping. Notice that at the point $\mathbf{x} = \sqrt{3/5}$ the mapping (and thus the implicit function) is non-differentiable. □

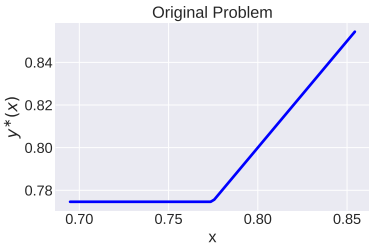


Figure 1: Plot of $\overline{y}^*(x)$.

To address the non-differentiability issue, let us introduce a perturbation-based "smoothing" technique, where for any fixed $\mathbf{x} \in \mathcal{X}$, the LL objective $h(\mathbf{x}, \mathbf{y})$ is augmented by an additional linear perturbation term $\mathbf{q}^T\mathbf{y}$, where $\mathbf{q}$ is a random vector sampled from some continuous distribution $\mathcal{Q}$. Specifically, let us define:

$$g(\mathbf{x}, \mathbf{y}) := h(\mathbf{x}, \mathbf{y}) + \mathbf{q}^T\mathbf{y} \text{ and} \tag{5}$$
$$\mathbf{y}^*(\mathbf{x}) := \arg\min_{\mathbf{y} \in \mathbb{R}^{d_\ell}} \{g(\mathbf{x}, \mathbf{y}) \mid A\mathbf{y} \leq \mathbf{b}\}. \tag{6}$$

Also, we denote $F(\mathbf{x}) := f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ as the respective "smoothed" implicit function. Such a perturbation is used to ensure that at a given $\mathbf{x} \in \mathcal{X}$, the strict complementarity (SC) property holds for the LL problem with probability 1 (w.p. 1); see the lemma below for the formal statement.

Lemma 2.3. *(Lu et al., 2020, Prop. 1) For a given $\mathbf{x} \in \mathcal{X}$, if $\mathbf{y}^*(\mathbf{x})$ is a KKT point of problem $\min_{\mathbf{y}\in\mathbb{R}^{d_\ell}} g(\mathbf{x}, \mathbf{y})$, $\mathbf{q}$ is generated from a continuous measure, and $\overline{A}(\mathbf{y}^*(\mathbf{x}))$ is full row rank, then the SC condition holds at $\mathbf{x}$, with probability 1 (w.p. 1), i.e., $\overline{A}(\mathbf{y}^*(\mathbf{x}))\mathbf{y}^*(\mathbf{x}) = \overline{\mathbf{b}}(\mathbf{y}^*(\mathbf{x})) \implies \overline{\boldsymbol{\lambda}}(\mathbf{y}^*(\mathbf{x})) > 0$ where $\overline{\boldsymbol{\lambda}}(\mathbf{y}^*(\mathbf{x}))$ is the corresponding vector of Lagrange multipliers.*

Combining SC ensured by Lemma 2.3 with Assumption 1, we can show that the implicit mapping $\mathbf{y}^*(\mathbf{x})$ is (almost surely) differentiable, which further implies that the implicit function $F(\mathbf{x})$ is differentiable at a given $\mathbf{x} \in \mathcal{X}$, and the corresponding gradient has a closed-form expression (please see Lemma 2.4 below). We would like to stress that the properties mentioned above (i.e., SC and differentiability) are defined locally, at a given point $\mathbf{x} \in \mathcal{X}$. These properties will be used later to design algorithms that approximately optimize the original problem (1). Finally, it is worth noting that, in the absence of such a perturbation term, we would have to introduce the SC property as an assumption, however since it is considered a strong assumption we choose to modify the problem instead such this property follows naturally.

2.2. Implicit Gradient

In this section, we derive a closed-form expression for the gradient of the implicit function $F(\mathbf{x})$.

Lemma 2.4 (Implicit Gradient, Appendix D.1.2)**.** *Under Assumption 2.1, for any given $\mathbf{x} \in \mathcal{X}$, we have*

$$\nabla \mathbf{y}^*(\mathbf{x}) = \big[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\big]^{-1} \cdot \big[-\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\big] \tag{7}$$

$$\nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) = -\big[\overline{A}\big[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\big]^{-1}\overline{A}^T\big]^{-1} \cdot \big[\overline{A}\big[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\big]^{-1}\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\big], \tag{8}$$

where we set $\overline{A} := \overline{A}(\mathbf{y}^*(\mathbf{x}))$.

Note that when LL problem (1b) does not have the LL constraints, the implicit gradient derived in Lemma 2.4 becomes exactly same as the one in Ghadimi & Wang (2018); Ji et al. (2021). Moreover, if the LL problem has only linear equality constraints, the differentiability of $\mathbf{y}^*(\mathbf{x})$ follows from the implicit function theorem under Assumptions 2.1(a) and 2.1(c) along with full row rankness of A. In fact, the expression of the implicit gradient stays the same as in Lemma 2.4 with $\overline{A}$ and $\overline{\boldsymbol{\lambda}}^*(\mathbf{x})$ replaced by A and $\boldsymbol{\lambda}^*(\mathbf{x})$, respectively (i.e., we use the full matrix A). Finally, using Lemma 2.4 we now have an expression of the implicit gradient as

$$\nabla F(\mathbf{x}) = \nabla_x f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) + [\nabla \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})). \tag{9}$$

From a practical perspective, the implicit gradient computation involves two parts, the (approximate) solution of the lower-level problem $\mathbf{y}^*(\mathbf{x})$ and the computation of the formulas (7), (8). For the solution of the LL problem there are several efficient solution methods, as it is a (strongly) convex optimization problem. For instance, we can solve it using a number of projected gradient descent steps; for more details about methods for solving the LL problem see appendix B. Furthermore, the computation of (7), (8) can be computationally intensive since these formulas involve Jacobians and Hessian inverses of the LL objective g. As these computations are also encountered in the unconstrained case (i.e., the LL problem has no constraints), we expect that in practice some of the known approximations from the relevant literature (Ghadimi & Wang, 2018; Hong et al., 2023) can also be applied in our case in order to reduce the complexity. For instance, we can use the Neumann series approximation for the Hessian inverse computation similar to the way it is used in the unconstrained case (e.g. see (Ghadimi & Wang, 2018, Lemma 3.1)).

2.2.1. Approximate Implicit Gradient

Note that computing $\nabla F(\mathbf{x})$ requires the precise knowledge of $\mathbf{y}^*(\mathbf{x})$ which is not possible for many problems of interest. Therefore, in practice we define the approximate implicit gradient as

$$\widehat{\nabla} F(\mathbf{x}) = \nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) + [\widehat{\nabla} \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})), \tag{10}$$

where $\widehat{\nabla} \mathbf{y}^*(\mathbf{x})$ is defined by setting the approximate LL solution $\widehat{\mathbf{y}}(\mathbf{x})$ in place of the exact one $\mathbf{y}^*(\mathbf{x})$ in expressions (7) and (8). In order to ensure that (10) returns a useful approximation of the (exact) implicit gradient, we impose a few assumptions on the quality of the estimate $\widehat{\mathbf{y}}(\mathbf{x})$.

Assumption 2.5. The approximate solution of (perturbed) LL problem (5) $\widehat{\mathbf{y}}(\mathbf{x})$ satisfies the following $\forall \mathbf{x} \in \mathcal{X}$:

(a) $\|\widehat{\mathbf{y}}(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\| \leq \delta$ for $\delta > 0$,

(b) $\widehat{\mathbf{y}}(\mathbf{x})$ is a feasible point, i.e., $A\widehat{\mathbf{y}}(\mathbf{x}) \leq \mathbf{b}$,

(c) It holds that $\overline{A}(\mathbf{y}^*(\mathbf{x})) = \overline{A}(\widehat{\mathbf{y}}(\mathbf{x}))$.

The LL problem requires the solution of a strongly convex linearly constrained task. As a result, Assumptions 2.5(a),(b) can be easily satisfied. Specifically, we can obtain approximate feasible solutions of given accuracy with known methods, such as projected gradient descent, or by using some convex optimization solver; in section B of the Appendix we provide one such method. Moreover, Assumption 2.5(c)

will be satisfied if we find a "sufficiently accurate" solution $\widehat{\mathbf{y}}(\mathbf{x})$. Specifically, from Calamai & Moré (1987, Theorem 4.1) we know that if $\widehat{\mathbf{y}}^k(\mathbf{x}) \in \mathcal{Y}$ is an arbitrary sequence that converges to a non-degenerate (i.e., Assumption 1(e) and SC holds) stationary solution $\mathbf{y}^*(\mathbf{x})$, then there exists an integer k_0 such that $\overline{A}(\mathbf{y}^*(\mathbf{x})) = \overline{A}(\widehat{\mathbf{y}}^k(\mathbf{x})), \forall k > k_0$.

Remark 2.6. There are certain special cases where we can obtain an upper bound for k_0. For instance, in the case of non-negative constraints $\mathbf{y} \geq \mathbf{0}$ it can be shown[2] that $\frac{L_h}{\mu_h} \log\left(2L_h\|\mathbf{y}^0 - \mathbf{y}^*(\mathbf{x})\|/\tau\right)$ iterations of the projected gradient descent method suffice to ensure that the active set of the approximate solution $\widehat{\mathbf{y}}(\mathbf{x})$ coincides with the active set of the exact one $\mathbf{y}^*(\mathbf{x})$ (see Nutini et al. (2019, Corollary 1)), where $\tau = \min_{i \in S(\mathbf{y}^*(\mathbf{x}))} \nabla_{y_i} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ and $\mathbf{y}^0$ is the algorithm's initialization. A similar result can be derived for the case with bound constraints $\mathbf{a} \leq \mathbf{y} \leq \mathbf{b}$.

Next, we introduce additional assumptions that are required to analyze the properties of (10).

Assumption 2.7. We assume that the following holds for problem (1), $\forall \mathbf{x}, \overline{\mathbf{x}} \in \mathcal{X}$ and $\mathbf{y}, \overline{\mathbf{y}} \in \mathbb{R}^{d_\ell}$:

(a) f has bounded gradients, i.e., $\|\nabla f(\mathbf{x}, \mathbf{y})\| \leq \overline{L}_f$.

(b) f has Lipschitz continuous gradients, i.e., $\|\nabla f(\mathbf{x}, \mathbf{y}) - \nabla f(\overline{\mathbf{x}}, \overline{\mathbf{y}})\| \leq L_f \|[\mathbf{x}; \mathbf{y}] - [\overline{\mathbf{x}}; \overline{\mathbf{y}}]\|$.

(c) h has Lipschitz continuous gradient in $\mathbf{y}$, i.e., $\|\nabla_y h(\mathbf{x}, \mathbf{y}) - \nabla_y h(\mathbf{x}, \overline{\mathbf{y}})\| \leq L_h \|\mathbf{y} - \overline{\mathbf{y}}\|$.

(d) h has Lipschitz continuous Hessian in $\mathbf{y}$, i.e., $\|\nabla^2_{yy} h(\mathbf{x}, \mathbf{y}) - \nabla^2_{yy} h(\mathbf{x}, \overline{\mathbf{y}})\| \leq L_{h_{yy}} \|\mathbf{y} - \overline{\mathbf{y}}\|$.

(e) h has Lipschitz continuous Jacobian, i.e., $\|\nabla^2_{xy} h(\mathbf{x}, \mathbf{y}) - \nabla^2_{xy} h(\mathbf{x}, \overline{\mathbf{y}})\| \leq L_{h_{xy}} \|\mathbf{y} - \overline{\mathbf{y}}\|$.

(f) h has a bounded Jacobian, $\|\nabla^2_{xy} h(\mathbf{x}, \mathbf{y})\| \leq \overline{L}_{h_{xy}}$.

Assumption 2.7 is standard in bilevel optimization literature (Ghadimi & Wang, 2018; Hong et al., 2023; Chen et al., 2021a; Ji et al., 2021) and is used to derive some useful properties of the (approximate) implicit gradient (Lemma 2.8, Appendix D.1.3). It is easy to see that Assumptions 2.1(a),(c) and 2.7 hold directly for the perturbed objective (5) with constants $\mu_g = \mu_h, L_h = L_g, L_{g_{yy}} = L_{h_{yy}}, L_{g_{xy}} = L_{h_{xy}}, \overline{L}_{g_{xy}} = \overline{L}_{h_{xy}}$; we also assume that Assumption 2.1(e) holds for the perturbed LL problem (5).

Lemma 2.8 (Appendix D.1.3)**.** *Suppose that Assumptions 2.1,2.5,2.7 hold. Then, for every $\mathbf{x} \in \mathcal{X}$ the following holds*

$$\|\widehat{\nabla} F(\mathbf{x}) - \nabla F(\mathbf{x})\| \leq L_F \cdot \delta$$
$$\|\nabla F(\mathbf{x})\| \leq \overline{L}_F \quad \textit{and} \quad \|\widehat{\nabla} F(\mathbf{x})\| \leq \overline{L}_F,$$

where $L_F = L_f + L_{\mathbf{y}^*}\overline{L}_f + L_f \overline{L}_{\mathbf{y}^*}$, and $\overline{L}_F = \left(1 + \overline{L}_{\mathbf{y}^*}\right)\overline{L}_f$; the constants $\overline{L}_{\mathbf{y}^*}, L_{\mathbf{y}^*}$ are defined in Lemmas D.7,D.9, respectively, provided in the Appendix.

[2] Assuming that $\nabla_{y_i} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) > 0, \forall i \in S(\mathbf{y}^*(\mathbf{x}))$.

2.2.2. Stochastic Implicit Gradient

In the stochastic setting, the (approximate) stochastic implicit gradient is computed as:

$$\widehat{\nabla} F(\mathbf{x}; \xi) = \nabla_x \widetilde{f}(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}); \xi) + [\widehat{\nabla} \mathbf{y}^*(\mathbf{x})]^T \nabla_y \widetilde{f}(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}); \xi). \tag{11}$$

Also, we make the following assumption on the stochastic gradients of the UL problem.

Assumption 2.9. We assume that the stochastic gradients are unbiased, i.e. $\mathbb{E}_\xi[\nabla \widetilde{f}(\mathbf{x}, \mathbf{y}; \xi)] = \nabla f(\mathbf{x}, \mathbf{y})$ and have bounded variance, i.e., $\mathbb{E}_\xi \|\nabla \widetilde{f}(\mathbf{x}, \mathbf{y}; \xi) - \nabla f(\mathbf{x}, \mathbf{y})\|^2 = \sigma_f^2$ for some $\sigma_f > 0$.

Assumption 2.9 is a typical assumption required to ensure that the approximate implicit stochastic gradient is also unbiased and has finite variance (Ghadimi & Wang, 2018; Hong et al., 2023; Chen et al., 2021a) as shown in Lemma 2.10 below.

Lemma 2.10 (Appendix D.1.4)**.** *Under Assumptions 2.1,2.5,2.7 and 2.9, the stochastic gradient estimate in* (11) *is unbiased, i.e.,*

$$\mathbb{E}_\xi[\widehat{\nabla} F(\mathbf{x}; \xi)] = \widehat{\nabla} F(\mathbf{x})$$

and has bounded variance , i.e.,

$$\mathbb{E}_\xi \|\widehat{\nabla} F(\mathbf{x}; \xi) - \widehat{\nabla} F(\mathbf{x})\| \leq \sigma_F^2$$

where $\sigma_F^2 = 2\sigma_f^2 + 2\overline{L}_{\mathbf{y}^*}\sigma_f^2$; where $\overline{L}_{\mathbf{y}^*}$ is defined in Lemma D.7 in the Appendix.

3. The SIGD Algorithms and Convergence

3.1. The Proposed Algorithms

In this section, we develop gradient-based methods for solving problem (1) by leveraging the smoothing-based technique introduced in the previous section. Recall that for any $\mathbf{x} \in \mathcal{X}$, we can introduce a perturbation to make the optimal solution $\mathbf{y}^*(\mathbf{x})$ of the perturbed LL problem differentiable (w.p. 1) by ensuring that the SC holds (w.p. 1). Notably, SC allows us to compute the implicit gradient in closed form as demonstrated in Lemma 2.4. Next, to proceed with the algorithm design, two choices are available for the perturbation $\mathbf{q}$. First, generate a perturbation for each $\mathbf{x} \in \mathcal{X}$ encountered in the algorithm. This approach will make the problem stochastic w.r.t. the perturbation $\mathbf{q}$ since sampling a $\mathbf{q}$ at each iteration would correspond to sampling a (biased) stochastic sample. The second option is to generate *a single* perturbation at the beginning of the algorithm and use it throughout the execution of the algorithm. It is worth mentioning that, both approaches perform equally well in our numerical experiments. For ease of analysis, we adopt the second approach. However, to justify such an approach,

Algorithm 1 [Deterministic] Smoothed Implicit Gradient Descent (**[D]SIGD**)

Input: Initial parameter $\mathbf{x}^0$, # of iteration T, LL solution accuracy, δ^r, σ, measure $\mathcal{Q}$
Sample $\mathbf{q} \sim \mathcal{Q}$ and perturb LL problem
for $r = 0, 1, \ldots, T-1$ **do**
  Find an approximate solution $\widehat{\mathbf{y}}(\mathbf{x}^r)$ s.t. Assumption 2.5 is satisfied.
  Compute $\widehat{\nabla} F(\mathbf{x}^r)$ using (10), $\widehat{\mathbf{d}}^r = \widetilde{\mathbf{x}}^r - \mathbf{x}^r$ with $\widetilde{\mathbf{x}}^r = \text{proj}_{\mathcal{X}}(\mathbf{x}^r - \widehat{\nabla} F(\mathbf{x}^r))$
  Select a^r s.t. the following Armijo-type rule condition is satisfied

$$\widehat{F}(\mathbf{x}^r) - \widehat{F}(\mathbf{x}^r + a^r \widehat{\mathbf{d}}^r) \geq -\sigma \cdot a^r [\widehat{\nabla} F(\mathbf{x}^r)]^T \widehat{\mathbf{d}}^r - \epsilon(\delta; r) \quad (12)$$

  where $\epsilon(\delta; r)$ depends on δ^r, α^r and problem-dependent parameters; $\widehat{F}(\cdot) = f(\cdot, \widehat{\mathbf{y}}(\cdot))$.
  Projected gradient step: $\mathbf{x}^{r+1} = \mathbf{x}^r + a^r \cdot \widehat{\mathbf{d}}^r$
end for

Algorithm 2 [Stochastic] Smoothed Implicit Gradient Descent (**[S]SIGD**)

Input: Initial parameter $\mathbf{x}^0$, # of iterations T, step-sizes $\{\beta^r\}_{r=0}^{T-1}$, LL solution accuracy δ
Sample $\mathbf{q} \sim \mathcal{Q}$ and perturb LL problem
for $r = 0, 1, \ldots, T-1$ **do**
  Find an approximate solution $\widehat{\mathbf{y}}(\mathbf{x}^r)$ s.t. Assumption 2.5 is satisfied.
  Compute $\widehat{\nabla} F(\mathbf{x}^r; \xi^r)$ using (11)
  Perform one stochastic projected gradient descent step: $\mathbf{x}^{r+1} = \text{proj}_{\mathcal{X}}(\mathbf{x}^r - \beta^r \widehat{\nabla} F(\mathbf{x}^r; \xi^r))$
end for

we need to establish that $F(\cdot)$ will be differentiable (w.p. 1) at the sequence of iterates encountered during the execution of the algorithm.

Towards this end, let us introduce some additional notations. Let us define $\bar{\mathcal{X}}_G := \{\bar{\mathbf{x}} : G(\mathbf{x}) \text{ is non-differentiable at } \bar{\mathbf{x}}\}$ as the set of non-differentiable points of the implicit function of the unperturbed problem (1a); define $\bar{\mathcal{X}}_F$ similarly. We denote $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}; \mathbf{q}) : \bar{\mathcal{X}} \times \mathbb{R}^{d_\ell} \to \mathbb{R}^{d_u}$ as the set mapping that maps the non-differentiable points of $G(\cdot)$, together with a given perturbation $\mathbf{q}$, to the non-differentiable points of $F(\cdot)$[3]. Next, we show that under certain assumptions on the sets $\bar{\mathcal{X}}_G$, $\bar{\mathcal{X}}_F$ and $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}; \mathbf{q})$ the iterates generated by a gradient-based algorithm will be differentiable w.p. 1.

Lemma 3.1 (Appendix D.2.1)**.** *Assuming that the set $\bar{\mathcal{X}}_G$ is countable and the mapping $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}; \cdot)$ is continuous. Further assume $\bar{\mathcal{X}}_F \subseteq \{\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}; \cdot) \mid \bar{\mathbf{x}} \in \bar{\mathcal{X}}_G\}$. Then the implicit function $F(\cdot)$ is differentiable at all the points $\{\mathbf{x}^r\}_{r=0}^T$ generated by a gradient-based algorithm w.p.* 1.

We note that the above lemma implies that even if the original function $G(\cdot)$ has a countable, but an infinite set of non-differentiable points, a gradient-based algorithm would generate a sequence of iterates that are differentiable w.p. 1. We remark that a key step in showing Lemma 3.1 is by exploring the structure of the linearly constrained bilevel problem. Specifically, observe that the active set in the expression of the implicit gradient is a *discrete random variable* as a function of $\mathbf{q}$ since the support of the active set is finite. This implies that the iterates generated by any gradient based algorithm will in general be mixed random variables (i.e., their CDF will be piecewise continuous). This fact combined with the assumption that $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}; \cdot)$ is continuous leads to Lemma 3.1.

Let us discuss the two assumptions in the above lemma. First, since the function $G(\cdot)$ is continuous, Rademacher's theorem implies that the set of non-differentiable points has zero measure. Although we cannot find sufficient conditions to guarantee that the set of non-differentiable points is countable, we believe that it is a reasonable assumption; see, e.g., Appendix D.2 for a simple example illustrating this. Second, the condition on the continuity of $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}; \cdot)$ is mild since $\mathbf{q}$ is a linear perturbation and naturally its magnitude will determine the perturbations in the non-differentiable points of the original function $G(\cdot)$. Please see Appendix D.2 for a simple example illustrating the continuity of $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}; \cdot)$.

Due to this result, in the following analysis, we assume that the iterates generated by our algorithms are differentiable almost surely. Further, our algorithm design is guided by the fact that unlike bilevel programs with unconstrained LL tasks (see Lemma 2.2(c) in (Ghadimi & Wang, 2018)), the implicit gradient $\nabla F(\mathbf{x})$ in (9) is not Lipschitz smooth in general. This implies that algorithms that provably converge only under the Lipschitz assumption, will not work in our case, particularly when the implicit function is non-convex. Towards this end, we propose the [Deterministic] Smoothed Implicit Gradient Descent ([D]SIGD) method (Alg. 1), a deterministic line-search-based method, which does not require Lipschitz smoothness or another special structure (e.g., convexity), and show asymptotic convergence (Theorem 3.3). Moreover, for the cases where the implicit function is weakly-convex, convex or strongly convex (but still not Lipschitz smooth) the [Stochastic] Smoothed Implicit Gradient Descent ([S]SIGD) method (Alg. 2) is developed, a stochastic gradient-based method, for which finite-time

[3] Lemma 2.3 states that given any $x \in \mathcal{X}$, adding a perturbation $\mathbf{q}$ makes the implicit function differentiable at that point (i.e., $\nabla G(\mathbf{x})$ exists w.p.1.). However, after a perturbation is added the perturbed implicit function $F(\cdot)$ might still have non-differentiable points, that depend on the non-differentiable points of $G(\cdot)$. Therefore, we define the set mapping $\bar{\boldsymbol{\epsilon}}$ as a function of both $\bar{x} \in \bar{\mathcal{X}}$ and $\mathbf{q}$. Please see Appendix D.2 for an example illustrating this.

convergence guarantees are derived (Theorem 3.6,3.8,3.9).

3.2. Convergence Analysis

As discussed above, in the context of algorithm design and analysis we sample a single perturbation $\mathbf{q}$, and keep it fixed during the algorithm execution. As a result, the algorithm is effectively optimizing the following smooth *surrogate* of the original problem (1):

$$\min_{\mathbf{x}\in\mathcal{X}} \left\{ F(\mathbf{x}) = f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) = \mathbb{E}_\xi[\widetilde{f}(\mathbf{x}, \mathbf{y}^*(\mathbf{x}); \xi)] \right\} \tag{13a}$$

$$\text{s.t. } \mathbf{y}^*(\mathbf{x}) \in \arg\min_{\mathbf{y}\in\mathbb{R}^{d_\ell}} \left\{ g(\mathbf{x}, \mathbf{y}) = h(\mathbf{x}, \mathbf{y}) + \mathbf{q}^T\mathbf{y} \mid A\mathbf{y} \le \mathbf{b} \right\}, \tag{13b}$$

where $\mathbf{q} \in \mathbb{R}^{d_\ell}$ is generated from a continuous measure only once and thus is considered fixed. Next, we show that the original problem (1) and the smoothed *surrogate* problem (13) are "close". Specifically, we show below that the original implicit function $G(\mathbf{x})$ and the "smoothed" implicit function $F(\mathbf{x})$ differ by a quantity that is controlled by the size of the perturbation vector $\mathbf{q}$.

Proposition 3.2 (Appendix D.2.2)**.** *Under Ass. 2.1 and 2.7, we have:*

$$|G(\mathbf{x}) - F(\mathbf{x})| \le \overline{L}_f \frac{\|\mathbf{q}\|}{\mu_g}, \ \forall\, \mathbf{x} \in \mathcal{X}.$$

Note that the only requirement on $\mathbf{q}$ is that it is generated from a continuous measure. Therefore we can always choose a distribution such that $\|\mathbf{q}\|$ is arbitrarily small. In the following two subsections, we will analyze the convergence for Alg. 1 and 2, respectively. Next, let us analyze Alg. 1. We have the following asymptotic result.

Theorem 3.3 (Appendix D.2.3)**.** *Suppose Ass. 2.1, 2.7 hold. At each iteration r of Alg. 1 we find $0 < a^r < 1$ such that the Armijo-type condition (12) is satisfied with $\epsilon(\delta; r) = L_f\delta^r + \overline{L}_F L_F a^r \delta^r + L_f \delta^{r+1} + L_F^2 \sigma a^r (\delta^r)^2 + 2L_F\overline{L}_F\sigma a^r\delta^r$. Further, we select δ^r such that Ass. 2.5 is satisfied, $\lim_{r\to\infty}\delta^r = 0$, and it holds that $\delta^r/a^r \sim \mathcal{O}(c^r)$, where c^r is some sequence with $\lim_{r\to\infty} c^r = 0$. In addition, the sequence $\widehat{\mathbf{d}}^r$ is selected such that it is gradient related to $\widehat{\nabla} F(\mathbf{x}^r)$, i.e., "for any subsequence $\{\mathbf{x}^r\}_{r\in\mathcal{R}}$ converging to a non-stationary point, the corresponding subsequence $\{\widehat{\mathbf{d}}^r\}_{r\in\mathcal{R}}$ is bounded and satisfies $\limsup_{r\to\infty, r\in\mathcal{R}} \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r < 0$" (Bertsekas, 1998, eq. 1.13). Then w.p. 1 the limit point $\bar{\mathbf{x}}$ of the sequence of iterates generated by [D]SIGD Alg. 1 is a stationary point.*

Note that in Theorem 3.3 only asymptotic convergence is guaranteed. However, this is the best we can do since we do not impose any Lipschitz smoothness or convexity assumptions. On the other hand, in the special cases where the implicit function is weakly-convex, strongly convex or convex (but still not Lipschitz smooth), it is possible to derive finite-time convergence guarantees as presented next. Towards this end, we need to impose the additional assumption that the set $\mathcal{X}$ is bounded; combining this property with Assumption 2.1 implies that $\mathcal{X}$ is compact. So, in the following results we assume that $\mathcal{X}$ is a compact set with diameter $\mathcal{D}_\mathcal{X} := \sup_{\mathbf{x},\bar{\mathbf{x}}\in\mathcal{X}} \|\mathbf{x} - \bar{\mathbf{x}}\|$.

Weakly Convex Objective. We make the following assumption on the implicit function $F(\cdot)$.

Assumption 3.4. We assume that for some $\rho > 0$ the implicit function $F(\mathbf{x})$ satisfies: $F(\mathbf{z}) \ge F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), \mathbf{z} - \mathbf{x}\rangle - \frac{\rho}{2}\|\mathbf{z} - \mathbf{x}\|^2 \ \forall\, \mathbf{x}, \mathbf{z} \in \mathbb{R}^{d_u}$.

Assumption 3.4 implies that the function $F(\mathbf{x}) + \frac{\hat{\rho}}{2}\|\mathbf{x}\|^2$ for $\hat{\rho} = \rho$ is convex while for $\hat{\rho} > \rho$ is strongly convex with modulus $\hat{\rho} - \rho$. Many problems of practical interest satisfy the weak-convexity, for example, phase retrieval (Davis et al., 2020), covariance matrix estimation (Chen et al., 2015), dictionary learning (Davis & Drusvyatskiy, 2019), Robust PCA (Candès et al., 2011) etc. (please see (Davis & Drusvyatskiy, 2019) and (Drusvyatskiy, 2017) for more details). For providing guarantees for the [S]SIGD algorithm we utilize a Moreau envelope based analysis. For this purpose, we first rephrase the UL problem as an unconstrained one: $\min_{\mathbf{x}\in\mathbb{R}^{d_u}} H(\mathbf{x}) := F(\mathbf{x}) + \mathbf{I}_\mathcal{X}(\mathbf{x})$, where $\mathbf{I}_\mathcal{X}(\mathbf{x})$ is the indicator function of set $\mathcal{X}$ defined as: $\mathbf{I}_\mathcal{X}(\mathbf{x}) := 0$ if $\mathbf{x} \in \mathcal{X}$ and $\mathbf{I}_\mathcal{X}(\mathbf{x}) := \infty$ if $\mathbf{x} \notin \mathcal{X}$. Below we define the Moreau envelope of $H(\mathbf{x})$.

Definition 3.5. Given $\lambda > 0$, the Moreau envelope of $H(\mathbf{x})$ is defined as

$$\begin{aligned} H_\lambda(\mathbf{x}) &:= \min_{\mathbf{z}\in\mathbb{R}^{d_u}} \left\{ H(\mathbf{z}) + \frac{1}{2\lambda}\|\mathbf{x} - \mathbf{z}\|^2 \right\} \\ &= \min_{\mathbf{z}\in\mathcal{X}} \left\{ F(\mathbf{z}) + \frac{1}{2\lambda}\|\mathbf{x} - \mathbf{z}\|^2 \right\}, \end{aligned}$$

where the second equality follows from the definition of $H(\mathbf{x})$. Moreover, we denote the proximal map of $H(\mathbf{x})$ as $\hat{\mathbf{x}} := \text{prox}_{\lambda H}(x)$ which is defined as

$$\begin{aligned} \hat{\mathbf{x}} &:= \arg\min_{\mathbf{z}\in\mathbb{R}^{d_u}} \left\{ H(\mathbf{z}) + \frac{1}{2\lambda}\|\mathbf{x} - \mathbf{z}\|^2 \right\} \\ &= \arg\min_{\mathbf{z}\in\mathcal{X}} \left\{ F(\mathbf{z}) + \frac{1}{2\lambda}\|\mathbf{x} - \mathbf{z}\|^2 \right\}. \end{aligned}$$

The norm of the gradient of the Moreau envelope satisfies the following:

$$\|\mathbf{x} - \hat{\mathbf{x}}\| = \lambda\|\nabla H_\lambda(\mathbf{x})\|, H(\hat{\mathbf{x}}) \le H(\mathbf{x}), \tag{14}$$

$$\text{and} \quad \text{dist}(0; \partial H(\hat{\mathbf{x}})) \le \|\nabla H_\lambda(\mathbf{x})\|, \tag{15}$$

where $\text{dist}(0; \partial H(\hat{\mathbf{x}})) = -\inf_{\mathbf{v}:\|\mathbf{v}\|\le 1} H'(\mathbf{x}; \mathbf{v})$ and $H'(\mathbf{x}; \mathbf{v})$ denotes the directional derivative of H at $\mathbf{x}$ in

direction $\mathbf{v}$. Note that a small gradient $\|\nabla H_\lambda(\mathbf{x})\|$ implies that $\mathbf{x}$ is near some point $\hat{\mathbf{x}}$ that is nearly stationary (Davis & Drusvyatskiy, 2019). Then we have the following result.

Theorem 3.6 (Appendix D.2.4). *Under Ass. 2.1, 2.5, 2.7, 2.9 and 3.4, with step-sizes $\beta^r = \beta$ for all $r \in \{0, \dots, T-1\}$ and for any constant $\hat{\rho} > \frac{3\rho}{2}$, the iterates generated by Algorithm 2 satisfy (w.p. 1)*

$$\frac{1}{T}\sum_{r=0}^{T-1}\mathbb{E}\|\nabla H_{1/\hat{\rho}}(\mathbf{x}^r)\|^2 \leq \frac{2\hat{\rho}}{2\hat{\rho}-3\rho}\left[\frac{H_{1/\hat{\rho}}(\mathbf{x}^0)-H^*}{\beta T} + \beta\hat{\rho}(\sigma_F^2+\overline{L}_F^2) + \frac{\hat{\rho}}{2\rho}L_F^2\delta^2\right].$$

Theorem 3.6 implies that with the choice of $\beta = \mathcal{O}(1/\sqrt{T})$, the [S]SIGD algorithm converges to a stationary point at a rate of $\mathcal{O}(1/\sqrt{T})$ with an additive error determined by the accuracy of the LL problem's solution δ (see Ass. 2.5).

Strongly Convex and Convex Objective. Next, we provide the guarantees for the case when the implicit function is strongly convex. We make the following assumption.

Assumption 3.7. We assume that the objective $F(\mathbf{x})$ is μ_F-strongly convex, i.e., $F(\mathbf{z}) \geq F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), \mathbf{z}-\mathbf{x}\rangle + \frac{\mu_F}{2}\|\mathbf{x}-\mathbf{z}\|^2 \; \forall \mathbf{z}, \mathbf{x} \in \mathcal{X}$. Note that for $\mu_F = 0$, the objective becomes convex.

Theorem 3.8 (Appendix D.2.5). *Under the Assumptions 2.1, 2.5, 2.7, 2.9 and 3.7, with $\mu_F > 0$ and the choice of step-sizes $\beta^r = \frac{1}{\mu_F(r+1)}$ the iterates generated by Algorithm 2 satisfy the following (w.p. 1),*

$$\mathbb{E}[F(\underline{\mathbf{x}}) - F^*] \leq \frac{(\sigma_F^2+\overline{L}_F^2)}{\mu_F}\frac{\log(T)}{T} + D_{\mathcal{X}}L_F\delta.$$

Theorem 3.9 (Appendix D.2.6). *Under Assumption 2.1, 2.5, 2.7, 2.9 and 3.7, with $\mu_F = 0$, and step-sizes $\beta^r = \beta$ for $r \in \{0, \dots, T-1\}$, the iterates generated by Algorithm 2 satisfy the following (w.p. 1),*

$$\mathbb{E}[F(\underline{\mathbf{x}}) - F^*] \leq \frac{\|\mathbf{x}^1-\mathbf{x}^*\|^2}{\beta T} + 2\beta(\sigma_F^2+\overline{L}_F^2) + D_{\mathcal{X}}L_F\delta,$$

where $\underline{\mathbf{x}} = \frac{1}{T}\sum_{r=0}^{T-1}\mathbf{x}^r$.

The results of Theorems 3.8 and 3.9 imply that the implicit function $F(\mathbf{x})$ converges to the optimal value at a rate of $\mathcal{O}(\log(T)/T)$ for strongly-convex objectives with diminishing step-sizes, and at a rate of $\mathcal{O}(1/\sqrt{T})$ for convex objectives with $\beta = \mathcal{O}(1/\sqrt{T})$. Note that convergence is shown to a neighborhood of the optimal solution where its size is determined by the size of the LL error δ.

4. Experiments

In this section, we evaluate the performance of Algorithms 1 and 2 via numerical experiments. First, we compare the performance of [D]SIGD to the recently proposed PDBO (Sow et al., 2022) for constrained bilevel optimization on a quadratic bilevel problem. Then in the second set of experiments, we evaluate the performance of [S]SIGD against popular adversarial training algorithms.

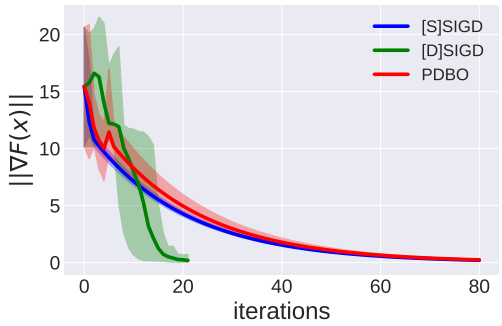


Figure 2: $\|\nabla F(\mathbf{x})\|$ vs # of iterations.

Quadratic Bilevel Optimization. Consider the quadratic bilevel problem of the form (1) with

$$f(\mathbf{x},\mathbf{y}) = \frac{1}{4}\|\mathbf{x}\|^2 + 10\mathbf{x}^T\mathbf{y} - \frac{1}{4}\|\mathbf{y}\|^2 + \mathbf{1}^T\mathbf{x} + \mathbf{1}^T\mathbf{y} + 1 \quad (16)$$

$$\text{and } h(\mathbf{x},\mathbf{y}) = \mathbf{x}^T\mathbf{y} + \frac{1}{2}\|\mathbf{y}\|^2 + x_1 + y_2, \quad (17)$$

and linear constraints of the form $|y_i| \leq 1, i \in \{1,2\}$. Here, $\mathbf{x} = [x_1, x_2]^T, \mathbf{y} = [y_1, y_2]^T$ with $x_i, y_i \in \mathbb{R}$ for $i \in \{1,2\}$, and $\mathbf{1} = [1,1]^T$. The evaluation criterion is the stationarity gap $\|\nabla F(\mathbf{x})\|$.

On this problem we execute [D]SIGD (Algorithm 1), [S]SIGD (Algorithm 2), and PDBO (Sow et al., 2022). In the first two cases, we solve the inner-level problem using 10 steps of projected gradient descent with stepsize 10^{-1}. For the stepsize of [S]SIGD, we choose $\beta = 0.1$, while in [D]SIGD we find the proper Armijo step-size by successively adapting (by increasing m) the quantity $a_r = (0.9)^m$ until condition (12) is met. In PDBO we select 10^{-1} for the stepsizes of both the primal and dual steps, and the number of inner iterations is set to 10. In Figure 2, we plot the convergence curves for the three algorithms with respect to number of iterations; the results are averaged over 10 runs. Note that the line search [D]SIGD method outperforms the fixed step-size [S]SIGD and PDBO while [S]SIGD performs similar to PDBO.

Adversarially Robust Learning.[4] We consider an adversarial learning problem of the form given in (2). For the perturbation, we focus on the ϵ-tolerant ℓ_∞-norm attack constraint, namely $\mathcal{Y} = \{\mathbf{y} \in \mathbb{R}^{d_\ell} \mid \|\mathbf{y}\|_\infty \leq \epsilon\}$, which can easily be expressed as a linear inequality constraint

[4]The code can be found in the following link: https://anonymous.4open.science/r/icml23-bilevel-gaussian/

CIFAR-100, $\epsilon = 8/255$							
Metrics	AT	TRADES	[S]SIGD (Gaussian variance σ^2)				
			2e−5	4e−5	6e−5	8e−5	1e−4
SA	$53.83_{\pm 0.19}$	$53.33_{\pm 0.18}$	$53.88_{\pm 0.22}$	$54.01_{\pm 0.24}$	$53.79_{\pm 0.14}$	$54.44_{\pm 0.18}$	$57.74_{\pm 0.22}$
RA	$27.36_{\pm 0.24}$	$28.44_{\pm 0.17}$	$27.43_{\pm 0.12}$	$28.22_{\pm 0.10}$	$28.12_{\pm 0.14}$	$27.14_{\pm 0.21}$	$25.22_{\pm 0.15}$
$\epsilon = 16/255$							
SA	$42.06_{\pm 0.17}$	$42.19_{\pm 0.23}$	$44.06_{\pm 0.19}$	$45.66_{\pm 0.25}$	$46.57_{\pm 0.22}$	$47.11_{\pm 0.32}$	$47.46_{\pm 0.44}$
RA	$15.10_{\pm 0.28}$	$16.59_{\pm 0.26}$	$15.51_{\pm 0.17}$	$14.18_{\pm 0.22}$	$13.92_{\pm 0.25}$	$13.54_{\pm 0.18}$	$13.42_{\pm 0.26}$

Table 1: Performance overview of different methods on CIFAR-100 (Krizhevsky et al., 2009) with ResNet-18 (He et al., 2016). The result $a_{\pm b}$ represents the mean a standard deviation b over 5 trials.

as in the LL problem of (2). Differently, though than the bilevel problems we are considering in this work, the robust learning problem is more challenging as the LL objective is not necessarily strongly-convex in $\mathbf{y}$. We consider two widely accepted adversarial learning methods as our baselines, namely AT (Madry et al., 2018) and TRADES (Zhang et al., 2019b). Also, we consider two representative datasets CIFAR-10/100 (Krizhevsky et al., 2009) and adopt the ResNet-18 (He et al., 2016) model; the results for CIFAR-10 are provided in Appendix C. In particular, we studied two widely used, (Madry et al., 2018; Wong et al., 2020) attack budget choices $\epsilon \in \{8/255, 16/255\}$. In the implementation of our [S]SIGD method, we adopt a perturbation generated by a Gaussian random vector $\mathbf{q}$ with variances from the following list $\sigma^2 \in \{2\text{e}{-5}, 4\text{e}{-5}, 6\text{e}{-5}, 8\text{e}{-5}, 1\text{e}{-4}, \}$, in order to study different levels of smoothness [5]. Moreover, for solving the LL problem in each iteration we select a fixed batch of samples. We choose f_i to be cross-entropy loss and $h_i = -f_i + \lambda \|\mathbf{y}_i\|^2$ for hyper-parameter $\lambda > 0$. For [S]SIGD, we follow the implementation of (Zhang et al., 2022) but with perturbations in the LL problem. We evaluate the robustly trained model with two metrics, namely the standard accuracy (SA) and robust accuracy (RA), where we evaluate the accuracy of the robustified model on the clean and attacked test set, respectively; the attacked set is generated using PGD-50-10 (Madry et al., 2018) (i.e., 50-step PGD attack with 10 restarts). Desirably, a well-trained model possesses high RA while maintaining simultaneously the SA at a high level.

Table 1 shows the performance overview of our experiments. We make the following observations. First, a low level of perturbation variance (*e.g.*, $\sigma^2 \in \{2\text{e}{-5}, 4\text{e}{-5}\}$) in general improves both SA as well as RA, which presents an enhanced RA-SA trade-off. For example, in the setting (CIFAR-100, $\epsilon = 16/255$), our algorithm boosts the RA by over 0.3% and the SA by 2%. Second, a high level of perturbation variance harms the robustness but results in high SA. This is reasonable, since the stochastic gradient becomes too noisy with large variances. Third, our method outperforms AT and closely matches the performance of the stronger baseline TRADES. However, we would like to stress that the intent of our work is not to design a specialized adversarial learning method, and thus robustness gap between our method and the strong baseline does not diminish the value of our method. Additional details are provided in Appendix C.2.

[5] Note that in practice the [S]SIGD algorithm also works without the addition of a perturbation, potentially though with inferior performance. However, in our experiments, we are perturbing the LL problem in order to be consistent with the theory and study the effect of different perturbation levels. Please see Appendix C.2 for additional experiments, in which the case where there is no perturbation is also considered.

5. Conclusion

In this work we develop a framework for the solution of a special class of constrained bilevel problems where the LL task has linear constraints and strongly convex objective. The key challenge we are dealing with in this problem class is the non-differentiability of the implicit function, which is addressed with the use of a perturbation-based smoothing technique. This allows us to compute the gradient of the implicit function, and develop first-order algorithms to find its stationary points. In the future we would be interested in studying other special classes of constrained bilevel problems (i.e., problems with different constraint sets) and their properties.

Acknowledgements.

I. Tsaknakis and M. Hong are supported in part by NSF grants CIF-1910385, CMMI-1727757.

References

Abedi, A., Hesamzadeh, M. R., and Romerio, F. An acopf-based bilevel optimization approach for vulnerability assessment of a power system. *International Journal of Electrical Power & Energy Systems*, 125:106455, 2021. ISSN 0142-0615.

Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., and Kolter, J. Z. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.

Allende, G. B. and Still, G. Solving bilevel programs with the kkt-approach. *Mathematical Programming*, 138:309–332, 2013.

Amos, B. and Kolter, J. Z. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pp. 136–145. PMLR, 2017.

Andriushchenko, M. and Flammarion, N. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

Arias, D. A., Mota, A. A., Mota, L. T. M., and Castro, C. A. A bilevel programming approach for power system operation planning considering voltage stability and economic dispatch. In *2008 IEEE/PES Transmission and Distribution Conference and Exposition: Latin America*, pp. 1–6, 2008. doi: 10.1109/TDC-LA.2008.4641718.

Bertrand, Q., Klopfenstein, Q., Blondel, M., Vaiter, S., Gramfort, A., and Salmon, J. Implicit differentiation of lasso-type models for hyperparameter optimization. In *International Conference on Machine Learning*, pp. 810–821. PMLR, 2020.

Bertrand, Q., Klopfenstein, Q., Massias, M., Blondel, M., Vaiter, S., Gramfort, A., and Salmon, J. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *Journal of Machine Learning Research*, 23(149):1–43, 2022.

Bertsekas, D. P. *Nonlinear programming, 2nd ed.* Athena Scientific Belmont, MA, 1998.

Calamai, P. H. and Moré, J. J. Projected gradient methods for linearly constrained problems. *Mathematical programming*, 39(1):93–116, 1987.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3): 1–37, 2011.

Cecchini, M., Ecker, J., Kupferschmid, M., and Leitch, R. Solving nonlinear principal-agent problems using bilevel programming. *European Journal of Operational Research*, 230(2):364–373, 2013.

Chang, T.-H., Hong, M., Wai, H.-T., Zhang, X., and Lu, S. Distributed learning in the nonconvex world: From batch data to streaming and beyond. *IEEE Signal Processing Magazine*, 37(3):26–38, 2020. doi: 10.1109/MSP.2020.2970170.

Chen, T., Sun, Y., and Yin, W. A single-timescale stochastic bilevel optimization method. *arXiv preprint arXiv:2102.04671*, 2021a.

Chen, T., Sun, Y., and Yin, W. Tighter analysis of alternating stochastic gradient method for stochastic nested problems. *arXiv preprint arXiv:2106.13781*, 2021b.

Chen, Y., Chi, Y., and Goldsmith, A. J. Exact and stable covariance estimation from quadratic sampling via convex programming. *IEEE Transactions on Information Theory*, 61(7):4034–4059, 2015.

Colson, B., Marcotte, P., and Savard, G. Bilevel programming: A survey. *4or*, 3(2):87–107, 2005.

Davis, D. and Drusvyatskiy, D. Stochastic model-based minimization of weakly convex functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

Davis, D., Drusvyatskiy, D., and Paquette, C. The nonsmooth landscape of phase retrieval. *IMA Journal of Numerical Analysis*, 40(4):2652–2695, 2020.

Dempe, S. and Zemkoho, A. *Bilevel optimization*. Springer, 2020.

Didi-Biha, M., Marcotte, P., and Savard, G. *Path-based formulations of a bilevel toll setting problem*, pp. 29–50. Springer US, Boston, MA, 2006. ISBN 978-0-387-34221-4. doi: 10.1007/0-387-34221-4_2.

Donti, P., Amos, B., and Kolter, J. Z. Task-based end-to-end model learning in stochastic optimization. *Advances in neural information processing systems*, 30, 2017.

Drusvyatskiy, D. The proximal point method revisited. *arXiv preprint arXiv:1712.06038*, 2017.

Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pp. 1165–1173. PMLR, 2017.

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pp. 1568–1577. PMLR, 2018.

Friesz, T. and Bernstein, D. *Foundations of Network Optimization and Games*. Complex Networks and Dynamic Systems. Springer US, 2015. ISBN 9781489975942.

Ghadimi, S. and Wang, M. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gould, S., Hartley, R., and Campbell, D. Deep declarative networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):3988–4004, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization*, 33(1):147–180, 2023. doi: 10.1137/20M1387341. URL https://doi.org/10.1137/20M1387341.

Ji, K., Yang, J., and Liang, Y. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning*, pp. 4882–4892. PMLR, 2021.

Kalashnikov, D. V., Camacho-Vallejo, J.-F., Askin, R., and Kalashnykova, N. Comparison of algorithms for solving a bi-level toll setting problem. *International journal of innovative computing, information & control: IJICIC*, 6: 3529–3549, 08 2010.

Khanduri, P., Zeng, S., Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A momentum-assisted single-timescale stochastic approximation algorithm for bilevel optimization, 2021a.

Khanduri, P., Zeng, S., Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in Neural Information Processing Systems*, 34, 2021b.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lebourg, G. Generic differentiability of lipschitzian functions. *Transactions of the American Mathematical Society*, 256:125–144, 1979.

Lecture. 5: Correspondences and berge's maximum theorem. *Math Camp Notes, Department of Economics, Yale University*, 2017.

Li, Y., Song, L., Wu, X., He, R., and Tan, T. Learning a bi-level adversarial network with global and local perception for makeup-invariant face verification. *Pattern Recognition*, 90:99–108, 2019.

Lin, G.-H., Xu, M., and Ye, J. J. On solving simple bilevel programs with a nonconvex lower level program. *Math. Program.*, 144(1–2):277–305, April 2014.

Liu, R., Gao, J., Zhang, J., Meng, D., and Lin, Z. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (12):10045–10067, 2021a.

Liu, R., Liu, X., Yuan, X., Zeng, S., and Zhang, J. A value-function-based interior-point method for non-convex bi-level optimization. In *International Conference on Machine Learning*, pp. 6882–6892. PMLR, 2021b.

Liu, R., Liu, X., Zeng, S., Zhang, J., and Zhang, Y. Value-function-based sequential minimization for bi-level optimization. *arXiv preprint arXiv:2110.04974*, 2021c.

Lu, S., Razaviyayn, M., Yang, B., Huang, K., and Hong, M. Finding second-order stationary points efficiently in smooth nonconvex linearly constrained optimization problems. *Advances in Neural Information Processing Systems*, 33:2811–2822, 2020.

Luo, Z.-Q., Pang, J.-S., and Ralph, D. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Mairal, J., Bach, F., and Ponce, J. Task-driven dictionary learning. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):791–804, 2011.

Marcotte, P., Savard, G., and Zhu, D. A trust region algorithm for nonlinear bilevel programming. *Oper. Res. Lett.*, 29:171–179, 11 2001. doi: 10.1016/S0167-6377(01)00092-X.

Migdalas, A. Bilevel programming in traffic planning: Models, methods and challenge. *Journal of global optimization*, 7(4):381–405, 1995.

Mirrlees, J. A. The theory of moral hazard and unobservable behaviour: Part i. *The Review of Economic Studies*, 66 (1):3–21, 1999.

Moosavi-Dezfooli, S.-M., Fawzi, A., Uesato, J., and Frossard, P. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9078–9086, 2019.

Nutini, J., Schmidt, M., and Hare, W. "active-set complexity" of proximal gradient: How long does it take to find the sparsity pattern? *Optimization Letters*, 13(4): 645–655, 2019.

Parise, F. and Ozdaglar, A. Sensitivity analysis for network aggregative games. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 3200–3205. IEEE, 2017.

Pedregosa, F. Hyperparameter optimization with approximate gradient. In *International conference on machine learning*, pp. 737–746. PMLR, 2016.

Raghunathan, A. U. and Biegler, L. T. Mathematical programs with equilibrium constraints (mpecs) in process engineering. *Computers & Chemical Engineering*, 27 (10):1381–1392, 2003. ISSN 0098-1354.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Shaban, A., Cheng, C.-A., Hatch, N., and Boots, B. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1723–1732. PMLR, 2019.

Sinha, A., Malo, P., and Deb, K. A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2017.

Sinha, A., Khandait, T., and Mohanty, R. A gradient-based bilevel optimization approach for tuning hyperparameters in machine learning. *arXiv preprint arXiv:2007.11022*, 2020.

Sow, D., Ji, K., Guan, Z., and Liang, Y. A constrained optimization approach to bilevel optimization with multiple inner minima. *arXiv preprint arXiv:2203.01123*, 2022.

Von Stackelberg, H. and Von, S. H. *The theory of the market economy*. Oxford University Press, 1952.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJx040EFvH.

Xiao, Q., Shen, H., Yin, W., and Chen, T. Alternating projected sgd for equality-constrained bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 987–1023. PMLR, 2023.

Yang, J., Ji, K., and Liang, Y. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34, 2021.

Yang, T., Yi, X., Wu, J., Yuan, Y., Wu, D., Meng, Z., Hong, Y., Wang, H., Lin, Z., and Johansson, K. H. A survey of distributed optimization. *Annual Reviews in Control*, 47: 278–305, 2019.

Ye, J. and Zhu, D. Optimality conditions for bilevel programming problems. *Optimization*, 33(1):9–27, 1995.

Yousefian, F. Bilevel distributed optimization in directed networks. In *2021 American Control Conference (ACC)*, pp. 2230–2235. IEEE, 2021.

Zhang, D., Zhang, T., Lu, Y., Zhu, Z., and Dong, B. You only propagate once: Accelerating adversarial training via maximal principle. *Advances in Neural Information Processing Systems*, 32, 2019a.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019b.

Zhang, Y., Zhang, G., Khanduri, P., Hong, M., Chang, S., and Liu, S. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *International Conference on Machine Learning*, pp. 26693–26712. PMLR, 2022.

A. Related Literature

Bilevel optimization problems, initially encountered in the context of Stackelberg (leader-follower) games (Von Stackelberg & Von, 1952), find applications in a multitude of areas, including machine learning (Liu et al., 2021a), economics (Mirrlees, 1999), power systems (Abedi et al., 2021; Arias et al., 2008), chemical industry (Raghunathan & Biegler, 2003), transport research (Didi-Biha et al., 2006; Kalashnikov et al., 2010); see (Colson et al., 2005; Dempe & Zemkoho, 2020; Sinha et al., 2017; Liu et al., 2021a) for a number of survey papers. The "classical" approaches for solving bilevel problems include the use of approximate descent methods (Shaban et al., 2019; Ghadimi & Wang, 2018; Franceschi et al., 2017), penalty methods (Lin et al., 2014), KKT reformulations-based approaches (Allende & Still, 2013), value function-based methods (Ye & Zhu, 1995; Sow et al., 2022), and trust-region algorithms (Marcotte et al., 2001). In addition, bilevel problems are known to be related to mathematical programs with equilibrium constraints (MPEC) (Luo et al., 1996).

Recently, motivated by machine learning applications, gradient-based approaches have gained popularity for solving bilevel optimization problems (Liu et al., 2021a), e.g., in hyperparameter optimization (Shaban et al., 2019; Franceschi et al., 2017; 2018), and meta learning (Rajeswaran et al., 2019; Franceschi et al., 2018). The majority of those works are focused on solving bilevel problems with unconstrained strongly convex LL problem, for both stochastic and deterministic objectives (Ghadimi & Wang, 2018; Hong et al., 2023; Khanduri et al., 2021a;b; Chen et al., 2021a; Ji et al., 2021; Chen et al., 2021b; Yang et al., 2021). An attractive property of such problems is the existence and easy computability of the implicit gradient. Moreover, under mild assumptions, the implicit gradient for these problems can be shown to be Lipschitz smooth (e.g., see (Ghadimi & Wang, 2018, Lemma 2.2) and (Khanduri et al., 2021b, Lemma 3.1)). In a recent work (Xiao et al., 2023) the authors develop an implicit gradient method for problems with equality constraints in the LL. In this case, similarly to the unconstrained one, the implicit gradient is differentiable and Lipschitz continuous. In contrast, for bilevel problems with linear inequality LL constraints the implicit gradient in general might not exist, and even if it exists computing it in closed-form is a challenging task. As discussed earlier, we develop a perturbation-based smoothing framework for that constrained LL problem that ensures the existence of the implicit gradient in an almost sure sense, and allows us to compute an expression for the implicit gradient.

In Liu et al. (2021c) and Sow et al. (2022) the authors have considered bilevel optimization with (general) constraints in the LL problem. Both papers develop a value function-based framework that leads to a single level problem with non-convex constraints. In Liu et al. (2021c) a sequential minimization approach is followed where the value-function and the LL constraints are incorporated into the objective using penalty or barrier functions. In Sow et al. (2022) a primal-dual-based framework is proposed in which the problem is regularized with the addition of a strongly-convex penalty term, while a constant error term is added to make the constraint set strictly feasible. In contrast, our approach relies only on a small linear perturbation which can be made arbitrarily small without practically changing the landscape of the LL problem.

There is also a line of works (Amos & Kolter, 2017; Agrawal et al., 2019; Donti et al., 2017; Gould et al., 2021) about implicit differentiation in deep learning literature. These works Deep-Learning-type (DL-type) are indeed related to ours, in the sense that at the core of both of them lies the computation of the gradient/Jacobian of the solution of an optimization problem. However, there are some key differences. First, in our work we consider a constrained bilevel optimization problem and we are interested in analyzing this problem from an optimization perspective. On the other hand, in the DL-type works the optimization problems that are studied describe the input-output relationships of neural networks layers and the main focus lies in deriving Jacobians for the backward pass. Secondly, in our work we study a special bilevel problem (the constraints are linear) and derive a closed form expression (assuming that we have access to the solution of the LL problem) for the implicit gradient. On the contrary, in the DL-type works the underlying problems have more general constraints and the Jacobian is usually computed using numerical methods (e.g., solving iteratively a system of KKT equations), rather than analytically. Finally, in our work the focus is on studying the properties of the bilevel problem (e.g. differentiability, approximation errors), developing (deterministic and stochastic) algorithms, and performing a convergence analysis. On the other hand, DL-type works focus mainly on the Jacobian computation and its implementation.

Finally, there is a number of works on implicit differentiation on non-smooth problems (Mairal et al., 2011; Bertrand et al., 2022; 2020). However, these works typically deal with special (non-smooth) LL problems, e.g., in (Mairal et al., 2011; Bertrand et al., 2020) the non-smooth term in the LL is the $\ell 1$-norm, and in (Bertrand et al., 2022) the non-smooth term is separable. On the contrary, in our work we are considering smooth LL problems and general linear inequality constraints.

B. Solution methods for the LL problem

The LL problem is a strongly convex linearly constrained optimization task. As a result, there exist many efficient ways to find its solutions. In order to discuss about them, we consider two different classes of problems depending on the exact form of the linear constraints and the difficulty of computing the respective projection operator: 1) the projection has a closed-form solution, 2) the projection requires the solution of an optimization problem. Before we proceed, we would like to stress that the problem we are solving, i.e., the bilevel problem with linear constraints in the LL, is a very challenging one, regardless of the specific form and the exact way we approach the solution of the LL problem.

In the first class of problems, where the projection can be computed in closed form, we have problems with special linear constraints. One characteristic example is box constraints, i.e. constraints of the form $\mathbf{a} \leq \mathbf{y} \leq \mathbf{b}$, where the inequalities apply in a component-wise manner. These constraints appear in applications, such as adversarial learning (see the motivating applications in the main text). In this case, we can use some first-order iterative algorithm to solve the LL problem and project each iterate onto the constraint set using the closed-form expression (which only incurs a constant cost per iteration). For instance, we can use the projected gradient descent method which probably converges to the optimal solution with a linear rate.

In the second class of problems, the projection operator does not possess a closed-form expression. In this case we can approach the LL problem as a convex optimization task, and solve it using some convex optimization solver (e.g. employing interior-point methods) to obtain a highly accurate solution with a complexity of $\mathcal{O}\left(p(d_\ell, k)\log(d_\ell/\epsilon)\right)$, where $p(\cdot)$ is some polynomial and ϵ is solution accuracy. Alternatively, as mentioned in the previous case, we can use a projected gradient descent-type method that enjoys a linear convergence rate guarantee. Differently from the previous case though the projection operator computed at each iteration requires the solution of an optimization problem. Nonetheless, the projection task we are referring to is a (strongly convex) quadratic linearly constrained problem, that is a special quadratic programming task, which is easy to solve in practice. In algorithm 3 we describe the solution of the LL using a projected gradient descent algorithm.

Algorithm 3 Projected Gradient Descent (**PGD**)

1: **Input:** Initial parameter $\mathbf{y}^0$, Current iterate $\mathbf{x}$, # iter T, step-sizes $\{\gamma^r\}_{r=0}^{T-1}$, Constraints $A, \mathbf{b}$
2: **for** $r = 0, 1, \ldots, T-1$ **do**
3: $\quad \mathbf{y}^{r+1} = \mathbf{y}^r - \gamma^r \nabla_y g(\mathbf{x}, \mathbf{y}^r)$
4: $\quad$ Project $\mathbf{y}^{r+1}$ to $\mathcal{Y} = \{\mathbf{y} \in \mathbb{R}^{d_\ell} | A\mathbf{y} \leq \mathbf{b}\}$ by solving the following QP:

$$\min_{\mathbf{y} \in \mathbb{R}^{d_\ell}} \|\mathbf{y} - \mathbf{y}^{r+1}\|^2 \text{ s.t. } A\mathbf{y} \leq \mathbf{b} \tag{18}$$

5: **end for**

C. Additional Experiments

In this section, we include additional experiments on quadratic bilevel optimization problems and Adversarial training along with the implementation details. First, we evaluate the performance of the [D]SIGD and [S]SIGD on quadratic bilevel optimization problems.

C.1. Numerical Results

We consider the following linearly constrained quadratic bilevel problems of the form (1) with the UL and the LL objectives defined as:

$$f(\mathbf{x}, \mathbf{y}) = \frac{1}{4}\|\mathbf{x}\|^2 + 5\mathbf{x}^T\mathbf{y} - \frac{1}{4}\|\mathbf{y}\|^2 \ , \ h(\mathbf{x}, \mathbf{y}) = \frac{1}{4}\|\mathbf{x}\|^2 + \frac{1}{2}\mathbf{x}^T\mathbf{y} + \frac{1}{4}\|\mathbf{y}\|^2 \tag{19}$$

$$f(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x}\|^2 + 2\mathbf{x}^T\mathbf{y} - \frac{1}{2}\|\mathbf{y}\|^2 \ , \ h(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y} + \frac{1}{2}\|\mathbf{y}\|^2. \tag{20}$$

$$f(\mathbf{x}, \mathbf{y}) = \frac{1}{4}\|\mathbf{x}\|^2 + 2\mathbf{x}^T\mathbf{y} - \frac{1}{4}\|\mathbf{y}\|^2 + 1 \ , \ h(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y} + \frac{1}{2}\|\mathbf{y}\|^2 + \mathbf{1^T x} + \mathbf{1^T y}. \tag{21}$$

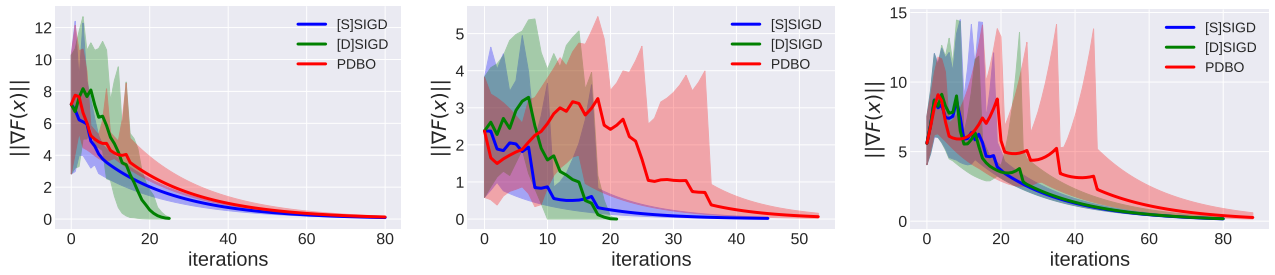


Figure 3: Convergence curves w.r.t. number of iterations. Left: problem (19); Center: problem (20); Right: problem (21).

Algorithm/Runtime(s)	Problem (19)	Problem (20)	Problem (21)	Problem (16)
[D]SIGD	7.13	10.34	9.12	6.73
[S]SIGD	1.34	1.39	3.90	1.28
PDBO	1.32	1.51	4.62	1.27

Table 2: The average runtime of the three algorithms we use in our experiments. The [D]SIGD is slower compared to the other two algorithms. However we would like to stress that the [D]SIGD was designed to handle difficult problems (e.g. without the convexity and Lipschitz gradients assumptions), rather than having speed in mind. On the other hand the [S]SIGD and PDBO algorithms (which enjoy convergence under stronger assumptions) achieve similar performance in terms of speed, with the exception of problem (21) where [S]SIGD appear to have a small edge.

In the first two cases, we have $d_u = d_l = 2$, and the linear constraints in the LL are of the form $-1 \leq y_i \leq 1,\ i \in \{1, 2\}$. In the third example, we have $d_u = d_l = 2$, and the linear constraints in the LL are of the form $-5 \leq y_i \leq 5,\ i \in \{1, 2\},\ -5 \leq y_1 + y_2 \leq 5$. We compare the performance of SIGD algorithms to recently proposed PDBO (Sow et al., 2022). In Figure 3 we present the evolution of the stationarity gap $\|\nabla F(\mathbf{x})\|$ during the execution of the three algorithms, for the problems (19), (20) and (21), respectively. The results are averaged over 10 random runs, and the variance of the results across these runs is reflected on the shaded region across the convergence curves. Also, the average runtime is presented in table 2. In our experiments, we choose the step-size using the backtracking line search for [D]SIGD as stated in Algorithm 1, while for [S]SIGD we choose a constant step-size. Note that since all problems are deterministic [S]SIGD utilizes a gradient estimator with zero variance.

In problem (19), we solve the LL problem using 10 steps of projected gradient descent with stepsize 0.1; in the case of [D]SIGD the stepsize is 1. For the stepsize of [S]SIGD, we choose $\beta = 0.1$, while in [D]SIGD we find the proper Armijo step-size by successively adapting (by increasing m) the quantity $a_r = (0.9)^m$ until condition (12) is met. In PDBO we select 0.1 for the stepsizes of both the primal and dual steps, and the number of inner iterations is set to 10. In problem (20), we solve the LL problem using 20 steps of projected gradient descent with stepsize 0.1; in the case of [D]SIGD the number of steps is 10 and the stepsize is 1. For the stepsize of [S]SIGD, we choose $\beta = 0.1$, while in [D]SIGD we find the proper Armijo step-size by successively adapting (by increasing m) the quantity $a_r = (0.95)^m$ until condition (12) is met. In PDBO we select 0.1 for the stepsizes of both the primal and dual steps, and the number of inner iterations is set to 20. In problem (21), we solve the LL problem (of both [D]SIGD and [S]SIGD) using 10 steps of projected gradient descent with stepsize 0.1. For the stepsize of [S]SIGD, we choose $\beta = 0.1$, while in [D]SIGD we find the proper Armijo step-size by successively adapting (by increasing m) the quantity $a_r = (0.9)^m$ until condition (12) is met. In PDBO we select 0.1 for the stepsizes of both the primal and dual steps, and the number of inner iterations is set to 10.

C.2. Adversarial Learning

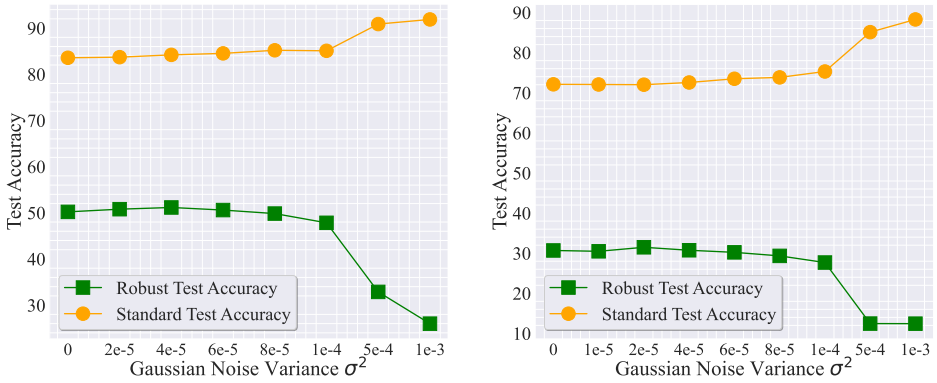
In this section, we present some additional results along with the implementation details for the adversarial learning problem. As noted earlier, we consider the adversarial learning problem of form (2). For learning the perturbation $\mathbf{y}^*(\mathbf{x})$, we focus on the ϵ-tolerant ℓ_∞-norm attack constraint, i.e., $\mathcal{Y} = \{\mathbf{y} \in \mathbb{R}^{d_\ell} \,|\, \|\mathbf{y}\|_\infty \leq \epsilon\}$. Note that this constraint can easily be expressed as a linear inequality constraint as in the LL problem in (2). In particular, we evaluate the performance of [S]SIGD on two widely used attack budget choices of $\epsilon \in \{8/255, 16/255\}$ (Madry et al., 2018; Zhang et al., 2019b; Wong et al., 2020; Andriushchenko & Flammarion, 2020; Zhang et al., 2019a). In the implementation of our

CIFAR-10, $\epsilon = 8/255$							
Metrics	AT	TRADES	[S]SIGD (Gaussian variance σ^2)				
			2e−5	4e−5	6e−5	8e−5	1e−4
SA	$80.78_{\pm 0.23}$	$80.23_{\pm 0.23}$	$80.70_{\pm 0.14}$	$81.20_{\pm 0.22}$	$81.52_{\pm 0.21}$	$83.19_{\pm 0.24}$	$85.08_{\pm 0.44}$
RA	$50.71_{\pm 0.21}$	$51.17_{\pm 0.19}$	$50.78_{\pm 0.21}$	$51.15_{\pm 0.19}$	$50.59_{\pm 0.18}$	$49.83_{\pm 0.23}$	$47.83_{\pm 0.13}$
$\epsilon = 16/255$							
SA	$70.31_{\pm 0.11}$	$70.22_{\pm 0.29}$	$71.43_{\pm 0.14}$	$72.79_{\pm 0.24}$	$73.50_{\pm 0.09}$	$73.98_{\pm 0.35}$	$75.31_{\pm 0.33}$
RA	$32.12_{\pm 0.18}$	$33.35_{\pm 0.14}$	$32.72_{\pm 0.25}$	$31.73_{\pm 0.10}$	$29.97_{\pm 0.14}$	$29.39_{\pm 0.15}$	$27.67_{\pm 0.07}$

Table 3: Performance overview of different methods on CIFAR-10 (Krizhevsky et al., 2009) with ResNet-18 (He et al., 2016). The result $a_{\pm b}$ represents the mean value a with a standard deviation of b over 5 random trials.

[S]SIGD method, we adopt a perturbation generated by a Gaussian random vector $\mathbf{q}$ with variances from the following list $\sigma^2 \in \{2\text{e}{-5}, 4\text{e}{-5}, 6\text{e}{-5}, 8\text{e}{-5}, 1\text{e}{-4}, \}$, in order to study different levels of smoothness. We choose f_i to be cross-entropy loss and $h_i = -f_i + \lambda \|\mathbf{y}_i\|^2$ with $\lambda > 0$ as a hyper-parameter. For solving (2), in each iteration we select a fixed batch of samples for both the UL and LL problems. Also, note that the ReLU-based neural networks commonly lead to a piece-wise linear decision boundary w.r.t. the inputs (Moosavi-Dezfooli et al., 2019). This implies that the implicit gradient in (11) can be further approximated using a Hessian-free implementation, where the Hessian of the LL problem can be approximated by λI (Zhang et al., 2022, Eq. (25)). Note that these approximations are common in practice and do not lead to performance degradation compared to the case when full Hessian is used to compute the implicit-gradient (Zhang et al., 2022, Table 5). Next, we analyze the effect of adding different perturbations $\mathbf{q}$ in the LL problem on the performance of [S]SIGD. Specifically, we choose $\mathbf{q} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ and evaluate the performance of [S]SIGD with σ^2.

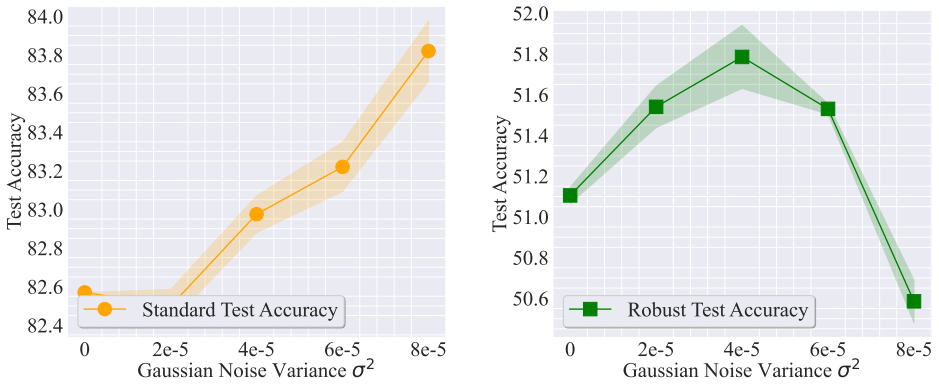
In Figure 4, we plot the robust accuracy (RA) and the standard accuracy (SA) with respect to the variance of the Gaussian perturbation vector used in the LL problem. As can be seen, the RA increases as the variance increases within a certain range. However, with stronger noise (i.e., $\sigma^2 > 10^{-4}$), the RA drops sharply, while the SA increases. This is reasonable, since high variance makes the true LL gradient noisy. For easier observation, in Figure 5 we zoom in the part of Figure 4 where $\sigma^2 \in [0, 8 \cdot 10^{-5}]$. It can be clearly seen that adding a small perturbation $\mathbf{q}$ helps in improving the RA.



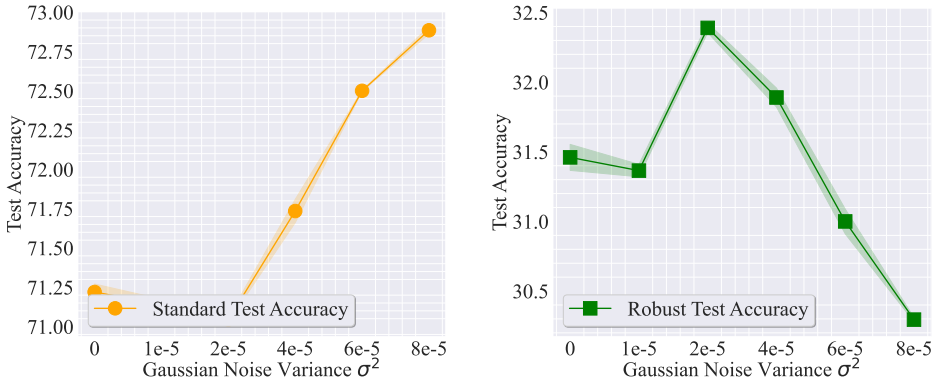
(a) $\epsilon = 8/255$ (b) $\epsilon = 16/255$

Figure 4: The influence of Gaussian variance on the RA and SA. The experiments are based on CIFAR-10 with ResNet-18 model.

Next, we compare the performance of [S]SIGD against two widely accepted adversarial learning methods as baselines, namely AT (Madry et al., 2018) and TRADES (Zhang et al., 2019b). Here, we present the results for CIFAR-10 dataset (Krizhevsky et al., 2009) and adopt the ResNet-18 (He et al., 2016). In Table 3, we compare the performance of [S]SIGD for different perturbation variances with classical AT (Madry et al., 2018) algorithm and TRADES (Zhang et al., 2019b); in Table 4 the runtime per epoch is presented. Note that for appropriate choice of perturbation variance [S]SIGD outperforms



(a) $\epsilon = 8/255$, CIFAR-10



(b) $\epsilon = 16/255$, CIFAR-10

Figure 5: The influence of Gaussian variance on the RA and SA.

Method	AT	TRADES	[S]SIGD (Gaussian variance σ^2)				
			2e−5	4e−5	6e−5	8e−5	1e−4
Runtime (s/epoch)	29.6	31.4	38.8	38.6	38.7	38.6	38.7

Table 4: Average runtime (in seconds) per epoch on CIFAR-10 (Krizhevsky et al., 2009) with ResNet-18 (He et al., 2016).

the classical AT algorithm while performs is only slightly worse compared to TRADES, especially, for higher attack budget of $\epsilon = 16/255$. Finally, in terms of runtime (per epoch) the proposed [S]SIGD is slower to the other baselines. However, it should be noted that AT and TRADES are methods tailored for the adversarial experiments we consider in this section, whereas [S]SIGD, under certain assumptions, can be applied to any bilevel problem of the form (1).

D. Proofs

D.1. Proofs of Section 2

D.1.1. Proof of Proposition 2.2

Note that the goal of Proposition 2.2 is to establish the continuity of the mapping $\overline{\mathbf{y}}^*(\mathbf{x})$ and the implicit function $G(\mathbf{x}) := f(\mathbf{x}, \overline{\mathbf{y}}^*(\mathbf{x}))$. In the following, we will show that under Assumption 2.1, $\overline{\mathbf{y}}^*(\mathbf{x})$ is in fact continuous, which will then utilize to establish the continuity of $G(\mathbf{x})$. Before starting the proof we need a few definitions. Consider the LL problem (1b) and let us denote the set $\mathcal{Y} = \{\mathbf{y} \in \mathbb{R}^{d_\ell} | A\mathbf{y} \leq \mathbf{b}\}$. Note that in general the constraint set $\mathcal{Y}$ can depend on the UL variable $\mathbf{x} \in \mathcal{X}$. For such cases, $\mathcal{Y}(\mathbf{x})$ is a set valued map $\mathcal{Y} : \mathcal{X} \to \mathbb{R}^{d_\ell}$ and is referred to as a correspondence. However, for the bilevel problem in (1a) and (1b) the correspondence $\mathcal{Y}$ is independent of $\mathbf{x} \in \mathcal{X}$ and is a fixed set. Also, we define

the upper-semi continuity (USC) and the lower-semi continuity (LSC) for the correspondence $\mathcal{Y}(\mathbf{x})$. To define these notions of continuity, we will utilize the notion of an ϵ-ball defined below.

Definition D.1 (ϵ-Ball)**.** For $\mathcal{Y} \subset \mathbb{R}^{d_\ell}$, and given $\epsilon > 0$, we define the open ball about $\mathcal{Y}$ as

$$B_\epsilon(\mathcal{Y}) := \left\{\mathbf{y} \in \mathbb{R}^{d_\ell} \;\middle|\; \|\mathbf{y} - \mathbf{y}'\| < \epsilon, \;\text{ for some } \mathbf{y}' \in \mathcal{Y}\right\},$$

where $\|\cdot\|$ is the standard Euclidean norm.

Using the ϵ-ball we define the Upper Semi-Continuity (USC) of the correspondence $\mathcal{Y}$.

Definition D.2 (Upper Semi-Continuity (USC))**.** The correspondence $\mathcal{Y} : \mathcal{X} \to \mathbb{R}^{d_\ell}$ is USC if for every $\mathbf{x} \in \mathcal{X}$ and $\epsilon > 0$, there exists a $\delta > 0$ such that $\mathcal{Y}(\mathbf{x}') \subset B_\epsilon(\mathcal{Y}(\mathbf{x})))$, if $\mathbf{x}' \in \mathcal{X}$ and $\|\mathbf{x} - \mathbf{x}'\| < \delta$.

Next, we define the notion of Lower Semi-Continuity (LSC).

Definition D.3 (Lower Semi-Continuity (LSC))**.** The correspondence $\mathcal{Y} : \mathcal{X} \to \mathbb{R}^{d_\ell}$ is LSC if for any sequence $\mathbf{x}_n$ in $\mathcal{X}$ that converges to a point $\mathbf{x} \in \mathcal{X}$, and $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$, there exists a sequence $\mathbf{y}_n$ such that $\mathbf{y}_n \in \mathcal{Y}(\mathbf{x}_n)$, for all $n \in \mathbb{N}$, and $\lim_{n\to\infty} \mathbf{y}_n = \mathbf{y}$.

Theorem D.4 (Berge's Theorem of Maximum (Lecture, 2017))**.** *Let $\mathcal{X} \subset \mathbb{R}^{d_u}$ be a non-empty set. Also, let $\mathcal{Y} : \mathcal{X} \to \mathbb{R}^{d_\ell}$ be a correspondence such that the set $\mathcal{Y}(\mathbf{x})$ is compact and non-empty for all $\mathbf{x} \in \mathcal{X}$, and $\mathcal{Y}$ is USC and LSC. Then, if $g : \mathcal{X} \times \mathbb{R}^{d_\ell} \to \mathbb{R}$ is a continuous function with $\overline{\mathbf{y}}^*(\mathbf{x})$ defined as*

$$\overline{\mathbf{y}}^*(\mathbf{x}) \in \operatorname*{arg\,min}_{\mathbf{y}\in\mathbb{R}^{d_\ell}} \left\{g(\mathbf{x}, \mathbf{y}) \;\middle|\; \mathbf{y} \in \mathcal{Y}(\mathbf{x})\right\},$$

the correspondence $\overline{\mathbf{y}}^*(\mathbf{x})$ is non-empty for all $\mathbf{x} \in \mathcal{X}$, and USC.

Remark D.5. If $\overline{\mathbf{y}}^*(\mathbf{x})$ is singleton, then USC implies the continuity of the map $\overline{\mathbf{y}}^*(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$. □

Next, we present the proof of Proposition 2.2.

Proof. The proof of proposition 2.2 follows from the application of Berge's theorem.

To begin with, note that for our problem the set $\mathcal{Y}$ is a fixed set independent of $\mathbf{x} \in \mathcal{X}$. We are going to verify the conditions of Theorem D.4. First, note from Assumption 2.1(b) that the set $\mathcal{Y}$ is non-empty and compact. Then, it is easy to see that $\mathcal{Y} \subset B_\epsilon(\mathcal{Y})$, for every $\epsilon > 0$, and that implies the USC of $\mathcal{Y}$. Moreover, since the set $\mathcal{Y}$ is independent of $\mathbf{x} \in \mathcal{X}$ and compact, for every sequence $\mathbf{x}_n \to \mathbf{x}$ in $\mathcal{X}$, we can always find a sequence $\mathbf{y}_n \to \mathbf{y}$, such that $\mathbf{y}_n, \mathbf{y} \in \mathcal{Y}$. Therefore, $\mathcal{Y}$ is LSC. Finally, using Assumption 2.1(a) we see that the function $g(\mathbf{x}, \mathbf{y})$ is continuous. Then, Theorem D.4 implies that the set $\overline{\mathbf{y}}^*(\mathbf{x})$ is non-empty and the correspondence USC.

Using the strong-convexity of $g(\mathbf{x}, \mathbf{y})$ with respect to $\mathbf{y}$ (Assumption 2.1(c)) we claim that $\overline{\mathbf{y}}^*(\mathbf{x})$ will be a singleton, and thereby a continuous mapping. Then, the continuity of $\overline{\mathbf{y}}^*(\mathbf{x})$ implies the continuity of $G(\mathbf{x}) := f(\mathbf{x}, \overline{\mathbf{y}}^*(\mathbf{x}))$, since the composition of two continuous functions is continuous. The proof is now complete. □

D.1.2. Proof of Lemma 2.4

Proof. In this proof we follow a reasoning similar to Parise & Ozdaglar (2017, Thm. 1). However, differently from that work we consider bilevel problems rather than Nash games. To begin with, consider the Lagrangian of problem (5), i.e.,

$$L(\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}) = g(\mathbf{x}, \mathbf{y}) + \boldsymbol{\lambda}^T \left(A\mathbf{y} - \mathbf{b}\right).$$

Then, for some fixed $\mathbf{x} \in \mathcal{X}$, consider a KKT point $(\mathbf{y}^*(\mathbf{x}), \boldsymbol{\lambda}^*(\mathbf{x}))$ of (5), for which it holds that,

- $\nabla_y L(\mathbf{x}, \mathbf{y}^*(\mathbf{x}), \boldsymbol{\lambda}^*(\mathbf{x})) = \nabla_y g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) + A^T \boldsymbol{\lambda}^*(\mathbf{x}) = 0$
- $[\boldsymbol{\lambda}^*(\mathbf{x})]^T \left(A\mathbf{y}^*(\mathbf{x}) - \mathbf{b}\right) = 0$
- $\boldsymbol{\lambda}^*(\mathbf{x}) \geq 0$
- $A\mathbf{y}^*(\mathbf{x}) - \mathbf{b} \leq 0.$

Now, consider the active constraints at $(\mathbf{y}^*(\mathbf{x}), \boldsymbol{\lambda}^*(\mathbf{x}))$, and to simplify notation let us set $\overline{A} := \overline{A}(\mathbf{y}^*(\mathbf{x}))$. Using the notations defined in Section 2 and the SC property, the KKT conditions given above can be equivalently rewritten as

$$\nabla_y g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) + \overline{A}^T \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) = 0, \quad \overline{A}\mathbf{y}^*(\mathbf{x}) - \overline{\mathbf{b}} = 0, \quad \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) > 0, \tag{22}$$

where $\overline{\boldsymbol{\lambda}}^*(\mathbf{x})$ is the subvector of $\boldsymbol{\lambda}^*(\mathbf{x})$ that contains only the elements whose indices correspond to the active constraints at $\mathbf{y} = \mathbf{y}^*(\mathbf{x})$. Moreover, notice that the point $(\mathbf{y}^*(\mathbf{x}), \boldsymbol{\lambda}^*(\mathbf{x}))$ is unique. The uniqueness of $\mathbf{y}^*(\mathbf{x})$ follows from the strong convexity of $g(\mathbf{x}, \cdot)$; the uniqueness of $\boldsymbol{\lambda}^*(\mathbf{x})$ results from the fact that matrix $\overline{A}$ has full row rank (which guarantees regularity, e.g., see (Bertsekas, 1998)).

As mentioned in section 2, the SC condition (from Lemma 2.3) combined with Assumption 2.1 implies that the mapping $\mathbf{y}^*(\mathbf{x})$ is differentiable almost surely (Friesz & Bernstein, 2015, Theorem 2.22). As a result, at any given point $\mathbf{x}$, we can consider a sufficiently small neighborhood around it, such that the active constraints $\overline{A}$ remain unchanged. Then, we can compute the gradient of (22) using the implicit function theorem as follows

$$\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) + \nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \nabla \mathbf{y}^*(\mathbf{x}) + \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) = 0 \tag{23}$$

$$\overline{A} \nabla \mathbf{y}^*(\mathbf{x}) = 0. \tag{24}$$

Solving the (23) for $\nabla \mathbf{y}^*(\mathbf{x})$ yields

$$\nabla \mathbf{y}^*(\mathbf{x}) = \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \left[-\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right], \tag{25}$$

where we exploited the fact that the Hessian matrix $\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ is positive definite and thus invertible. Substituting (25) into (24) gives

$$\begin{aligned} &\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \left[-\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right] = 0 \\ \implies &\nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) = -\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))^{-1}\right]\overline{A}^T\right]^{-1} \left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]. \end{aligned}$$

Finally, note that the KKT point $\mathbf{y}^*(\mathbf{x})$ corresponds to the unique global minimum of (5), due to the strong convexity of $g(\mathbf{x}, \cdot)$. The proof is now complete. □

D.1.3. The proof of Lemma 2.8

The proof of Lemma 2.8 requires several intermediate results which we provide below. Note that under Assumption 2.5(c) it holds that $\overline{A}(\mathbf{y}^*(\mathbf{x})) = \overline{A}(\widehat{\mathbf{y}}(\mathbf{x}))$; for simplicity we will denote these matrices as $\overline{A}$ in the derivations of this subsection. Moreover, for any given matrix A we will denote with L_A the maximum value of the quantity $\|\overline{A}\left(\widehat{\mathbf{y}}(\mathbf{x})\right)\|$, across all $\mathbf{x} \in \mathcal{X}$.

Lemma D.6. *Suppose that Assumption 2.1,2.5,2.7 hold. Then for any $\mathbf{x} \in \mathcal{X}$, we have:*

(a) $\left\|\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y})\right]^{-1}\right\| \leq \frac{1}{\mu_g}, \ \forall \mathbf{y} \in \mathbb{R}^{d_\ell}$.

(b) $\left\|\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\right\| \leq \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}} \delta$.

(c) $\left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y})\right]^{-1} \overline{A}^T\right]^{-1}\right\| \leq \overline{L}_A, \ \forall \mathbf{y} \in \mathbb{R}^{d_\ell}$.

(d) $\left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \overline{A}^T\right]^{-1} - \left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \overline{A}^T\right]^{-1}\right\| \leq L_A^2 \overline{L}_A^2 \frac{1}{\mu_g^2} L_{g_{yy}} \delta$.

Proof. a) We know that $g(\mathbf{x}, \mathbf{y})$ is strongly convex in $\mathbf{y}$ with modulus μ_g. Therefore, for any $\mathbf{x} \in \mathcal{X}$ we have

$$\begin{aligned} &\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}) \succeq \mu_g I \succ 0, \forall \mathbf{y} \in \mathbb{R}^{d_\ell} \\ \implies &0 \prec \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y})\right]^{-1} \preceq \frac{1}{\mu_g} I, \forall \mathbf{y} \in \mathbb{R}^{d_\ell} \\ \implies &\left\|\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y})\right]^{-1}\right\| \leq \frac{1}{\mu_g}, \forall \mathbf{y} \in \mathbb{R}^{d_\ell}. \end{aligned} \tag{26}$$

b) To begin with, notice that for arbitrary square invertible matrices P, Q we have

$$\left\|P^{-1}-Q^{-1}\right\| = \left\|P^{-1}(Q-P)Q^{-1}\right\| \leq \left\|Q^{-1}(P-Q)\right\| \left\|P^{-1}\right\| \leq \left\|Q^{-1}\right\| \|P-Q\| \left\|P^{-1}\right\|. \tag{27}$$

Then, using the above inequality we get

$$\begin{aligned}
&\left\|\left[\nabla^2_{yy} g(\mathbf{x},\mathbf{y}^*(\mathbf{x}))\right]^{-1} - \left[\nabla^2_{yy} g(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\right\| \\
&\leq \left\|\left[\nabla^2_{yy} g(\mathbf{x},\mathbf{y}^*(\mathbf{x}))\right]^{-1}\right\| \left\|\nabla^2_{yy} g(\mathbf{x},\mathbf{y}^*(\mathbf{x})) - \nabla^2_{yy} g(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x}))\right\| \left\|\left[\nabla^2_{yy} g(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\right\| \\
&\leq \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}} \|\mathbf{y}^*(\mathbf{x}) - \widehat{\mathbf{y}}(\mathbf{x})\| \\
&\leq \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}} \delta,
\end{aligned}$$

where in the second inequality we used the result from Lemma D.6(a) and the Lipschitz Hessian property of g in $\mathbf{yy}$ (Assumption 2.7(d)); in the third inequality we use the Assumption 2.5(a) for $\mathbf{y}(\mathbf{x})$.

c) In our problem we have that g strongly convex in $\mathbf{y}$ and Lipschitz gradient in $\mathbf{y}$. Thus, for any $\mathbf{x} \in \mathcal{X}$ we have

$$\begin{aligned}
&L_y I \succeq \nabla^2_{yy} g(\mathbf{x},\mathbf{y}^*) \succeq \mu_g I \succ 0 \\
\Longrightarrow\ & 0 \prec \frac{1}{L_y} I \preceq \left[\nabla^2_{yy} g(\mathbf{x},\mathbf{y})\right]^{-1} \preceq \frac{1}{\mu_g} I, \forall \mathbf{y} \in \mathbb{R}^{d_\ell}.
\end{aligned}$$

Also, for every $\mathbf{x} \in \mathcal{X}$, we have that

$$\left\|\overline{A}^T \mathbf{z}\right\|^2 = \mathbf{z}^T \overline{A}\overline{A}^T \mathbf{z} \geq \lambda_{min}(\overline{A}\overline{A}^T)\|\mathbf{z}\|^2, \forall \mathbf{z} \in \mathbb{R}^{d_\ell}.$$

Using the above two lower bound we get

$$\mathbf{z}^T \overline{A} \left[\nabla^2_{yy} g(\mathbf{x},\mathbf{y})\right]^{-1} \overline{A}^T \mathbf{z} \geq \frac{1}{L_y} \lambda_{min}(\overline{A}\overline{A}^T)\|\mathbf{z}\|^2 > 0, \forall \mathbf{z} \in \mathbb{R}^{d_\ell} \setminus \{0\},$$

where the last inequality follows from the fact that $\overline{A}$ is full row rank which implies that $\lambda_{min}(\overline{A}\overline{A}^T) > 0, \forall \mathbf{x} \in \mathcal{X}$.

Since the above inequality holds for every $\mathbf{z} \in \mathbb{R}^{d_\ell} \setminus \{0\}$, it $\forall \mathbf{x} \in \mathcal{X}$ we get that

$$\begin{aligned}
\overline{A} \left[\nabla^2_{yy} g(\mathbf{x},\mathbf{y})\right]^{-1} \overline{A}^T &\succeq \frac{\lambda_{min}(\overline{A}\overline{A}^T)}{L_y} I \succ 0 \\
\left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x},\mathbf{y})\right]^{-1} \overline{A}^T\right]^{-1} &\preceq \frac{L_y}{\lambda_{min}(\overline{A}\overline{A}^T)} I \\
\left\|\left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x},\mathbf{y})\right]^{-1} \overline{A}^T\right]^{-1}\right\| &\leq \frac{L_y}{\lambda_{min}(\overline{A}\overline{A}^T)}.
\end{aligned}$$

Finally, for the given matrix A, consider the submatrix $\overline{A} = \overline{A}(\widehat{\mathbf{y}}(\mathbf{x}))$ generated by considering only the subset of its rows corresponding to the active constraints at $\widehat{\mathbf{y}}(\mathbf{x})$. From Assumption 2.1(c) we know that $\overline{A}(\widehat{\mathbf{y}}(\mathbf{x}))$ is full row rank for every $\mathbf{x} \in \mathcal{X}$, and so we can ensure that $\lambda_{min}\left(\overline{A}\left(\widehat{\mathbf{y}}(\mathbf{x})\right)\overline{A}\left(\widehat{\mathbf{y}}(\mathbf{x})\right)^T\right) > 0, \forall \mathbf{x} \in \mathcal{X}$. Then, we denote with λ_{min} the minimum value of the quantity $\lambda_{min}\left(\overline{A}\left(\widehat{\mathbf{y}}(\mathbf{x})\right)\overline{A}\left(\widehat{\mathbf{y}}(\mathbf{x})\right)^T\right)$ across all $\mathbf{x} \in \mathcal{X}$. Therefore, we conclude that

$$\left\|\left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x},\mathbf{y})\right]^{-1} \overline{A}^T\right]^{-1}\right\| \leq \frac{L_y}{\lambda_{min}} := \overline{L}_A.$$

d) Applying formula (27) with $P = \overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\overline{A}^T, Q = \overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\overline{A}^T$ we get

$$\begin{aligned}
&\left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1} - \left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\right\| \\
&\leq \left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\right\| \left\|\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\overline{A}^T - \overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\overline{A}^T\right\| \\
&\qquad\qquad \left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\right\| \\
&\leq \left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\right\| \left\|\overline{A}\left[\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{y}(\mathbf{x}))\right]^{-1}\right]\overline{A}^T\right\| \\
&\qquad\qquad \left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\right\| \\
&\leq \left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\right\| \|\overline{A}\| \left\|\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\right\| \left\|\overline{A}^T\right\| \\
&\qquad\qquad \left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\right\| \\
&\leq L_A^2 \overline{L}_A^2 \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}}\delta,
\end{aligned}$$

where in the final inequality we used the bounds derived in Lemma D.6(b), D.6(c), and the bound $\|\overline{A}\| \leq L_A$.

The proof is now complete. □

Now let us bound the norm of the gradients of the mappings $\boldsymbol{\lambda}^*(\mathbf{x})$ and $\mathbf{y}^*(\mathbf{x})$.

Lemma D.7. *Under Assumptions 2.1,2.5,2.7, the gradients of the mappings $\boldsymbol{\lambda}^*(\mathbf{x})$ and $\mathbf{y}^*(\mathbf{x})$ satisfy the following bounds for every $\mathbf{x} \in \mathcal{X}$,*

$$\begin{aligned}
&\|\nabla\overline{\boldsymbol{\lambda}}^*(\mathbf{x})\| \leq \overline{L}_{\boldsymbol{\lambda}^*}, \quad \left\|\widehat{\nabla}\overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right\| \leq \overline{L}_{\boldsymbol{\lambda}^*} \\
&\|\nabla\mathbf{y}^*(\mathbf{x})\| \leq \overline{L}_{\mathbf{y}^*}, \quad \|\widehat{\nabla}\mathbf{y}^*(\mathbf{x})\| \leq \overline{L}_{\mathbf{y}^*}
\end{aligned}$$

where $\overline{L}_{\boldsymbol{\lambda}^*} = \frac{1}{\mu_g}\overline{L}_A L_A \overline{L}_{g_{xy}}$ and $\overline{L}_{\mathbf{y}^*} = \frac{1}{\mu_y}\left(\overline{L}_{g_{xy}} + L_A\overline{L}_{\boldsymbol{\lambda}^*}\right)$. Note that $\widehat{\nabla}\overline{\boldsymbol{\lambda}}^*(\mathbf{x})$ and $\widehat{\nabla}\mathbf{y}^*(\mathbf{x})$ are obtained by substituting the estimate $\widehat{\mathbf{y}}(\mathbf{x})$ in place of $\mathbf{y}^*(\mathbf{x})$ in the expressions $\nabla\overline{\boldsymbol{\lambda}}^*(\mathbf{x})$ and $\nabla\mathbf{y}^*(\mathbf{x})$, respectively (Please see Lemma 2.4).

Proof. From Lemma 2.4 we have

$$\nabla\overline{\boldsymbol{\lambda}}^*(\mathbf{x}) = -\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]$$

Then, taking the norm of this quantity we get,

$$\begin{aligned}
\left\|\nabla\overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right\| &= \left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]\right\| \\
&\leq \left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\right\| \|\overline{A}\| \left\|\left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}\right\| \|\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| \\
&\leq \overline{L}_A L_A \frac{1}{\mu_g}\overline{L}_{g_{xy}} := \overline{L}_{\boldsymbol{\lambda}^*},
\end{aligned}$$

where in the last inequality we used Lemma D.6(a), D.6(c) and Assumption 2.7(f).

Similarly, for $\|\widehat{\nabla}\overline{\boldsymbol{\lambda}}^*(\mathbf{x})\|$ we have that

$$\begin{aligned}
\left\|\widehat{\nabla}\overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right\| &= \left\|\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\overline{A}^T\right]^{-1}\left[\overline{A}\left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\nabla^2_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]\right\| \\
&\leq \overline{L}_A L_A \frac{1}{\mu_g}\overline{L}_{g_{xy}} = \overline{L}_{\boldsymbol{\lambda}^*}.
\end{aligned}$$

Moving to the bound of $\|\nabla \mathbf{y}^*(\mathbf{x})\|$, we know from Lemma 2.4 that the formula of the gradient of $\mathbf{y}^*(\mathbf{x})$ is

$$\nabla \mathbf{y}^*(\mathbf{x}) = \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \left[-\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right]. \tag{28}$$

Then, we have that

$$\begin{aligned}
\|\nabla \mathbf{y}^*(\mathbf{x})\| &= \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \left[-\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right] \right\| \\
&\leq \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \right\| \left\| \left[-\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right] \right\| \\
&\leq \frac{1}{\mu_g} \left(\|\nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| + \|\overline{A}\| \left\|\nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right\| \right) \\
&\leq \frac{1}{\mu_g} \left(\overline{L}_{g_{xy}} + L_A \overline{L}_{\boldsymbol{\lambda}^*} \right) := \overline{L}_{\mathbf{y}^*},
\end{aligned}$$

where in the second inequality we used we used Lemma D.6(a); the third inequality follows from Assumption 2.7(f) and the bound for $\left\|\nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right\|$ we derived above.

Similarly, for $\left\|\widehat{\nabla} \mathbf{y}(\mathbf{x})\right\|$ we can obtain the following bound

$$\begin{aligned}
\left\|\widehat{\nabla} \mathbf{y}(\mathbf{x})\right\| &= \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \left[-\nabla^2_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) - \overline{A}^T \widehat{\nabla} \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right] \right\| \\
&\leq \frac{1}{\mu_g} \left(\|\nabla^2_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\| + \|\overline{A}\| \left\|\widehat{\nabla} \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right\| \right) \\
&\leq \frac{1}{\mu_g} \left(\overline{L}_{g_{xy}} + L_A \overline{L}_{\boldsymbol{\lambda}^*} \right) = \overline{L}_{\mathbf{y}^*}.
\end{aligned}$$

The proof is now complete. □

In the next two results we are going to present bounds for the difference of the exact and approximate gradients of the mappings $\overline{\boldsymbol{\lambda}}^*(\mathbf{x})$ and $\nabla \mathbf{y}^*(\mathbf{x})$.

Lemma D.8. *Suppose that Assumptions 2.1,2.5,2.7 hold. Then, the following bound holds*

$$\|\nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) - \widehat{\nabla} \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\| \leq L_{\boldsymbol{\lambda}^*} \delta,$$

where $L_{\boldsymbol{\lambda}^*} = \left(\frac{1}{\mu_g}\right)^3 \overline{L}_A^2 L_A^3 L_{g_{yy}} \overline{L}_{g_{xy}} + \frac{1}{\mu_g} \overline{L}_A L_A L_{g_{xy}} + \left(\frac{1}{\mu_g}\right)^2 \overline{L}_A L_A L_{g_{yy}} \overline{L}_{g_{xy}}$.

Proof. Using the derivation of $\nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})$ from Lemma 2.4, and its approximation $\widehat{\nabla} \overline{\boldsymbol{\lambda}}^*(\mathbf{x})$ where we substitute $\mathbf{y}^*(\mathbf{x})$ with $\widehat{\mathbf{y}}(\mathbf{x})$ in the formula of the former, that is,

$$\widehat{\nabla} \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) = -\left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \overline{A}^T\right]^{-1} \left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \nabla^2_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right],$$

we obtain

$$\begin{aligned}
\left\|\nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) - \widehat{\nabla} \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right\| = \Bigg\| &\left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \overline{A}^T\right]^{-1} \left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right] \\
&- \left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \overline{A}^T\right]^{-1} \left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \nabla^2_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right] \Bigg\|.
\end{aligned}$$

Below, we use the following notation in order to simplify the derivations.

$H(\mathbf{x}) = \left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \overline{A}^T\right]$, $G(\mathbf{x}) = \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1}$, $M(\mathbf{x}) = \nabla^2_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$

$\widehat{H}(\mathbf{x}) = \left[\overline{A} \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \overline{A}^T\right]$, $\widehat{G}(\mathbf{x}) = \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}$, $\widehat{M}(\mathbf{x}) = \nabla^2_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))$

Then, we have that

$$
\begin{aligned}
\|\nabla\overline{\boldsymbol{\lambda}}^*(\mathbf{x}) - \widehat{\nabla}\overline{\boldsymbol{\lambda}}^*(\mathbf{x})\| &= \left\|H^{-1}(\mathbf{x})\overline{A}G(\mathbf{x})M(\mathbf{x}) - \widehat{H}^{-1}(\mathbf{x})\overline{A}\widehat{G}(\mathbf{x})\widehat{M}(\mathbf{x})\right\| \\
&\overset{(a)}{\leq} \left\|H^{-1}(\mathbf{x})\overline{A}G(\mathbf{x})M(\mathbf{x}) - \widehat{H}^{-1}(\mathbf{x})\overline{A}G(\mathbf{x})M(\mathbf{x})\right\| \\
&+ \left\|\widehat{H}^{-1}(\mathbf{x})\overline{A}G(\mathbf{x})M(\mathbf{x}) - \widehat{H}^{-1}(\mathbf{x})\overline{A}\widehat{G}(\mathbf{x})\widehat{M}(\mathbf{x})\right\| \\
&\leq \left\|H^{-1}(\mathbf{x}) - \widehat{H}^{-1}(\mathbf{x})\right\| \|\overline{A}\| \|G(\mathbf{x})\| \|M(\mathbf{x})\| \\
&+ \left\|\widehat{H}^{-1}(\mathbf{x})\right\| \|\overline{A}\| \left\|G(\mathbf{x})M(\mathbf{x}) - \widehat{G}(\mathbf{x})\widehat{M}(\mathbf{x})\right\| \\
&\overset{(b)}{\leq} \left\|H^{-1}(\mathbf{x}) - \widehat{H}^{-1}(\mathbf{x})\right\| \|\overline{A}\| \|G(\mathbf{x})\| \|M(\mathbf{x})\| \\
&+ \left\|\widehat{H}^{-1}(\mathbf{x})\right\| \|\overline{A}\| \left[\left\|G(\mathbf{x})M(\mathbf{x}) - G(\mathbf{x})\widehat{M}(\mathbf{x})\right\| + \left\|G(\mathbf{x})\widehat{M}(\mathbf{x}) - \widehat{G}(\mathbf{x})\widehat{M}(\mathbf{x})\right\|\right] \\
&\leq \left\|H^{-1}(\mathbf{x}) - \widehat{H}^{-1}(\mathbf{x})\right\| \|\overline{A}\| \|G(\mathbf{x})\| \|M(\mathbf{x})\| \\
&+ \left\|\widehat{H}^{-1}(\mathbf{x})\right\| \|\overline{A}\| \|G(\mathbf{x})\| \left\|M(\mathbf{x}) - \widehat{M}(\mathbf{x})\right\| + \left\|\widehat{H}^{-1}(\mathbf{x})\right\| \|\overline{A}\| \left\|G(\mathbf{x}) - \widehat{G}(\mathbf{x})\right\| \left\|\widehat{M}(\mathbf{x})\right\| \\
&\overset{(c)}{\leq} \overline{L}_A^2 L_A^2 \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}}\delta L_A \frac{1}{\mu_g}\overline{L}_{g_{xy}} + \overline{L}_A L_A \frac{1}{\mu_g} L_{g_{xy}}\delta + \overline{L}_A L_A \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}}\delta\overline{L}_{g_{xy}} \\
&= \left(\left(\frac{1}{\mu_g}\right)^3 \overline{L}_A^2 L_A^3 L_{g_{yy}}\overline{L}_{g_{xy}} + \frac{1}{\mu_g}\overline{L}_A L_A L_{g_{xy}} + \left(\frac{1}{\mu_g}\right)^2 \overline{L}_A L_A L_{g_{yy}}\overline{L}_{g_{xy}}\right)\delta.
\end{aligned}
$$

In (a) we add and subtract the term $\widehat{H}^{-1}(\mathbf{x})\overline{A}G(\mathbf{x})M(\mathbf{x})$ and apply the triangle inequality. In (b) we add and subtract the term $G(\mathbf{x})\widehat{M}(\mathbf{x})$ and apply the triangle inequality. In (c) we use Lemma D.6(d) for $\|H^{-1}(\mathbf{x}) - \widehat{H}^{-1}(\mathbf{x})\|$, the bound $\|\overline{A}\| \leq L_A$, Lemma D.6(a) for $\|G(\mathbf{x})\|$, Lemma D.6(c) for $\|H^{-1}(\mathbf{x})\|$ and $\|\widehat{H}^{-1}(\mathbf{x})\|$, Assumption 2.7(f) for $\|M(\mathbf{x})\|$ and $\|\widehat{M}(\mathbf{x})\|$, Assumption 2.7(e) for $\|M(\mathbf{x}) - \widehat{M}(\mathbf{x})\|$, and finally Lemma D.6(b) for $\|G(\mathbf{x}) - \widehat{G}(\mathbf{x})\|$.

The proof is now complete. □

Lemma D.9. *Suppose that Assumptions 2.1,2.5,2.7 hold. Then, the following bound holds*

$$\|\nabla\mathbf{y}^*(\mathbf{x}) - \widehat{\nabla}\mathbf{y}(\mathbf{x})\| \leq L_{\mathbf{y}^*}\delta,$$

where $L_{\mathbf{y}^*} = \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}}\overline{L}_{g_{xy}} + \frac{1}{\mu_g}L_{g_{xy}} + \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}}L_A\overline{L}_{\boldsymbol{\lambda}^*} + \frac{1}{\mu_g}L_A L_{\boldsymbol{\lambda}^*}$.

Proof. From Lemma 2.4 we have that

$$\nabla\mathbf{y}^*(\mathbf{x}) = \left[\nabla^2_{yy}g(\mathbf{x},\mathbf{y}^*(\mathbf{x}))\right]^{-1}\left[-\nabla^2_{xy}g(\mathbf{x},\mathbf{y}^*(\mathbf{x})) - \overline{A}^T\nabla\overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right].$$

We can also get $\widehat{\nabla}\mathbf{y}(\mathbf{x})$ by substituting $\widehat{\mathbf{y}}(\mathbf{x})$ in place of $\mathbf{y}^*(\mathbf{x})$ in the above formula, i.e.,

$$\widehat{\nabla}\mathbf{y}(\mathbf{x}) = \left[\nabla^2_{yy}g(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1}\left[-\nabla^2_{xy}g(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x})) - \overline{A}^T\widehat{\nabla}\overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right].$$

Then, we have that

$$
\begin{aligned}
\left\|\nabla \mathbf{y}^*(\mathbf{x}) - \widehat{\nabla}\mathbf{y}(\mathbf{x})\right\| &= \Big\| \left[\nabla_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \left[-\nabla_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right] \\
&\quad - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \left[-\nabla_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) - \overline{A}^T \widehat{\nabla} \overline{\boldsymbol{\lambda}}^*(\mathbf{x})\right] \Big\| \\
&\overset{(a)}{\leq} \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \nabla_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \nabla_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) \right\| \\
&+ \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \overline{A}^T \widehat{\nabla} \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) \right\| \\
&\overset{(b)}{\leq} \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \nabla_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \left[\nabla_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \nabla_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \right\| \\
&+ \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \nabla_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \nabla_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) \right\| \\
&+ \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) \right\| \\
&+ \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \overline{A}^T \widehat{\nabla} \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) \right\| \\
&\leq \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \right\| \left\| \nabla_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \right\| \\
&+ \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \right\| \left\| \nabla_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - \nabla_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) \right\| \\
&+ \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right]^{-1} - \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \right\| \left\| \overline{A}^T \right\| \left\| \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) \right\| \\
&+ \left\| \left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \right\| \left\| \overline{A}^T \right\| \left\| \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) - \widehat{\nabla} \overline{\boldsymbol{\lambda}}^*(\mathbf{x}) \right\| \\
&\overset{(c)}{\leq} \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}} \delta \overline{L}_{g_{xy}} + \frac{1}{\mu_g} L_{g_{xy}} \delta + \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}} \delta L_A \overline{L}_{\boldsymbol{\lambda}^*} + \frac{1}{\mu_g} L_A L_{\boldsymbol{\lambda}^*} \delta \\
&= \left(\left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}} \overline{L}_{g_{xy}} + \frac{1}{\mu_g} L_{g_{xy}} + \left(\frac{1}{\mu_g}\right)^2 L_{g_{yy}} L_A \overline{L}_{\boldsymbol{\lambda}^*} + \frac{1}{\mu_g} L_A L_{\boldsymbol{\lambda}^*} \right) \delta.
\end{aligned}
$$

In (a) the triangle inequality was used. In (b) we add and subtract the expressions $\left[\nabla_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \nabla_{xy} g(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))$ and $\left[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right]^{-1} \overline{A}^T \nabla \overline{\boldsymbol{\lambda}}^*(\mathbf{x})$ in the first and second terms, respectively. In (c) we apply Lemma D.6(a), D.6(b), Assumption 2.7(e), 2.7(f), the bound $\|\overline{A}\| \leq L_A$, Lemmas D.7 and D.8.

The proof is now complete. □

Now we have all the results needed to prove Lemma 2.8.

Proof of Lemma 2.8. To begin with, the exact and approximate (due to the inexact solution of the LL problem) implicit gradients of the objective $F(\mathbf{x})$, are given below.

$$
\begin{aligned}
\nabla F(\mathbf{x}) &= \nabla_x f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) + \left[\nabla \mathbf{y}^*(\mathbf{x})\right]^T \nabla_y f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) \\
\widehat{\nabla} F(\mathbf{x}) &= \nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) + \left[\widehat{\nabla}\mathbf{y}(\mathbf{x})\right]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})).
\end{aligned}
$$

Then, we can compute the norm of their difference.

$$
\begin{aligned}
\|\widehat{\nabla} F(\mathbf{x}) - \nabla F(\mathbf{x})\| &= \|\nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) + [\nabla \widehat{\mathbf{y}}(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) \\
&\qquad - \nabla_x f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) - [\nabla \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| \\
&\overset{(a)}{\leq} \|\nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) - \nabla_x f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| \\
&\qquad + \|[\widehat{\nabla} \mathbf{y}(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) - [\nabla \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| \\
&\overset{(b)}{\leq} \|\nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) - \nabla_x f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| \\
&\qquad + \|[\widehat{\nabla} \mathbf{y}(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) - [\nabla \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\| \\
&\qquad + \| [\nabla \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) - [\nabla \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| \\
&\overset{(c)}{\leq} L_f \|\widehat{\mathbf{y}}(\mathbf{x}) - \mathbf{y}^*(\mathbf{x}))\| + \|\widehat{\nabla} \mathbf{y}(\mathbf{x}) - \nabla \mathbf{y}^*(\mathbf{x}))\| \|\nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\| \\
&\qquad + \|\nabla \mathbf{y}^*(\mathbf{x}))\| \|\nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) - \nabla_y f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| \\
&\overset{(d)}{\leq} L_f \|\widehat{\mathbf{y}}(\mathbf{x}) - \mathbf{y}^*(\mathbf{x}))\| + \|\widehat{\nabla} \mathbf{y}(\mathbf{x}) - \nabla \mathbf{y}^*(\mathbf{x}))\| \|\nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\| \\
&\qquad + L_f \|\nabla \mathbf{y}^*(\mathbf{x}))\| \|\widehat{\mathbf{y}}(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\| \\
&\overset{(e)}{\leq} L_f \delta + L_{\mathbf{y}^*} \delta \overline{L}_f + L_f \overline{L}_{\mathbf{y}^*} \delta \\
&= \left(L_f + L_{\mathbf{y}^*} \overline{L}_f + L_f \overline{L}_{\mathbf{y}^*}\right) \delta := L_F \delta,
\end{aligned} \tag{29}
$$

where $L_F = L_f + L_{\mathbf{y}^*} \overline{L}_f + L_f \overline{L}_{\mathbf{y}^*}$. Also, in inequality (a) above we apply the triangle inequality; in (b) we add and subtract the term $[\nabla \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))$, and use triangle inequality; in (c) and (d) we use the Lipschitz gradient property of f (Assumption 2.5(b)); in (e) we apply Assumptions 2.7(a) and 2.5(a), and Lemmas D.7 and D.9.

Now consider the expression $\|\nabla F(\mathbf{x})\|$. We have that

$$
\begin{aligned}
\|\nabla F(\mathbf{x})\| &= \left\|\nabla_x f(\mathbf{x}, \mathbf{y}^*(\mathbf{x})) + [\nabla \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\right\| \\
&\leq |\nabla_x f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| + \|\nabla \mathbf{y}^*(\mathbf{x})\| \, \|\nabla_y f(\mathbf{x}, \mathbf{y}^*(\mathbf{x}))\| \\
&\leq \left(1 + \overline{L}_{\mathbf{y}^*}\right) \overline{L}_f := \overline{L}_F,
\end{aligned}
$$

where we applied Assumption 2.7(a) and Lemma D.7. Similarly, we can see that

$$
\begin{aligned}
\left\|\widehat{\nabla} F(\mathbf{x})\right\| &= \left\|\nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) + \left[\widehat{\nabla} \mathbf{y}(\mathbf{x})\right]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\right\| \\
&\leq \|\nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\| + \left\|\widehat{\nabla} \mathbf{y}(\mathbf{x})\right\| \|\nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\| \\
&\leq \left(1 + \overline{L}_{\mathbf{y}^*}\right) \overline{L}_f = \overline{L}_F.
\end{aligned}
$$

Therefore, the proof is completed. □

D.1.4. Proof of Lemma 2.10

Proof. From the definition of the stochastic gradient in (11) we have

$$
\widehat{\nabla} F(\mathbf{x}; \xi) = \nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}); \xi) + [\widehat{\nabla} \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}); \xi).
$$

Taking expectation on both sides and utilizing Assumption 2.9, we get

$$
\begin{aligned}
\mathbb{E}_\xi[\widehat{\nabla} F(\mathbf{x}; \xi)] &= \mathbb{E}_\xi\left[\nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}); \xi) + [\widehat{\nabla} \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}); \xi)\right] \\
&= \mathbb{E}_\xi\left[\nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}); \xi)\right] + [\widehat{\nabla} \mathbf{y}^*(\mathbf{x})]^T \mathbb{E}_\xi\left[\nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}); \xi)\right] \\
&= \nabla_x f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}); \xi) + [\widehat{\nabla} \mathbf{y}^*(\mathbf{x})]^T \nabla_y f(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}); \xi) \\
&= \widehat{\nabla} F(\mathbf{x}).
\end{aligned}
$$

Similarly, for the variance of the stochastic implicit gradient, we have

$$\begin{aligned}\mathbb{E}_\xi\|\widehat{\nabla}F(\mathbf{x};\xi)-\widehat{\nabla}F(\mathbf{x})\|^2 &= \mathbb{E}_\xi\big\|\nabla_x f(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x});\xi)+[\widehat{\nabla}\mathbf{y}^*(\mathbf{x})]^T\nabla_y f(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x});\xi)\\ &\qquad -\big[\nabla_x f(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x}))+[\widehat{\nabla}\mathbf{y}^*(\mathbf{x})]^T\nabla_y f(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x}))\big]\big\|^2\\ &\overset{(a)}{\leq} 2\,\mathbb{E}_\xi\|\nabla_x f(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x});\xi)-\nabla_x f(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x}))\|^2\\ &\qquad +2\,\|\widehat{\nabla}\mathbf{y}^*(\mathbf{x})]\|^2\,\mathbb{E}_\xi\|\nabla_y f(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x});\xi)-\nabla_y f(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x}))\|^2\\ &\overset{(b)}{\leq} 2\sigma_f^2+2\overline{L}_{\mathbf{y}^*}\sigma_f^2 \coloneqq \sigma_F^2,\end{aligned}$$

where (a) follows from $\|\mathbf{x}+\mathbf{y}\|^2\leq 2\|\mathbf{x}\|^2+2\|\mathbf{y}\|^2$ and (b) results from Assumption 2.9 and the application of Lemma D.7.

Therefore, we have the proof. □

D.2. Proofs of Section 3

Example. Before introducing the proofs, we present the following example

$$\min_{x\in[0,1]} x+y^*(x) \qquad \text{where} \qquad y^*(x)\in\arg\min_{y\in\mathbb{R}}\Big\{(x-y)^2\Big|1/2\leq y\leq 1\Big\}$$

Note that for the above problem, we have

$$y^*(x)=\begin{cases}1/2 & \text{for } x\leq 1/2\\ x & \text{for } x>1/2\end{cases}.$$

Therefore, $y^*(x)$ is non-differentiable at $\bar{x}=1/2$.

Now, let us consider the perturbed version of the above problem

$$\min_{x\in[0,1]} x+y^*(x) \qquad \text{where} \qquad y_q^*(x)\in\arg\min_{y\in\mathbb{R}}\Big\{(x-y)^2+qy\Big|1/2\leq y\leq 1\Big\}$$

where q is sampled from a continuous distribution. For the perturbed problem the mapping $y_q^*(x)$ becomes

$$y_q^*(x)=\begin{cases}1/2 & \text{for } x\leq (1+q)/2\\ x & \text{for } x>(1+q)/2\end{cases}.$$

Note that adding the perturbation makes the mapping $y^*(x)$ differentiable at the original non-differentiable point $x=1/2$ and this point of non-differentiability got perturbed to a random point $\bar{\boldsymbol{\epsilon}}(\bar{x};q)=\bar{x}+q/2$. Note that this set mapping $\bar{\boldsymbol{\epsilon}}(\bar{x};q)$ can be empty when the point $\bar{x}+q/2\notin[0,1]$.

Note from the above example we can observe that the mapping $\bar{\boldsymbol{\epsilon}}(\bar{x};q)$ is continuous in q and is a function of the non-differentiable points of the original unperturbed problem.

D.2.1. Proof of Lemma 3.1

In this section, we provide the proof of Lemma 3.1 when the iterates $\{\mathbf{x}^r\}_{r=0}^{\infty}$ are generated using GD updates with constant learning rate. We note that the proof can be easily extended to the iterates generated by Algorithm 1 with the expense of a much more complicated notation.

Proof. Let the iterates $\{\mathbf{x}^r\}_{r=0}^{T}$ with $\mathbf{x}^r\in\mathcal{X}$ be a countable sequence generated according to the following update rule:

$$\mathbf{x}^{r+1}=\mathbf{x}^r-\eta\widehat{\nabla}F(\mathbf{x}^r),$$

where η is a fixed step-size and $\widehat{\nabla}F(\mathbf{x}^r)$ is the approximate implicit gradient defined in (10). First, note that a direct consequence of Lemma 2.3 is that $F(\cdot)$ is differentiable w.p. 1 at iterate $\mathbf{x}^0\in\mathcal{X}$ (since the perturbation is added to the

original bilevel problem (1a) for $\mathbf{x} = \mathbf{x}^0$). Next, we note that $\mathbf{x}^1$ is generated using: $\mathbf{x}^1 = \mathbf{x}^0 - \eta\widehat{\nabla}F(\mathbf{x}^0)$. Naturally, $\mathbf{x}^1$ will be a random variable dependent on the perturbation $\mathbf{q}$. This dependence of $\mathbf{x}^1$ on $\mathbf{q}$ comes through the gradient $\widehat{\nabla}F(\mathbf{x}^1)$, where $\widehat{\nabla}F(\mathbf{x}^1)$ depends on $\mathbf{q}$ through the active set $\overline{A}$ and the mapping $\widehat{\mathbf{y}}(\mathbf{x}^1)$. Please recall that $\widehat{\mathbf{y}}(\mathbf{x}^1)$ is the approximate solution of the perturbed LL problem in (5). Also recall that the definition of $\widehat{\nabla}F(\mathbf{x}^1)$ was given in (10), and provided below for convenience:

$$\widehat{\nabla}F(\mathbf{x}^1) = \nabla_x f(\mathbf{x}^1, \widehat{\mathbf{y}}(\mathbf{x}^1)) + [\widehat{\nabla}\mathbf{y}^*(\mathbf{x}^1)]^T \nabla_y f(\mathbf{x}^1, \widehat{\mathbf{y}}(\mathbf{x}^1)), \tag{30}$$

where

$$\widehat{\nabla}\mathbf{y}^*(\mathbf{x}) = \big[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\big]^{-1} \cdot \big[-\nabla^2_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x})) - \overline{A}^T \widehat{\nabla}\overline{\boldsymbol{\lambda}}^*(\mathbf{x})\big] \tag{31}$$

$$\widehat{\nabla}\overline{\boldsymbol{\lambda}}^*(\mathbf{x}) = -\big[\overline{A}\big[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\big]^{-1}\overline{A}^T\big]^{-1} \cdot \big[\overline{A}\big[\nabla^2_{yy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\big]^{-1}\nabla^2_{xy} g(\mathbf{x}, \widehat{\mathbf{y}}(\mathbf{x}))\big]. \tag{32}$$

Next, we make the following key observation:

- The active set $\overline{A}$ of the LL problem is a discrete random variable with finite support. Note that this is a direct consequence of the fact that as $\mathbf{q}$ varies $\overline{A}$ can only take values in a finite set depending on the set of active constraints at the approximate solution $\widehat{\mathbf{y}}(\mathbf{x}^1)$ of the LL problem.

Now let us look at the expression of the implicit gradient $\widehat{\nabla}F(\mathbf{x}^1)$ given in (30). The set of assumptions stated in Assumption 2.7 implies that $\widehat{\nabla}F(\mathbf{x}^1)$ is a smooth composition of terms involving the active set $\overline{A}$ and the approximate LL solution $\widehat{\mathbf{y}}(\mathbf{x}^1)$ that depend on $\mathbf{q}$. The observation above implies that $\widehat{\nabla}F(\mathbf{x}^1)$ will be random variable with piecewise continuous cumulative distribution function (CDF) since $\widehat{\nabla}F(\mathbf{x}^1)$ is composition of a *discrete* random variable $\overline{A}$ with a (possibly either discrete or continuous) random variable $\widehat{\mathbf{y}}(\mathbf{x}^1)$. We refer to such a random variable as a *mixed* random variable since it is a composition of a discrete random variable ($\overline{A}$) and a potentially continuous random variable ($\widehat{\mathbf{y}}(\mathbf{x}^1)$). This further implies that the iterate $\mathbf{x}^1 = \mathbf{x}^r - \eta\widehat{\nabla}F(\mathbf{x}^r)$ will also be a mixed random variable. Applying this argument successively to iterates $\{\mathbf{x}^r\}_{r=2}^T$, we can conclude that the random variables $\{\mathbf{x}^r\}_{r=2}^T$ will also be mixed random variables since each gradient update is a composition of two mixed random variables.

Next, let us consider the probability of the event that the sequence of random variables $\{\mathbf{x}^r\}_{r=0}^T$ will be non-differentiable. This is equivalent to saying that $\mathbf{x}^r$ for any $r \in [T]$ belongs to the set of non-differentiable points, $\mathcal{X}_F$, of the implicit function $F(\cdot)$. Recall that from the Assumption in Lemma 3.1 the set of non-differentiable points of $F(\cdot)$, $\mathcal{X}_F$, are characterized by the mapping $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}; \mathbf{q})$ where $\bar{\mathbf{x}} \in \bar{\mathcal{X}}_G$ (Please see the Example above).

Therefore, we are interested in the probability of the event that we have $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}, \mathbf{q}) = \mathbf{x}^r$ for any $\bar{\mathbf{x}} \in \bar{\mathcal{X}}_G$ and $\{\mathbf{x}^r\}_{r=0}^T$, which is same as the following probability:

$$\mathbb{P}\Big[\bigcup_{\bar{x}\in\bar{\mathcal{X}}_G, r\in[T]} \bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}, \mathbf{q}) = \mathbf{x}^r\Big].$$

To evaluate this probability we consider the event that for any given $\bar{\mathbf{x}} \in \bar{\mathcal{X}}_G$ and $r \in [T]$ we have $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}, \mathbf{q}) = \mathbf{x}^r$. The probability of this event can be evaluated as:

$$\mathbb{P}\left[\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}, \mathbf{q}) = \mathbf{x}^r\right] = \mathbb{P}[\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}, \mathbf{q}) - \mathbf{x}^r = 0] = 0.$$

The last equality follows from the fact that $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}, \mathbf{q})$ is a continuous random variable (see Lemma 3.1) while $\mathbf{x}^r$ is a mixed random variable[6]. Specifically, note that the CDF of the two random variables are different which implies that the random variable $\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}, \mathbf{q}) - \mathbf{x}^r$ is a composition of a continuous and a mixed random variable and the probability that it takes a fixed value is zero.

Now, using this we evaluate

$$\mathbb{P}\Big[\bigcup_{\bar{x}\in\bar{\mathcal{X}}, r\in[T]} \bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}, \mathbf{q}) = \mathbf{x}^r\Big] \leq \sum_{\bar{x}\in\bar{\mathcal{X}}} \sum_{r\in[T]} \mathbb{P}\Big[\bar{\boldsymbol{\epsilon}}(\bar{\mathbf{x}}, \mathbf{q}) = \mathbf{x}^r\Big] = 0,$$

where the first inequality follows from the union bound and the fact that the set $\bar{\mathcal{X}}$ is countable. Hence, the lemma is proved. □

[6] In this statement, we have made an implicit assumption that the functions $\bar{\epsilon}(\bar{\mathbf{x}}, \cdot)$ and $\mathbf{x}^r$ as a function of $\mathbf{q}$ do not share common support of non-zero measure.

D.2.2. Proof of Proposition 3.2

Proof. From Assumption 2.1 we know that $h(\mathbf{x},\mathbf{y})$ (and thus $g(\mathbf{x},\mathbf{y})$) is strongly convex in $\mathbf{y}$ with modulus $\mu_g = \mu_h$. As a result we have that

$$h(\mathbf{x},\mathbf{y}^*(\mathbf{x})) \geq h(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x})) + \langle \nabla_y h(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x})), \mathbf{y}^*(\mathbf{x}) - \overline{\mathbf{y}}^*(\mathbf{x})\rangle + \frac{\mu_g}{2}\|\mathbf{y}^*(\mathbf{x}) - \overline{\mathbf{y}}^*(\mathbf{x})\|^2 \tag{33}$$

$$g(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x})) \geq g(\mathbf{x},\mathbf{y}^*(\mathbf{x})) + \langle \nabla_y g(\mathbf{x},\mathbf{y}^*(\mathbf{x})), \overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\rangle + \frac{\mu_g}{2}\|\overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\|^2. \tag{34}$$

By definition $\overline{\mathbf{y}}^*(\mathbf{x})$ is the global minimum of the objective $h(\mathbf{x},\mathbf{y})$, and so it holds that $\langle \nabla_y h(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x})), \mathbf{y}^*(\mathbf{x}) - \overline{\mathbf{y}}^*(\mathbf{x})\rangle \geq 0$. Similarly, $\mathbf{y}^*(\mathbf{x})$ is the global minimum of the objective $g(\mathbf{x},\mathbf{y})$, and so it holds that $\langle \nabla_y g(\mathbf{x},\mathbf{y}^*(\mathbf{x})), \overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\rangle \geq 0$. Then, using the above inequalities and adding (33) and (34), we get

$$\begin{aligned}
&h(\mathbf{x},\mathbf{y}^*(\mathbf{x})) + g(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x})) \geq h(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x})) + g(\mathbf{x},\mathbf{y}^*(\mathbf{x})) + \mu_g\|\overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\|^2\\
&\mu_g\|\overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\|^2 \leq [h(\mathbf{x},\mathbf{y}^*(\mathbf{x})) - g(\mathbf{x},\mathbf{y}^*(\mathbf{x}))] + [g(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x})) - h(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x}))]\\
&\mu_g\|\overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\|^2 \leq -\mathbf{q}^T\mathbf{y}^*(\mathbf{x}) + \mathbf{q}^T\overline{\mathbf{y}}^*(\mathbf{x})\\
&\|\overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\|^2 \leq \frac{\mathbf{q}^T\left(\overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\right)}{\mu_g}\\
&\|\overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\|^2 \leq \frac{\|\mathbf{q}^T\|\|\overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\|}{\mu_g}\\
&\|\overline{\mathbf{y}}^*(\mathbf{x}) - \mathbf{y}^*(\mathbf{x})\| \leq \frac{\|\mathbf{q}\|}{\mu_g}.
\end{aligned}$$

Using the above bound and the fact that f is Lipschitz continuous (it follows from the bounded gradient assumption 2.7(a)) it is easy to see that

$$|F(\mathbf{x}) - G(\mathbf{x})| = |f(\mathbf{x},\mathbf{y}^*(\mathbf{x})) - f(\mathbf{x},\overline{\mathbf{y}}^*(\mathbf{x}))| \leq \overline{L}_f\|\mathbf{y}^*(\mathbf{x}) - \overline{\mathbf{y}}^*(\mathbf{x})\| \leq \overline{L}_f\frac{\|\mathbf{q}\|}{\mu_g}.$$

Therefore, the proof is complete. □

D.2.3. Proof of Theorem 3.3

Lemma D.10. *Under Assumption 2.1, 2.5, 2.7, $\nabla F(\mathbf{x})$ is almost surely continuous at a neighborhood around x, for any given $\mathbf{x} \in \mathcal{X}$.*

Proof. To begin with, we already established in Lemma 2.4 that F is almost surely differentiable at any given $\mathbf{x} \in \mathcal{X}$. Therefore, for any $\mathbf{x} \in \mathcal{X}$ there exists (almost surely) a neighborhood around it such that the matrix $\overline{A}$ corresponding to the active constraints at $\mathbf{y}^*(\mathbf{x})$ remains unchanged, where the gradient $\nabla\mathbf{y}^*(\mathbf{x})$ is defined in eq. (7), (8). Further, since $\overline{A}$ is locally (i.e., around any given $\mathbf{x}$) constant, and the formulas in (7), (8) can be seen as the results of a number of continuous operations over continuous functions, it is implied that $\nabla\mathbf{y}^*(\mathbf{x})$ is also a continuous function at a neighborhood around $\mathbf{x}$ almost surely. As a result, $\nabla F(\mathbf{x}) = \nabla_x f(\mathbf{x},\mathbf{y}^*(\mathbf{x})) + [\nabla\mathbf{y}^*(\mathbf{x})]^T\nabla_y f(\mathbf{x},\mathbf{y}^*(\mathbf{x}))$ is almost surely continuous locally around any given $\mathbf{x} \in \mathcal{X}$. □

Proof of Theorem 3.3. Here we follow a reasoning similar to the proof of Bertsekas (1998, Prop. 1.2.1). However, there are a number of differences that make this proof more challenging. First, in our setting we are optimizing an inexact version of the objective $\widehat{F}(\mathbf{x}) = f(\mathbf{x},\widehat{\mathbf{y}}(\mathbf{x}))$, using an approximate version of the gradient $\widehat{\nabla}F(\mathbf{x})$. Since the approximate gradient we are using is not the gradient of the objective $\widehat{F}(\mathbf{x})$ (the gradient of this function might not even exist), we consider a modification of the standard Armijo rule where an additional error term is present. Secondly, in the proof below the (classical) mean value theorem (Bertsekas, 1998, Prop. 1.23) cannot be applied, since we cannot ensure that F is (surely) differentiable at any given interval over $\mathbf{x}$. As we are going to show below, we use an alternative mean value theorem that does not require such assumption.

To begin with, we know that for the exact implicit objective $F(\mathbf{x})$ and gradient $\nabla F(\mathbf{x})$ (quantities to which we do not have access to) we can find at each iteration r a step-size a^r such that the following condition holds

$$F(\mathbf{x}^r) - F(\mathbf{x}^r + a^r \mathbf{d}^r) \geq -\sigma a^r \left[\nabla F(\mathbf{x}^r)\right]^T \mathbf{d}^r, \tag{35}$$

where $\mathbf{d}^r = \widetilde{\mathbf{x}}^r - \mathbf{x}^r$ with $\widetilde{\mathbf{x}}^r = \text{proj}_{\mathcal{X}}(\mathbf{x}^r - \nabla F(\mathbf{x}^r))$.

Next, the difference between the (approximate) objective values of two successive iterates (for simplicity we will use the notation $\mathbf{x}^{r+1} = \mathbf{x}^r + a^r \mathbf{d}^r$, $\widehat{\mathbf{x}}^{r+1} = \mathbf{x}^r + a^r \widehat{\mathbf{d}}^r$; $\widehat{\mathbf{d}}^r$ is defined in Algorithm 1) is

$$\begin{aligned}
\widehat{F}(\mathbf{x}^r) - \widehat{F}(\widehat{\mathbf{x}}^{r+1}) &= \widehat{F}(\mathbf{x}^r) - F(\mathbf{x}^r) + F(\mathbf{x}^r) - F(\mathbf{x}^{r+1}) + F(\mathbf{x}^{r+1}) - F(\widehat{\mathbf{x}}^{r+1}) + F(\widehat{\mathbf{x}}^{r+1}) - \widehat{F}(\widehat{\mathbf{x}}^{r+1}) \\
&= f(\mathbf{x}^r, \widehat{\mathbf{y}}(\mathbf{x}^r)) - f(\mathbf{x}^r, \mathbf{y}^*(\mathbf{x}^r)) + F(\mathbf{x}^r) - F(\mathbf{x}^{r+1}) + F(\mathbf{x}^{r+1}) - F(\widehat{\mathbf{x}}^{r+1}) \\
&\qquad + f(\widehat{\mathbf{x}}^{r+1}, \mathbf{y}^*(\widehat{\mathbf{x}}^{r+1})) - f(\widehat{\mathbf{x}}^{r+1}, \widehat{\mathbf{y}}(\widehat{\mathbf{x}}^{r+1})) \\
&\geq -L_f \|\mathbf{y}^*(\mathbf{x}^r) - \widehat{\mathbf{y}}(\mathbf{x}^r)\| + F(\mathbf{x}^r) - F(\mathbf{x}^{r+1}) - \overline{L}_F \|\mathbf{x}^{r+1} - \widehat{\mathbf{x}}^{r+1}\| \\
&\qquad - L_f \|\mathbf{y}^*(\mathbf{x}^{r+1}) - \widehat{\mathbf{y}}(\mathbf{x}^{r+1})\| \\
&\geq -L_f \delta^r - \sigma a^r \left[\nabla F(\mathbf{x}^r)\right]^T \mathbf{d}^r - \overline{L}_F a^r \|\mathbf{d}^r - \widehat{\mathbf{d}}^r\| - L_f \delta^{r+1} \\
&\geq -L_f \delta^r - \sigma a^r \left[\nabla F(\mathbf{x}^r)\right]^T \mathbf{d}^r - \overline{L}_F L_F a^r \delta^r - L_f \delta^{r+1} \\
&= -\sigma a^r \left[\nabla F(\mathbf{x}^r)\right]^T \mathbf{d}^r - \epsilon_1(\delta; r),
\end{aligned} \tag{36}$$

where we set $\epsilon_1(\delta; r) = L_f \delta^r + \overline{L}_F L_F a^r \delta^r + L_f \delta^{r+1}$. In the first inequality, we used the Lipschitz continuity of f and F; in the second inequality Assumption 2.5(a) and condition (35) were applied; in the third inequality the non-expansive property of the projection operator was used.

Also, we have that

$$\begin{aligned}
\nabla^T F(\mathbf{x}^r) \mathbf{d}^r &= \left(\nabla F(\mathbf{x}^r) - \widehat{\nabla} F(\mathbf{x}^r) + \widehat{\nabla} F(\mathbf{x}^r)\right)^T \left(\mathbf{d}^r + \widehat{\mathbf{d}}^r - \widehat{\mathbf{d}}^r\right) \\
&= \left(\nabla F(\mathbf{x}^r) - \widehat{\nabla} F(\mathbf{x}^r)\right)^T \left(\mathbf{d}^r - \widehat{\mathbf{d}}^r\right) + \left(\nabla F(\mathbf{x}^r) - \widehat{\nabla} F(\mathbf{x}^r)\right)^T \widehat{\mathbf{d}}^r \\
&+ \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \left(\mathbf{d}^r - \widehat{\mathbf{d}}^r\right) + \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r \\
&\leq \|\nabla F(\mathbf{x}^r) - \widehat{\nabla} F(\mathbf{x}^r)\| \|\mathbf{d}^r - \widehat{\mathbf{d}}^r\| + \|\nabla F(\mathbf{x}^r) - \widehat{\nabla} F(\mathbf{x}^r)\| \|\widehat{\mathbf{d}}^r\| \\
&+ \|\widehat{\nabla} F(\mathbf{x}^r)\| \|\mathbf{d}^r - \widehat{\mathbf{d}}^r\| + \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r \\
&\leq L_F^2 \left(\delta^r\right)^2 + L_F \overline{L}_F \delta^r + L_F \overline{L}_F \delta^r + \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r \\
&= \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r + \epsilon_2(\delta; r),
\end{aligned} \tag{37}$$

where $\epsilon_2(\delta; r) = L_F^2 \left(\delta^r\right)^2 + 2 L_F \overline{L}_F \delta^r$. Notice that the results in the second inequality follow from Lemma 2.8.

Then, combining (36) and (37) we get

$$\begin{aligned}
\widehat{F}(\mathbf{x}^r) - \widehat{F}(\widehat{\mathbf{x}}^{r+1}) &\geq -\sigma a^r \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r - \epsilon_1(\delta; r) - \sigma a^r \epsilon_2(\delta; r) \\
&= -\sigma a^r \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r - \epsilon(\delta; r),
\end{aligned}$$

where $\epsilon(\delta; r) = \epsilon_1(\delta; r) + \sigma a^r \epsilon_2(\delta; r)$; notice that $\lim_{\delta \to 0} \epsilon(\delta) = 0$. In conclusion, we can follow this (inexact) Armijo-type rule in our inexact problem; the existence of the (Armijo) step-size is guaranteed by its existence for the exact problem (35).

Now let us move to the main part of the proof, which follows the reasoning used in Bertsekas (1998, Prop. 1.2.1). Let $\{\mathbf{x}^r\} \in \mathcal{X}$ be the iterate sequence of our algorithm, and let $\bar{\mathbf{x}} \in \mathcal{X}$ be a limit point; the existence of such point is guaranteed by the closedness of the set $\mathcal{X}$. Moreover, it is established in Proposition 2.2 that $F(\mathbf{x})$ is continuous, and as a result it

holds that $\lim_{r\to+\infty} F(\mathbf{x}^r) = F(\bar{\mathbf{x}})$. The latter results combined with the fact that all convergent sequences are also Cauchy sequences, implies that $\lim_{r\to+\infty}(F(\mathbf{x}^r) - F(\mathbf{x}^{r+1})) = 0$.

We want to show that $\bar{\mathbf{x}}$ is a stationary point of $F(\mathbf{x})$. We are going to show that by assuming that the opposite holds, i.e., $\bar{\mathbf{x}}$ is not a stationary point of $F(\mathbf{x})$, and arriving at a contradiction. From the Armijo rule of our problem we have that

$$\widehat{F}(\mathbf{x}^r) - \widehat{F}(\widehat{\mathbf{x}}^{r+1}) \geq -\sigma a^r \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r - \epsilon(\delta; r). \tag{38}$$

Then, consider the following

$$\begin{aligned}
\widehat{F}(\mathbf{x}^r) - \widehat{F}(\widehat{\mathbf{x}}^{r+1}) &= \widehat{F}(\mathbf{x}^r) - F(\mathbf{x}^r) + F(\mathbf{x}^r) - F(\mathbf{x}^{r+1}) + F(\mathbf{x}^{r+1}) - F(\widehat{\mathbf{x}}^{r+1}) + F(\widehat{\mathbf{x}}^{r+1}) - \widehat{F}(\widehat{\mathbf{x}}^{r+1}) \\
&= f(\mathbf{x}^r, \widehat{\mathbf{y}}(\mathbf{x}^r)) - f(\mathbf{x}^r, \mathbf{y}^*(\mathbf{x}^r)) + F(\mathbf{x}^r) - F(\mathbf{x}^{r+1}) + F(\mathbf{x}^{r+1}) - F(\widehat{\mathbf{x}}^{r+1}) \\
&+ f(\widehat{\mathbf{x}}^{r+1}, \mathbf{y}^*(\widehat{\mathbf{x}}^{r+1})) - f(\widehat{\mathbf{x}}^{r+1}, \widehat{\mathbf{y}}(\widehat{\mathbf{x}}^{r+1})) \\
&\leq L_f \|\mathbf{y}^*(\mathbf{x}^r) - \widehat{\mathbf{y}}(\mathbf{x}^r)\| + F(\mathbf{x}^r) - F(\mathbf{x}^{r+1}) + \overline{L}_F \|\mathbf{x}^{r+1} - \widehat{\mathbf{x}}^{r+1}\| \\
&\qquad + L_f \|\mathbf{y}^*(\mathbf{x}^{r+1}) - \widehat{\mathbf{y}}(\mathbf{x}^{r+1})\| \\
&\leq L_f \delta^r + F(\mathbf{x}^r) - F(\mathbf{x}^{r+1}) + \overline{L}_F a^r \|\mathbf{d}^r - \widehat{\mathbf{d}}^r\| + L_f \delta^{r+1} \\
&\leq F(\mathbf{x}^r) - F(\mathbf{x}^{r+1}) + L_f \delta^r + \overline{L}_F L_F a^r \delta^r + L_f \delta^{r+1} \\
&= F(\mathbf{x}^r) - F(\mathbf{x}^{r+1}) + \epsilon_1(\delta; r).
\end{aligned}$$

In the first inequality, we used the Lipschitz continuity of f and F; in the second inequality Assumption 2.5(a) and condition (35) were applied; in the third inequality the non-expansive property of the projection operator was used. Using the above derivation we can bound the left-hand side of inequality (38) as follows

$$F(\mathbf{x}^r) - F(\mathbf{x}^{r+1}) + \epsilon_1(\delta; r) \geq \widehat{F}(\mathbf{x}^r) - \widehat{F}(\widehat{\mathbf{x}}^{r+1}) \geq -\sigma a^r \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r - \epsilon(\delta; r).$$

It is easy to see that the left-hand side in the above inequality tends to 0. Therefore, $\lim_{r\to+\infty} \left(-\sigma a^r \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r - \epsilon(\delta; r)\right) \leq 0$. In addition, we know that $\lim_{r\to+\infty} \epsilon(\delta; r) = 0$ and $-\sigma a^r \left[\widehat{\nabla} F(\mathbf{x})\right]^T \widehat{\mathbf{d}}^r \geq 0, \forall \mathbf{x} \in \mathcal{X}$. From the above statements we can conclude that

$$\lim_{r\to+\infty} \sigma a^r \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r = 0. \tag{39}$$

Moreover, from the gradient-related assumption we know that for a non-stationary point $\bar{\mathbf{x}}$ we have that

$$\limsup_{r\to\infty, r\in\mathcal{R}} \left[\widehat{\nabla} F(\mathbf{x}^r)\right]^T \widehat{\mathbf{d}}^r < 0, \tag{40}$$

where$\{\mathbf{x}^r\}_{\mathcal{R}}$ is subsequence with $\lim\limits_{r\to\infty, r\in\mathcal{R}} \mathbf{x}^r = \bar{\mathbf{x}}$. Then, the conditions (39), (40) imply that

$$\lim_{r\to\infty, r\in\mathcal{R}} a^r = 0.$$

In the subsequence $\mathcal{R}$ we can find an index $\bar{r} \geq 0$ such that

$$\widehat{F}(\mathbf{x}^r) - \widehat{F}\left(\mathbf{x}^r + \left(\frac{a^r}{\beta}\right)\widehat{\mathbf{d}}^r\right) < -\sigma\left(\frac{a^r}{\beta}\right)\widehat{\nabla}^T F(\mathbf{x}^r)\widehat{\mathbf{d}}^r - \epsilon(\delta; r), \forall r \in \mathcal{R}, r \geq \bar{r}. \tag{41}$$

Similarly with the proof of Bertsekas (1998, Prop. 1.2.1) let us introduce the following sequences:

$$\widehat{\mathbf{p}}^r = \frac{\widehat{\mathbf{d}}^r}{\|\widehat{\mathbf{d}}^r\|}, \bar{a}^r = \frac{a^r \|\widehat{\mathbf{d}}^r\|}{\beta}$$

The first sequence $\{\widehat{\mathbf{p}}^r\}$ is bounded and so it admits a limit point $\bar{\mathbf{p}}$ with $\|\bar{\mathbf{p}}\| = 1$, that is $\lim_{r\to+\infty, r\in\overline{\mathcal{R}}} \mathbf{p}^r = \bar{\mathbf{p}}$, where $\overline{\mathcal{R}}$ denotes the indices of a subsequence of $\mathcal{R}$. In addition, taking into account the facts that $\lim_{r\to+\infty, r\in\mathcal{R}} a^r = 0$ and the fact that the sequence $\{\|\mathbf{d}^r\|\}_{\mathcal{R}}$ is bounded we can easily see that $\lim_{r\to+\infty, r\in\mathcal{R}} \bar{a}^r = 0$.

Dividing both sides of (41) by $\bar{a}^r$ and using the definitions of $\widehat{\mathbf{p}}^r$ and $\bar{a}_r$ from above we get

$$\frac{\widehat{F}(\mathbf{x}^r) - \widehat{F}\left(\mathbf{x}^r + \bar{a}^r\widehat{\mathbf{p}}^r\right)}{\bar{a}^r} < -\sigma\left[\widehat{\nabla}F(\mathbf{x}^r)\right]^T\widehat{\mathbf{p}}^r - \epsilon(\delta; r), \forall r \in \overline{\mathcal{R}}, r > \bar{r}. \tag{42}$$

Then, we have that (for convenience we adopt the notation $\widehat{\mathbf{x}}^{r+1} = \mathbf{x}^r + \bar{a}^r\widehat{\mathbf{p}}^r$)

$$\begin{aligned}
\widehat{F}(\mathbf{x}^r) - \widehat{F}(\widehat{\mathbf{x}}^{r+1}) &= \widehat{F}(\mathbf{x}^r) - F(\mathbf{x}^r) + F(\mathbf{x}^r) - F(\widehat{\mathbf{x}}^{r+1}) + F(\widehat{\mathbf{x}}^{r+1}) - \widehat{F}(\widehat{\mathbf{x}}^{r+1}) \\
&= f(\mathbf{x}^r, \widehat{\mathbf{y}}(\mathbf{x}^r)) - f(\mathbf{x}^r, \mathbf{y}^*(\mathbf{x}^r)) + F(\mathbf{x}^r) - F(\widehat{\mathbf{x}}^{r+1}) \\
&\qquad + f(\widehat{\mathbf{x}}^{r+1}, \mathbf{y}^*(\widehat{\mathbf{x}}^{r+1})) - f(\widehat{\mathbf{x}}^{r+1}, \widehat{\mathbf{y}}(\widehat{\mathbf{x}}^{r+1})) \\
&\geq -L_f\|\mathbf{y}^*(\mathbf{x}^r) - \widehat{\mathbf{y}}(\mathbf{x}^r)\| - L_f\|\mathbf{y}^*(\widehat{\mathbf{x}}^{r+1}) - \widehat{\mathbf{y}}(\widehat{\mathbf{x}}^{r+1})\| + F(\mathbf{x}^r) - F(\widehat{\mathbf{x}}^{r+1}) \\
&\geq F(\mathbf{x}^r) - F(\widehat{\mathbf{x}}^{r+1}) - L_f\delta^r - L_f\delta^{r+1},
\end{aligned} \tag{43}$$

where the first inequality above follows the Lipschitz continuity of f, and the second inequality is an application of Assumption 2.5(a). Incorporating inequality (43) into (42) results to

$$\frac{F(\mathbf{x}^r) - F(\mathbf{x}^r + \bar{a}^r\widehat{\mathbf{p}}^r)}{\bar{a}_r} - L_f\frac{\delta^r + \delta^{r+1}}{\bar{a}_r} < -\sigma\left[\widehat{\nabla}F(\mathbf{x}^r)\right]^T\widehat{\mathbf{p}}^r - \epsilon(\delta; r), \forall r \in \overline{\mathcal{R}}, r > \bar{r}. \tag{44}$$

Lebourg's mean value theorem (Lebourg, 1979, Theorem 1.7) implies that

$$\frac{F(\mathbf{x}^r) - F(\mathbf{x}^r + \bar{a}^r\widehat{\mathbf{p}}^r)}{\bar{a}_r} = \mathbf{u}^{\mathbf{T}}\widehat{\mathbf{p}}^{\mathbf{r}}$$

with $\mathbf{u} \in \vartheta\mathbf{F}(\mathbf{x}^{\mathbf{r}} + \widetilde{\mathbf{a}}^{\mathbf{r}}\widehat{\mathbf{p}}^{\mathbf{r}})$ and $\widetilde{a}^r \in [0, \bar{a}^r]$, where $\vartheta F(\cdot)$ is the Clarke subdifferntial of F. We know that F is almost surely continuously differentiable (Lemma D.10) at any $\mathbf{x}^r + \widetilde{a}^r\widehat{\mathbf{p}}^r \in \mathcal{X}$, and so the Clarke subdifferential at $\mathbf{x}^r + \bar{a}^r\widehat{\mathbf{p}}^r$ becomes w.p. 1 equal to $\nabla F(\mathbf{x}^r + \widetilde{a}^r\widehat{\mathbf{p}}^r)$. Note that the we cannot use here the (classical) mean value theorem (Bertsekas, 1998, Prop. 1.23), as in the proof of Bertsekas (1998, Prop. 1.2.1), because it requires that the function $F(\mathbf{x})$ is (surely) differentiable on the interval $[\mathbf{x}^r, \mathbf{x}^r + \bar{a}^r\widehat{\mathbf{p}}^r]$.

Then, we can rewrite the expression in (44) as follows

$$-L_f\beta\frac{\delta^r + \delta^{r+1}}{a_r\|\widehat{\mathbf{d}}^r\|} - \left[\nabla F(\mathbf{x}^r + \widetilde{a}^r\widehat{\mathbf{p}}^r)\right]^T\widehat{\mathbf{p}}^r < -\sigma\left[\widehat{\nabla}F(\mathbf{x}^r)\right]^T\widehat{\mathbf{p}}^r - \epsilon(\delta; r), \forall r \in \overline{\mathcal{R}}, r > \bar{r},$$

where $\widetilde{a}^r \in [0, \bar{a}^r]$.

Using the assumption that $0 \leq \frac{\delta^r}{a^r} \sim \mathcal{O}(c^r)$, where c^r is some sequence with $\lim_{r\to\infty, r\in\overline{\mathcal{R}}} c^r = 0$, and the fact that $\lim_{r\to\infty, r\in\overline{\mathcal{R}}} \|\widehat{\mathbf{d}}^r\| \neq 0$ (because of the assumption that the sequence $\mathbf{x}^r$ converges to a non-stationary point), we compute the limit in the above expression and get

$$\begin{aligned}
-\left[\nabla F(\bar{\mathbf{x}})\right]^T\bar{\mathbf{p}} &< -\sigma\left[\nabla F(\bar{\mathbf{x}})\right]^T\bar{\mathbf{p}} \\
0 &< (1-\sigma)\left[\nabla F(\bar{\mathbf{x}})\right]^T\bar{\mathbf{p}} \\
0 &< \left[\nabla F(\bar{\mathbf{x}})\right]^T\bar{\mathbf{p}}.
\end{aligned} \tag{45}$$

However, note that $\left[\widehat{\nabla}F(\mathbf{x}^r)\right]^T\widehat{\mathbf{p}}^r = \frac{\widehat{\nabla}^T F(\mathbf{x}^r)\widehat{\mathbf{d}}^r}{\|\widehat{\mathbf{d}}^r\|}$ and therefore if we take limits in both sides we obtain

$$\left[\nabla F(\bar{\mathbf{x}})\right]^T\bar{\mathbf{p}}^r \leq \frac{\limsup_{r\to\infty, r\in\overline{\mathcal{R}}} \widehat{\nabla}^T F(\mathbf{x}^r)\widehat{\mathbf{d}}^r}{\limsup_{r\to\infty, r\in\overline{\mathcal{R}}} \|\widehat{\mathbf{d}}^r\|} < 0, \tag{46}$$

due to the gradient-related assumption. We notice that expressions (45) and (46) lead to a contradiction. Therefore, $\bar{\mathbf{x}}$ is a stationary point of $F(\mathbf{x})$.

The proof is now complete. □

D.2.4. WEAKLY-CONVEX OBJECTIVE: PROOF OF THEOREM 3.6

Proof. Define $\hat{\mathbf{x}}^r = \arg\min_{\mathbf{z}\in\mathbb{R}^{d_u}} \Big\{H(\mathbf{z}) + \frac{\hat{\rho}}{2}\|\mathbf{x}^r - \mathbf{z}\|^2\Big\}$. Using the definition of Moreau envelope, we have

$$
\begin{aligned}
\mathbb{E}[H_{1/\hat{\rho}}(\mathbf{x}^{r+1})] &\leq \mathbb{E}\Big[F(\hat{\mathbf{x}}^r) + \frac{\hat{\rho}}{2}\|\mathbf{x}^{r+1} - \hat{\mathbf{x}}^r\|^2\Big] \\
&\overset{(a)}{=} \mathbb{E}\Big[F(\hat{\mathbf{x}}^r) + \frac{\hat{\rho}}{2}\|\text{proj}_{\mathcal{X}}(\mathbf{x}^r - \beta\widehat{\nabla}F(\mathbf{x}^r;\xi^r)) - \text{proj}_{\mathcal{X}}(\hat{\mathbf{x}}^r)\|^2\Big] \\
&\overset{(b)}{\leq} \mathbb{E}\Big[F(\hat{\mathbf{x}}^r) + \frac{\hat{\rho}}{2}\|\mathbf{x}^r - \beta\widehat{\nabla}F(\mathbf{x}^r;\xi^r) - \hat{\mathbf{x}}^r\|^2\Big] \\
&= F(\hat{\mathbf{x}}^r) + \frac{\hat{\rho}}{2}\mathbb{E}\Big[\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 - 2\langle \mathbf{x}^r - \hat{\mathbf{x}}^r, \beta\widehat{\nabla}F(\mathbf{x}^r;\xi^r)\rangle + \beta^2\|\widehat{\nabla}F(\mathbf{x}^r;\xi^r)\|^2\Big] \\
&\overset{(c)}{\leq} F(\hat{\mathbf{x}}^r) + \frac{\hat{\rho}}{2}\mathbb{E}\Big[\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 - 2\langle \mathbf{x}^r - \hat{\mathbf{x}}^r, \beta\widehat{\nabla}F(\mathbf{x}^r;\xi^r)\rangle \\
&\qquad\qquad + 2\beta^2\|\widehat{\nabla}F(\mathbf{x}^r;\xi^r) - \widehat{\nabla}F(\mathbf{x}^r)\|^2 + 2\beta^2\|\widehat{\nabla}F(\mathbf{x}^r)\|^2\Big] \\
&\overset{(d)}{\leq} F(\hat{\mathbf{x}}^r) + \frac{\hat{\rho}}{2}\Big[\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 - 2\beta\langle \mathbf{x}^r - \hat{\mathbf{x}}^r, \widehat{\nabla}F(\mathbf{x}^r)\rangle + 2\beta^2\big(\sigma_F^2 + \overline{L}_F^2\big)\Big] \\
&\leq F(\hat{\mathbf{x}}^r) + \frac{\hat{\rho}}{2}\Big[\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 + 2\beta\underbrace{\langle \hat{\mathbf{x}}^r - \mathbf{x}^r, \nabla F(\mathbf{x}^r)\rangle}_{\text{Term I}} \\
&\qquad\qquad + 2\beta\underbrace{\langle \hat{\mathbf{x}}^r - \mathbf{x}^r, \widehat{\nabla}F(\mathbf{x}^r) - \nabla F(\mathbf{x}^r)\rangle}_{\text{Term II}} + 2\beta^2\big(\sigma_F^2 + \overline{L}_F^2\big)\Big],
\end{aligned} \tag{47}
$$

where (a) follows from the fact that $\mathbf{x}^{r+1} \in \mathcal{X}$ and $\hat{\mathbf{x}}^r \in \mathcal{X}$; (b) results from the non-expansiveness of the projection operator; (c) uses $\|\mathbf{a} - \mathbf{b}\|^2 = 2\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2$; and (d) results from the application of Lemmas 2.8 and 2.10.

Next, considering Term I and Term II separately in (47) above. For Term I, we get using the weak convexity of $F(\cdot)$

$$
\text{Term I} = \langle \hat{\mathbf{x}}^r - \mathbf{x}^r, \nabla F(\mathbf{x}^r)\rangle \leq F(\hat{\mathbf{x}}^r) - F(\mathbf{x}^r) + \frac{\rho}{2}\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2
$$

We bound Term II using the Young's inequality as

$$
\begin{aligned}
\text{Term II} &= \langle \hat{\mathbf{x}}^r - \mathbf{x}^r, \widehat{\nabla}F(\mathbf{x}^r) - \nabla F(\mathbf{x}^r)\rangle \\
&\leq \frac{\rho}{2}\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 + \frac{1}{2\rho}\|\widehat{\nabla}F(\mathbf{x}^r) - \nabla F(\mathbf{x}^r)\|^2 \\
&\overset{(e)}{\leq} \frac{\rho}{2}\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 + \frac{1}{2\rho}L_F^2\delta^2
\end{aligned}
$$

where (e) follows from Lemma 2.8. Next, substituting the bounds of Term I and Term II in (47) and using the definition of $\hat{\mathbf{x}}^r$, we get

$$
\begin{aligned}
\mathbb{E}[H_{1/\hat{\rho}}(\mathbf{x}^{r+1})] &\leq F(\hat{\mathbf{x}}^r) + \frac{\hat{\rho}}{2}\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 + \hat{\rho}\beta\Big[F(\hat{\mathbf{x}}^r) - F(\mathbf{x}^r) + \rho\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2\Big] \\
&\qquad + \frac{\hat{\rho}\beta}{2\rho}L_F^2\delta^2 + \beta^2\hat{\rho}\big(\sigma_F^2 + \overline{L}_F^2\big) \\
&\leq H_{1/\hat{\rho}}(\mathbf{x}^r) + \hat{\rho}\beta\underbrace{\Big[F(\hat{\mathbf{x}}^r) - F(\mathbf{x}^r) + \rho\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2\Big]}_{\text{Term III}} + \frac{\hat{\rho}\beta}{2\rho}L_F^2\delta^2 + \beta^2\hat{\rho}\big(\sigma_F^2 + \overline{L}_F^2\big).
\end{aligned} \tag{48}
$$

Next, we bound Term III in (48) above.

$$
\begin{aligned}
\text{Term III} &= F(\hat{\mathbf{x}}^r) - F(\mathbf{x}^r) + \rho\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 \\
&= F(\hat{\mathbf{x}}^r) + \frac{\hat{\rho}}{2}\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 - F(\mathbf{x}^r) + \frac{2\rho - \hat{\rho}}{2}\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 \\
&\leq \frac{3\rho - 2\hat{\rho}}{2}\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 \leq \frac{3\rho - 2\hat{\rho}}{2\hat{\rho}^2}\|\nabla H_{1/\hat{\rho}}(\mathbf{x}^r)\|^2,
\end{aligned}
$$

where the last equality follows from (14) and the first inequality follows from the fact that $F(\mathbf{x}) + \frac{\hat{\rho}}{2}\|\mathbf{x}^r - \mathbf{x}\|^2$ is $(\hat{\rho} - \rho)$-strongly convex. This implies the following

$$\begin{aligned} F(\hat{\mathbf{x}}^r) + \frac{\hat{\rho}}{2}\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 - F(\mathbf{x}^r) &\leq -\langle \nabla F(\hat{\mathbf{x}}^r) + \hat{\rho}(\mathbf{x}^r - \hat{\mathbf{x}}^r), x^r - \hat{x}^r\rangle - \frac{\hat{\rho} - \rho}{2}\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2 \\ &\leq \frac{\rho - \hat{\rho}}{2}\|\mathbf{x}^r - \hat{\mathbf{x}}^r\|^2, \end{aligned}$$

where the second inequality results from the definition of $\hat{\mathbf{x}}^r$.

Finally, substituting Term III in (48) and rearranging the terms we get:

$$\beta\left[\frac{2\hat{\rho} - 3\rho}{2\hat{\rho}}\right]\|\nabla H_{1/\hat{\rho}}(\mathbf{x}^r)\|^2 \leq \mathbb{E}\big[H_{1/\hat{\rho}}(\mathbf{x}^r) - H_{1/\hat{\rho}}(\mathbf{x}^{r+1})\big] + \beta^2\hat{\rho}\big(\sigma_F^2 + \overline{L}_F^2\big) + \frac{\hat{\rho}\beta}{2\rho}L_F^2\delta^2.$$

Summing over all $r \in \{0, 1, \ldots, T-1\}$ and dividing by T, we get

$$\frac{1}{T}\sum_{r=0}^{T-1}\|\nabla H_{1/\hat{\rho}}(\mathbf{x}^r)\|^2 \leq \left[\frac{2\hat{\rho}}{2\hat{\rho} - 3\rho}\right]\left[\frac{H_{1/\hat{\rho}}(\mathbf{x}^0) - H^*}{\beta T} + \beta\hat{\rho}\big(\sigma_F^2 + \overline{L}_F^2\big) + \frac{\hat{\rho}}{2\rho}L_F^2\delta^2\right].$$

Therefore, we have the result. □

D.2.5. Strongly-Convex Objective: Proof of Theorem 3.8

Proof. Using the update rule of the Algorithm 2, we have

$$\begin{aligned} \mathbb{E}\|\mathbf{x}^{r+1} - \mathbf{x}^*\|^2 &= \mathbb{E}\|\text{proj}_X(\mathbf{x}^r - \beta^r\widehat{\nabla}F(\mathbf{x}^r;\xi^r)) - \mathbf{x}^*\|^2 \\ &= \mathbb{E}\|\text{proj}_X(\mathbf{x}^r - \beta^r\widehat{\nabla}F(\mathbf{x}^r;\xi^r)) - \text{proj}_X(\mathbf{x}^*)\|^2 \\ &\overset{(a)}{\leq} \mathbb{E}\|\mathbf{x}^r - \beta^r\widehat{\nabla}F(\mathbf{x}^r;\xi^r) - \mathbf{x}^*\|^2 \\ &= \mathbb{E}\Big[\|\mathbf{x}^r - \mathbf{x}^*\|^2 + (\beta^r)^2\|\widehat{\nabla}F(\mathbf{x}^r;\xi^r)\|^2 - 2\beta^r\langle \mathbf{x}^r - \mathbf{x}^*, \widehat{\nabla}F(\mathbf{x}^r;\xi^r)\rangle\Big] \\ &\overset{(b)}{\leq} \mathbb{E}\Big[\|\mathbf{x}^r - \mathbf{x}^*\|^2 + 2(\beta^r)^2\|\widehat{\nabla}F(\mathbf{x}^r;\xi^r) - \widehat{\nabla}F(\mathbf{x}^r)\|^2 \\ &\qquad + 2(\beta^r)^2\|\widehat{\nabla}F(\mathbf{x}^r)\|^2 - 2\beta^r\langle \mathbf{x}^r - \mathbf{x}^*, \widehat{\nabla}F(\mathbf{x}^r)\rangle\Big] \\ &\overset{(c)}{\leq} \mathbb{E}\Big[\|\mathbf{x}^r - \mathbf{x}^*\|^2 + 2(\beta^r)^2(\sigma_F^2 + B_F^2) - 2\beta^r\langle \mathbf{x}^r - \mathbf{x}^*, \widehat{\nabla}F(\mathbf{x}^r)\rangle\Big] \\ &\leq \mathbb{E}\Big[\|\mathbf{x}^r - \mathbf{x}^*\|^2 + 2(\beta^r)^2(\sigma_F^2 + B_F^2) - 2\beta^r\langle \mathbf{x}^r - \mathbf{x}^*, \nabla F(\mathbf{x}^r)\rangle \\ &\qquad - 2\beta^r\langle \mathbf{x}^r - \mathbf{x}^*, \widehat{\nabla}F(\mathbf{x}^r) - \nabla F(\mathbf{x}^r)\rangle\Big] \\ &\overset{(d)}{\leq} \mathbb{E}\Big[\|\mathbf{x}^r - \mathbf{x}^*\|^2 + 2(\beta^r)^2(\sigma_F^2 + B_F^2) - 2\beta^r[F(\mathbf{x}^r) - F^*] - \mu_F\beta^r\|\mathbf{x}^r - \mathbf{x}^*\|^2 \\ &\qquad + 2\beta^r\|\mathbf{x}^r - \mathbf{x}^*\|\|\widehat{\nabla}F(\mathbf{x}^r) - \nabla F(\mathbf{x}^r)\|\Big] \\ &\overset{(e)}{\leq} \mathbb{E}\Big[(1 - \mu_F\beta^r)\|\mathbf{x}^r - \mathbf{x}^*\|^2 + 2(\beta^r)^2(\sigma_F^2 + B_F^2) - 2\beta^r[F(\mathbf{x}^r) - F^*] + 2\beta^r D_{\mathcal{X}}L_F\delta\Big] \end{aligned}$$

where in (a) the non-expansive property of the projection operator is applied, in (b) we add and subtract the term $\widehat{\nabla}F(\mathbf{x}^r)$, use the well-known inequality $\|\mathbf{a} + \mathbf{b}\|^2 \leq 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$, and in the last term we utilize Lemma 2.10; (c) results from the application of Lemmas 2.8 and 2.10; (d) uses strong-convexity (Assumption 3.7) of $F(\cdot)$ and Cauchy-Schwartz inequality; finally, (e) results from the application of Lemma 2.8 and Assumption 2.1(b).

Rearranging the terms, we get

$$2\beta^r \mathbb{E}[F(\mathbf{x}^r) - F^*] \leq \mathbb{E}\Big[(1-\mu_F\beta^r)\|\mathbf{x}^r - \mathbf{x}^*\|^2 - \|\mathbf{x}^{r+1} - \mathbf{x}^*\|^2 + 2(\beta^r)^2(\sigma_F^2 + B_F^2) + 2\beta^r D_{\mathcal{X}} L_F \delta\Big]$$
$$\mathbb{E}[F(\mathbf{x}^r) - F^*] \leq \mathbb{E}\Big[\Big(\frac{1}{2\beta^r} - \frac{\mu_F}{2}\Big)\|\mathbf{x}^r - \mathbf{x}^*\|^2 - \frac{1}{2\beta^r}\|\mathbf{x}^{r+1} - \mathbf{x}^*\|^2 + \beta^r(\sigma_F^2 + B_F^2) + D_{\mathcal{X}} L_F \delta\Big].$$

Summing over $r \in \{0, \ldots, T-1\}$, multiplying by $1/T$ and using the fact that $\beta^r = \frac{1}{\mu_F(r+1)}$, we get

$$\frac{1}{T}\sum_{r=0}^{T-1} \mathbb{E}[F(\mathbf{x}^r) - F^*] \leq \frac{\mu_F}{2T}\sum_{r=0}^{T-1} \mathbb{E}\Big[r\|\mathbf{x}^r - \mathbf{x}^*\|^2 - (r+1)\|\mathbf{x}^{r+1} - \mathbf{x}^*\|^2\Big] + \frac{1}{T}\sum_{r=0}^{T-1} \beta^r(\sigma_F^2 + B_F^2) + D_{\mathcal{X}} L_F \delta.$$

Telescoping the sum we get the following

$$\frac{1}{T}\sum_{r=0}^{T-1} \mathbb{E}[F(\mathbf{x}^r) - F^*] \leq -\frac{\mu_F}{2}\|\mathbf{x}^T - \mathbf{x}^*\|^2 + \frac{(\sigma_F^2 + B_F^2)}{\mu_F}\frac{\log(T)}{T} + D_{\mathcal{X}} L_F \delta$$
$$\leq \frac{(\sigma_F^2 + B_F^2)}{\mu_F}\frac{\log(T)}{T} + D_{\mathcal{X}} L_F \delta.$$

Therefore, we have the proof. □

D.2.6. Convex Objective: Proof of Theorem 3.9

Proof. From the update rule of Algorithm 2, we have for $\beta^r = \beta$ for all $r \in \{0, 1, \ldots, T-1\}$

$$\begin{aligned}
\mathbb{E}[\|\mathbf{x}^{r+1} - \mathbf{x}^*\|^2] &= \mathbb{E}[\|\text{proj}_{\mathcal{X}}(\mathbf{x}^r - \beta\widehat{\nabla} F(\mathbf{x}^r;\xi^r)) - \mathbf{x}^*\|^2] \\
&\overset{(a)}{=} \mathbb{E}[\|\text{proj}_{\mathcal{X}}(\mathbf{x}^r - \beta\widehat{\nabla} F(\mathbf{x}^r);\xi^r) - \text{proj}_{\mathcal{X}}(\mathbf{x}^*)\|^2] \\
&\overset{(b)}{\leq} \mathbb{E}[\|\mathbf{x}^r - \beta\widehat{\nabla} F(\mathbf{x}^r;\xi^r) - \mathbf{x}^*\|^2] \\
&\overset{(c)}{=} \mathbb{E}[\|\mathbf{x}^r - \mathbf{x}^*\|^2 + \beta^2\|\widehat{\nabla} F(\mathbf{x}^r;\xi^r)\|^2 - 2\beta\langle \mathbf{x}^r - \mathbf{x}^*, \widehat{\nabla} F(\mathbf{x}^r;\xi^r)\rangle] \\
&\overset{(d)}{\leq} \mathbb{E}[\|\mathbf{x}^r - \mathbf{x}^*\|^2 + 2\beta^2\|\widehat{\nabla} F(\mathbf{x}^r;\xi^r) - \widehat{\nabla} F(\mathbf{x}^r)\|^2 + 2\beta^2\|\widehat{\nabla} F(\mathbf{x}^r)\|^2 \\
&\qquad - 2\beta\langle \mathbf{x}^r - \mathbf{x}^*, \nabla F(\mathbf{x}^r)\rangle - 2\beta\langle \mathbf{x}^r - \mathbf{x}^*, \widehat{\nabla} F(\mathbf{x}^r) - \nabla F(\mathbf{x}^r)\rangle] \\
&\overset{(e)}{\leq} \mathbb{E}[\|\mathbf{x}^r - \mathbf{x}^*\|^2 + 2\beta^2\sigma_F^2 + 2\beta^2\overline{L}_F^2 - 2\beta(F(\mathbf{x}^r) - F^*) \\
&\qquad + 2\beta\|\mathbf{x}^r - \mathbf{x}^*\|\,\|\widehat{\nabla} F(\mathbf{x}^r) - \nabla F(\mathbf{x}^r)\|] \\
&\overset{(f)}{\leq} \mathbb{E}[\|\mathbf{x}^r - \mathbf{x}^*\|^2 + 2\beta^2(\sigma_F^2 + \overline{L}_F^2) - 2\beta(F(\mathbf{x}^r) - F^*) + 2\beta D_{\mathcal{X}} L_F \delta]
\end{aligned}$$

where (a) follows from the fact that $\mathbf{x}^* \in \mathcal{X}$; (b) results from the non-expansiveness of the projection operator; (c) uses $\|\mathbf{a} - \mathbf{b}\|^2 = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - 2\langle \mathbf{a}, \mathbf{b}\rangle$; (d) utilizes the fact that $\|\mathbf{a} - \mathbf{b}\|^2 = 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$ and unbiased gradient from Lemma 2.10; (e) results from Lemmas 2.8 and 2.10, the convexity assumption of the implicit function (Assumption 3.7 with $\mu_F = 0$) and the Cauchy-Schwartz inequality; and (f) results from Assumption 2.1(b) and Lemma 2.8.

Summing over $r \in \{0, 1 \ldots, T-1\}$, multiplying by $1/T$ and rearranging the terms, we get

$$\frac{1}{T}\sum_{r=0}^{T-1} \mathbb{E}[F(\mathbf{x}^r) - F^*] \leq \frac{\|\mathbf{x}^1 - \mathbf{x}^*\|^2}{2\beta T} + \beta(\sigma_F^2 + \overline{L}_F^2) + D_{\mathcal{X}} L_F \delta.$$

Using Jensen's inequality and denoting $\underline{\mathbf{x}} = \frac{1}{T}\sum_{r=0}^{T-1}\mathbf{x}^r$, we get

$$\mathbb{E}[F(\underline{\mathbf{x}}) - F^*] \leq \frac{\|\mathbf{x}^1 - \mathbf{x}^*\|^2}{2\beta T} + \beta(\sigma_F^2 + \overline{L}_F^2) + D_{\mathcal{X}} L_F \delta.$$

Therefore, we have the proof. □