Nonsmooth Implicit Differentiation:
Deterministic and Stochastic Convergence Rates

Riccardo Grazzi [1] **Massimiliano Pontil** [1 2] **Saverio Salzo** [3 1]

Abstract

We study the problem of efficiently computing the derivative of the fixed-point of a parametric non-differentiable contraction map. This problem has wide applications in machine learning, including hyperparameter optimization, meta-learning and data poisoning attacks. We analyze two popular approaches: iterative differentiation (ITD) and approximate implicit differentiation (AID). A key challenge behind the nonsmooth setting is that the chain rule does not hold anymore. We build upon the work by Bolte et al. (2022), who prove linear convergence of nonsmooth ITD under a piecewise Lipschitz smooth assumption. In the deterministic case, we provide a linear rate for AID and an improved linear rate for ITD which closely match the ones for the smooth setting. We further introduce NSID, a new stochastic method to compute the implicit derivative when the contraction map is defined as the composition of an outer map and an inner map which is accessible only through a stochastic unbiased estimator. We establish rates for the convergence of NSID, encompassing the best available rates in the smooth setting. We also present illustrative experiments confirming our analysis.

1. Introduction

In this paper, we study the problem of efficiently approximating a generalized derivative (or Jacobian) of the solution map of the parametric fixed point equation

$$w(\lambda) = \Phi(w(\lambda), \lambda) \quad (\lambda \in \mathbb{R}^m), \tag{1}$$

when Φ is not differentiable, but only piecewise differentiable. We address both the case that Φ can be explicitly evaluated, and the case that Φ has the composite form

$$\begin{aligned} \Phi(w, \lambda) &= G(T(w, \lambda), \lambda) \\ T(w, \lambda) &= \mathbb{E}[\hat{T}_\xi(w(\lambda), \lambda)], \end{aligned} \tag{2}$$

where the external map G can be evaluated, but the inner map T is accessible only via a stochastic estimator $\hat{T}_\xi$, with ξ a random variable.

A main motivation for computing the *implicit* derivative of (1) is provided by bilevel optimization, which aims to minimize an upper level objective function of $w(\lambda)$. Important examples are given by hyperparameter optimization and meta-learning (Franceschi et al., 2018; Lee et al., 2019), where (1) expresses the optimality conditions of a lower-level minimization problem. Further examples include learning a surrogate model for data poisoning attacks (Xiao et al., 2015; Muñoz-González et al., 2017), deep equilibrium models (Bai et al., 2019) or OptNet (Amos & Kolter, 2017). All these problems may present nonsmooth mappings Φ. For instance, consider hyperparameter optimization or data poisoning attacks for SVMs, or meta-learning for image classification, where Φ is evaluated through the forward pass of a neural net with RELU activations (Bertinetto et al., 2019; Lee et al., 2019; Rajeswaran et al., 2019). In addition, when such settings are applied to large datasets, evaluating the map Φ would be too costly, but we can usually apply stochastic methods through the composite stochastic structure in (2), where only T involves a computation on the full training set (e.g., a gradient descent step).

Nowadays, automatic differentiation techniques (Griewank & Walther, 2008) popular for deep learning, can also be used to efficiently, i.e. with a cost of the same order of that of approximating $w(\lambda)$, approximate Jacobian-vector (or vector-Jacobian) products of $w(\lambda)$ by relying only on an implementation of an iterative solver for problem (1). There are two main approaches to achieve this: ITerative Differentiation (ITD) (e.g., Maclaurin et al. (2015); Franceschi et al. (2017)), which differentiates through the steps of the solver for (1), and Approximate Implicit Differentiation (AID) (e.g., Pedregosa (2016); Lorraine et al. (2020)), which relies on approximately solving the linear system emerging from the implicit expression for the Jacobian-vector product. Despite the analysis of such methods has been usually done

[1]CSML, Istituto Italiano di Tecnologia, Genoa, Italy [2]Department of Computer Science, University College, London, UK [3]Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Universitá La Sapienza, Rome, Italy. Correspondence to: Riccardo Grazzi <riccardo.grazzi@iit.it>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

in the case that Φ is smooth, there are now several open source implementations relying on popular deep learning frameworks (e.g., Grazzi et al. (2020); Blondel et al. (2022); Liu & Liu (2021)), which practitioners can use even when Φ is not differentiable. However, when Φ is not differentiable despite existing algorithmic proposals (Ochs et al., 2015; Frecon et al., 2018), establishing theoretical convergence guarantees is challenging, since even if the solution map w is almost everywhere differentiable and the Clarke subgradient is well defined, the chain rule of differentiation, exploited by AID and ITD approaches, does not hold.

Recently Bolte & Pauwels (2021) introduced the notion of conservative derivatives as an effective tool to rigorously address automatic differentiation of neural networks with nondifferentiable activations (e.g., ReLU). Moreover, if $\Phi(\cdot, \lambda)$ is a contraction and under the general assumption that Φ is piecewise Lipschitz smooth with finite pieces, Bolte et al. (2022) provide an asymptotic linear convergence rate for deterministic ITD.[1] However, such rate is worse than that of the smooth case and we are not aware of any result of this type for the AID method and for the stochastic setting of problem (2), even when $G(v, \lambda) = v$. In particular the compositional structure (2) allows us to cover e.g., proximal stochastic gradient methods, which are a common and practical example of nonsmooth optimization algorithms, but it adds additional challenges since we do not have access to an unbiased estimator of Φ as for the smooth stochastic case studied in (Grazzi et al., 2021; 2023).

Contributions We present theoretical guarantees on AID and ITD for the approximation of the conservative derivative of the fixed point solution of (1), building upon the framework of Bolte et al. (2022). Specifically:

- We prove non-asymptotic linear convergence rates for deterministic ITD and AID which, from one hand extend the rates for the case where Φ is Lipschitz smooth given in (Grazzi et al., 2020), which are fully recovered as a special case, and on the other end, improve the result in (Bolte et al., 2022) for nonsmooth ITD. The given bounds indicate that AID converges faster than ITD, which we verify empirically. We also identify cases in which this difference in performance in favor of AID might be large due to nondifferentiable regions.

- We propose the first nonsmooth stochastic AID approach with proven convergence rates, which we name *nonsmooth stochastic implicit differentiation* (NSID). Notably, we prove that NSID can converge to a true conservative Jacobian-vector product with rate $O(1/k)$, where k is the number of samples, provided that the fixed-point problem is solved with rate $O(1/k)$.

- Finally, we provide experiments on two bilevel optimization problems, i.e. hyperparameter optimization and adversarial poisoning attacks, confirming our theoretical findings.

Related Work When Φ is differentiable and under some regularity assumptions, approximation guarantees have been established for AID and ITD approaches in the deterministic setting (Pedregosa, 2016; Grazzi et al., 2020), and for AID in the special case of the stochastic setting (2) where $G(v, \lambda) = v$ (Grazzi et al., 2021; 2023). Furthermore, several works established convergence rates and, in the stochastic setting, sample complexity results for bilevel optimization algorithms relying on AID and ITD approaches, see e.g., (Ghadimi & Wang, 2018; Ji et al., 2021; Arbel & Mairal, 2021; Chen et al., 2021).
Aside from (Bolte et al., 2022), in the nonsmooth case, Bertrand et al. (2020; 2022) present deterministic and sparsity-aware nonsmooth ITD and AID procedures together with asymptotic linear convergence guarantees when $w(\lambda)$ is the solution of a composite minimization problem where one component has a sum structure. Contrary to this work and to (Bolte et al., 2022), their results rely on some differentiability assumptions on the algorithms, which are verified after a finite number of iterations. For bilevel optimization, some recent works have provided stochastic algorithms with convergence rates for the special case where the lower-level problem has linear (Khanduri et al., 2023) or equality (Xiao et al., 2023) constraints.

2. Preliminaries

Notation If U and V are two nonempty sets, we denote by $F\colon U \rightrightarrows V$ a *set-valued mapping* which associates to an element of U a subset of V. A *selection* of F is a single-valued function $f\colon U \to V$ such that, for every $x \in U$, $f(x) \in F(x)$. We denote with $\|\cdot\|$ the Euclidean and operator norm when applied to vectors and matrices, respectively. Set inclusion is denoted by $\subset$. We define Minkowski operations on sets of matrices as follows: if $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^{n\times p}$ and $\mathcal{C} \subset \mathbb{R}^{p\times d}$ then

$$
\begin{aligned}
\mathcal{A} + \mathcal{B} &:= \{A + B \mid A \in \mathcal{A}, B \in \mathcal{B}\}, \\
\mathcal{A}\mathcal{C} &:= \{AC \mid A \in \mathcal{A}, C \in \mathcal{C}\} \\
\mathcal{A}^* &:= \{A^* \mid A \in \mathcal{A}\} \quad \text{with } * \in \{\top, -1\}.
\end{aligned}
$$

We let $\mathrm{co}(\mathcal{A})$ be the convex envelope of $\mathcal{A}$, and define $\|\mathcal{A}\|_{\sup} = \sup\{\|A\| \mid A \in \mathcal{A}\}$. It will be convenient to define for every $\mathcal{A} \subset \mathbb{R}^{n\times(p_1+p_2)}$ the map acting between sets of matrices, which we still denote by $\mathcal{A}$, such that for every $\mathcal{X} \subset \mathbb{R}^{p_1\times p_2}$

$$
\mathcal{A}(\mathcal{X}) := \mathcal{A}\begin{bmatrix} \mathcal{X} \\ I_{p_2} \end{bmatrix} := \big\{A_1 X + A_2 \,\big|\, [A_1, A_2] \in \mathcal{A}, X \in \mathcal{X}\big\}, \tag{3}
$$

[1] Therein, referred to as piggyback automatic differentiation.

where I_{p_2} is the identity matrix of dimensions $p_2 \times p_2$.

For any integer $r \geqslant 1$ we set $[r] = \{1, \ldots, r\}$. If $F \colon \mathbb{R}^{p_1+p_2} \to \mathbb{R}^d$ is differentiable, we denote by $F'(x) \in \mathbb{R}^{d\times(p_1+p_2)}$ the derivative of F (its Jacobian) at x and by $\partial_1 F(x) \in \mathbb{R}^{d\times p_1}$ and $\partial_2 F(x) \in \mathbb{R}^{d\times p_2}$ the partial derivatives of F with respect to the first and second block of variables respectively. Let $F : U \subset \mathbb{R}^d \to \mathbb{R}^p$, we say that F is smooth (nonsmooth) if it is differentiable (not differentiable) and *Lipschitz smooth with constant* L or L-*smooth* if it is smooth and $\exists L > 0$ such that $\forall x, y \in U$ $\|F'(x) - F'(y)\| \leqslant L\|x - y\|$. For a random vector $\xi \in \mathbb{R}^d$, we denote with $\mathbb{E}[\xi]$ its expectation and with $\mathrm{Var}[\xi] = \mathbb{E}\|\xi - \mathbb{E}[\xi]\|^2$ its variance. In our assumptions we will consider the class of the so called *definable* functions, which includes the large majority of functions used for machine learning applications (see Appendix A).

2.1. Conservative Derivatives

We provide some definitions related to path differentiability and sets of matrices and vectors. They are mostly borrowed, possibly with slight modifications, from (Bolte & Pauwels, 2021), where additional details can be found.

Definition 2.1 (Conservative Derivatives)**.** Let $U \subset \mathbb{R}^p$ be an open set and $F \colon U \subset \mathbb{R}^p \to \mathbb{R}^d$ be a locally Lipschitz continuous mapping. We say that a set-valued mapping $D_F \colon U \rightrightarrows \mathbb{R}^{d\times p}$ is a *conservative derivative* of F, if D_F has closed graph, nonempty compact values, and for every absolutely continuous curve $\gamma \colon [0,1] \to U \subset \mathbb{R}^p$ we have that, for almost every $t \in [0,1]$

$$\frac{d}{dt} F(\gamma(t)) = V\gamma'(t), \quad \forall\, V \in D_F(\gamma(t)). \tag{4}$$

The function F is called *path differentiable* if it admits a conservative derivative.

Conservative derivatives are extensively analyzed in (Bolte & Pauwels, 2021). Some key properties are that: (1) they are almost everywhere single-valued and equal to classical derivatives; (2) for path differentiable functions, the Clarke subgradient is the minimal conservative derivative up to a convex envelope; (3) chain rule holds for conservative derivatives; (4) locally Lipschitz definable mappings admit conservative derivatives. We also point out that – as it is usual for generalized derivatives – conservative derivatives are unique only up to a set of Lebesgue measure zero. This accounts for the fact that there are multiple ways to express a path differentiable function as a composition of others but applying the chain rule produces conservative derivatives that can differ but are always valid.

Similarly to (Bolte et al., 2022), to address the fact that conservative derivatives are set-valued mappings, we will use the following quantity to measure the error in the conservative derivative approximation.

Definition 2.2 (Excess)**.** Let $\mathcal{A}$ and $\mathcal{B}$ be two bounded subsets of matrices or vectors. The *excess*[2] *of* $\mathcal{A}$ *over* $\mathcal{B}$ is

$$e(\mathcal{A}, \mathcal{B}) := \sup_{A\in\mathcal{A}} \inf_{B\in B} \|A - B\|.$$

Note that $e(\mathcal{A}, \mathcal{B}) = 0 \implies \mathcal{A} \subset \mathcal{B}$ and $\|\mathcal{A}\|_{\sup} = e(\mathcal{A}, \{0\})$. The excess satisfies several properties similar to the ones of a distance, even though it is not symmetric (see Lemma B.1). Similarly to (Scholtes, 2012) we give the following concept of piecewise continuity and smoothness (which is slightly more general than that given in (Bolte & Pauwels, 2021)).

Definition 2.3. Let $F_1, \ldots, F_r \colon U \subset \mathbb{R}^p \to \mathbb{R}^d$ be continuous mappings defined on a nonempty open set U. A *continuous selection of* $F_1, \ldots, F_r$ is a continuous mapping $F \colon U \to \mathbb{R}^d$ such that for every $x \in U \colon F(x) \in \{F_1(x), \ldots, F_r(x)\}$. In such case the *active index set mapping* is the set-valued mapping $I_F \colon U \rightrightarrows [r]$, with $I_F(x) = \{i \in [r] \,|\, F_i(x) = F(x)\}$. Moreover, if the F_i's are differentiable we set $D^s_F \colon U \subset \mathbb{R}^p \rightrightarrows \mathbb{R}^{d\times p}$ such that

$$D^s_F(x) = \mathrm{co}(\{F'_i(x) \,|\, i \in I_F(x)\}), \tag{5}$$

where $F'_i(x)$ is the classical derivative (Jacobian) of F_i at x.

Theorem 2.4. *Let $F \colon U \subset \mathbb{R}^p \to \mathbb{R}^d$ be a continuous selection of definable and continuously differentiable mappings $F_1, \ldots, F_r \colon U \to \mathbb{R}^d$. Then F is definable if and only if $I_F \colon \mathbb{R}^p \rightrightarrows [r]$ is definable, and in such case D^s_F is a conservative derivative of F.*

We can also define partial conservative derivatives. If $p = p_1 + p_2$ and $F \colon U \subset \mathbb{R}^{p_1+p_2} \to \mathbb{R}^d$, we have $D_F \colon U \rightrightarrows \mathbb{R}^{d\times(p_1+p_2)}$ and we set $D_{F,1} \colon U \rightrightarrows \mathbb{R}^{d\times p_1}$ and $D_{F,2} \colon U \rightrightarrows \mathbb{R}^{d\times p_2}$ such that for $j \in \{1, 2\}$

$$D_{F,j}(x) = \big\{A_j \,|\, [A_1, A_2] \in D_F(x)\big\}.$$

Finally, we denote by $F'(x)$ an arbitrary element of $D_F(x)$ and by $\partial_1 F(x) \in \mathbb{R}^{d\times p_1}$ and $\partial_2 F(x) \in \mathbb{R}^{d\times p_2}$ the first and second block component of $F'(x)$ respectively, which yield the classical (partial) derivatives if F is differentiable. By building on (Bolte et al., 2022, Lemma 3), we prove the following result (the proof is in Appendix B).

Lemma 2.5. *Let $F \colon U \subset \mathbb{R}^p \to \mathbb{R}^d$ be a continuous definable selection of the definable Lipschitz smooth mappings $F_1, \ldots, F_r \colon U \to \mathbb{R}^d$. Let L_i be the Lipschitz constant of F'_i and set $L = \max_{1\leqslant i\leqslant r} L_i$. Then for every $x \in U$, there exist $R_x > 0$ such that for every $x' \in U$*

$$e(D^s_F(x'), D^s_F(x)) \leqslant L_x(x')\|x - x'\|, \tag{6}$$

where

$$L_x(x') := \begin{cases} L & \textit{if } \|x - x'\| \leqslant R_x \\ L + M_x/R_x & \textit{otherwise} \end{cases}$$

[2] e is referred in (Bolte et al., 2022) as gap, while the standard name is excess (Beer, 1993, Section 1.5).

and $M_x := \max_{i\in[m]} \min_{j\in I_F(x)} \|F'_i(x) - F'_j(x)\|$.

Note that in the smooth case ($r = 1$), (6) corresponds to global L-smoothness (since $M_x = 0$), while in general it is weaker. In particular, the quantity $L + \frac{M_x}{R_x}$ is well defined even when F is not differentiable at x, but blows up when x approaches a point of non-differentiability, e.g., for $\mathrm{ReLU}(x) = \max(0, x)$, $\lim_{x\to 0^+} M_x/R_x = \infty$, since if $x \neq 0$ $M_x = 1$ and $R_x = |x|$, while for $x = 0$, $M_x/R_x = 0$ since $M_x = 0$ and $R_x > 0$ can be chosen arbitrarily.

3. Differentiating a Parametric Fixed Point

Instances of Parametric Fixed Point Equations A general class of problems that can be recast in the form (1) is that of the parametric monotone inclusion problem

$$0 \in A_\lambda(w) + B_\lambda(w), \tag{7}$$

where $A_\lambda : \mathbb{R}^d \rightrightarrows \mathbb{R}^d$ and $B_\lambda : \mathbb{R}^d \to \mathbb{R}^d$ are multi-valued and single-valued maximal monotone operators respectively. These types of problems are at the core of convex analysis and can cover a number of optimizations problems including minimization problems as well as variational inequalities and saddle points problems. It is a standard fact (see (Bauschke & Combettes, 2017)) that (7) can be rewritten as the equation

$$R_{\gamma A_\lambda}(w - \gamma B_\lambda(w)) = w \quad (\gamma > 0),$$

where $R_{\gamma A_\lambda}$ is the resolvent of the operator γA_λ. This gives a fixed-point equation of a composite form, and comparing with (2), it is clear that we can also address situations in which $B_\lambda = \mathbb{E}[\hat{B}_\lambda(\cdot, \xi)]$. Bolte et al. (2024) investigates conservative derivatives of the solution map of such monotone inclusion problems in nonsmooth settings.

A special case of (7) is the minimization problem

$$\min_w \mathbb{E}[\hat{f}_\lambda(w, \xi)] + g_\lambda(w), \tag{8}$$

where $f_\lambda = \mathbb{E}\hat{f}_\lambda(\cdot, \xi)$ is convex L-smooth, while g_λ is convex lower semicontinuous extended-real valued. This can be cast into (2) by setting $\eta \in [0, 2/L[$, $\hat{T}_\xi(w, \lambda) = w - \eta\hat{\nabla} f_\lambda(w, \xi)$ and $G(w, \lambda) = \mathrm{Prox}_{\eta g_\lambda}(w)$ with $\mathrm{Prox}_h(x) = \arg\min_y (h(x) + (1/2)\|x - y\|^2)$ being the proximity operator of h. Several machine learning problems can be written in form (8) where g_λ is nonsmooth, e.g., LASSO, elastic net, (dual) SVM.

Main assumptions Referring to problem (1), when Φ is differentiable and $\|\partial_1\Phi(w(\lambda), \lambda)\| \leqslant q < 1$, by differentiating (1) we have

$$\begin{aligned} w'(\lambda) &= \partial_1\Phi(w(\lambda), \lambda)w'(\lambda) + \partial_2\Phi(w(\lambda), \lambda) \\ w'(\lambda) &= (I - \partial_1\Phi(w(\lambda), \lambda))^{-1}\partial_2\Phi(w(\lambda), \lambda). \end{aligned} \tag{9}$$

The first relation above shows that $w'(\lambda) \in \mathbb{R}^{d\times p}$ is a fixed point of the map $X \mapsto \partial_1\Phi(w(\lambda), \lambda)X + \partial_2\Phi(w(\lambda), \lambda)$. Here, dealing with the nonsmooth case, we will mimic the above formulas. The crucial assumption of our analysis is the following.

Assumption 3.1. Let $O_\Lambda \subset \mathbb{R}^m$ be an open set and $\Lambda \subset O_\Lambda$ be a nonempty closed and convex set.

(i) $\Phi : \mathbb{R}^d \times O_\Lambda \to \mathbb{R}^d$ is definable and a continuous selection of the L-Lipschitz smooth definable mappings $\Phi_1, \ldots, \Phi_r$ and we set $D_\Phi : \mathbb{R}^d \times O_\Lambda \rightrightarrows \mathbb{R}^{d\times(d+m)}$,

$$D_\Phi(u, \lambda) = D^s_\Phi(u, \lambda) = \mathrm{co}(\{\Phi'_i(u, \lambda) \mid i \in I_\Phi(u, \lambda)\}). \tag{10}$$

(ii) For all $(u, \lambda) \in \mathbb{R}^p \times O_\Lambda$, $\|D_{\Phi,1}(u, \lambda)\|_{\sup} \leqslant q < 1$.

Theorem 2.4 ensures that D_Φ, as defined in (10), is a conservative derivative of Φ. Moreover, recalling (4), it is easy to see that Assumption 3.1(ii) ensures that $\Phi(\cdot, \lambda)$ is a q-contraction and hence that there exists a unique fixed point of $\Phi(\cdot, \lambda)$ that we will denote by $w(\lambda)$. Finally, if $A \in D_{\Phi,1}(u, \lambda)$, we have $\|A\| < 1$ and hence $I - A$ is invertible. Thus, mimicking what happens for the smooth case in (9) one defines

$$D^{\mathrm{imp}}_w(\lambda) = \big\{(I - A_1)^{-1}A_2 \mid [A_1, A_2] \in D_\Phi(w(\lambda), \lambda)\big\} \tag{11}$$

$$D^{\mathrm{fix}}_w : \lambda \rightrightarrows \mathrm{fix}[D_\Phi(w(\lambda), \lambda)], \tag{12}$$

where $\mathrm{fix}[D_\Phi(u, \lambda)]$ is the unique fixed "point" of the map $\mathcal{X} \mapsto \mathcal{A}(\mathcal{X})$, where $\mathcal{A} = D_\Phi(u, \lambda)$ (see equation (3)), which acts between compact sets of $d \times m$ matrices. In (Bolte et al., 2021) it is proved that if Φ is path differentiable and Assumption 3.1(ii) holds, the set-valued mappings D^{imp}_w and D^{fix}_w are both conservative derivatives of $w(\lambda)$ and $D^{\mathrm{imp}}_w(\lambda) \subset D^{\mathrm{fix}}_w(\lambda)$.

Assumption 3.1 yields the following lemma through a direct application of Lemma 2.5.

Lemma 3.2. *Under Assumption 3.1(i), for every $\lambda \in \Lambda$, there exist $R_\lambda > 0$ such that for every $u \in \mathbb{R}^d$*

$$e(D_\Phi(u, \lambda), D_\Phi(w(\lambda), \lambda)) \leqslant C_\lambda(u)\|u - w(\lambda)\|,$$

where

$$C_\lambda(u) := \begin{cases} L & \text{if } \|u - w(\lambda)\| \leqslant R_\lambda \\ L + M_\lambda/R_\lambda & \text{otherwise} \end{cases} \tag{13}$$

and $M_\lambda := \max_{i\in[r]} \min_{j\in I_\Phi(w(\lambda),\lambda)} \|\Phi'_i(w(\lambda), \lambda) - \Phi'_j(w(\lambda), \lambda)\|$.

Lemma 3.2 can be used as a substitute for the Lipschitz smoothness of Φ with respect to the first variable, indeed note that in our analysis λ (and hence $w(\lambda)$) is fixed.

Remark 3.3. Our theoretical analysis requires only that Φ is definable piecewise smooth and that the inequality in Lemma 3.2 holds for some conservative derivatives of Φ, even if it is not computed according to (10). One such situation occurs for instance when Φ has the structure of a finite sum, that is, $\Phi = \sum_{i=1}^{n} \Phi^{(i)}$, where each $\Phi^{(i)}$ satisfies Assumption 3.1(i) with corresponding conservative derivative $D^s_{\Phi^{(i)}}$. Then, it is clear that Φ is still definable and piecewise Lipschitz smooth. Moreover, using the properties of conservative derivatives (see Corollary 4 in (Bolte & Pauwels, 2020)), $D_\Phi = \sum_{i=1}^{n} D^s_{\Phi^{(i)}}$ is a conservative derivative of Φ. Thus, using the property of the excess (see Lemma B.1(ii)) it directly follows that the inequality in Lemma 3.2, and hence our theory, still holds for such Φ.

4. Deterministic Iterative and Approximate Implicit Differentiation

We now formalize two deterministic methods for approximating the conservative derivative of the solution map w.

Iterative Differentiation (ITD) This method approximates $D_w^{\text{fix}}(\lambda)$ through the following iterative procedure, starting from $w_0(\lambda) \in \mathbb{R}^d$, $D_{w_0}(\lambda) = \{0\}$,

$$\text{for } t = 1, 2 \dots \quad \left\lfloor \begin{array}{l} w_t(\lambda) = \Phi(w_{t-1}(\lambda), \lambda) \\ D_{w_t}(\lambda) = D_\Phi(w_{t-1}(\lambda), \lambda) \begin{bmatrix} D_{w_{t-1}}(\lambda) \\ I_m \end{bmatrix}, \end{array} \right. \tag{14}$$

where we used the definition in (3). Note that the iteration for $D_{w_t}(\lambda)$ is based on the chain rule and results in a conservative derivative of $w_t(\lambda)$. This is the same set-valued iteration studied in (Bolte et al., 2022). We note that if $\Phi(\cdot, \lambda)$ is a q-contraction, it holds $\|w_t(\lambda) - w(\lambda)\| = O(q^t)$.

Approximate Implicit Differentiation with Fixed Point (AID-FP) An alternative method for approximating the implicit conservative derivative is the following. Assume that $w_t(\lambda)$ is generated by any algorithm converging to $w(\lambda)$ (for instance the one in (14)), then, starting from $D^0_{w_t}(\lambda) = \{0\}$, define

$$\text{for } k = 1, 2 \dots \quad \left\lfloor \; D^k_{w_t}(\lambda) = D_\Phi(w_t(\lambda), \lambda) \begin{bmatrix} D^{k-1}_{w_t}(\lambda) \\ I_m \end{bmatrix}. \right. \tag{15}$$

Efficient Implementation In practice we do not compute the full set-valued iterations in (14) and (15), but rather we select just one element at each iteration. Moreover, if we let $x \in \mathbb{R}^m$ and $y \in \mathbb{R}^d$, the ITD method can exploit automatic differentiation to efficiently compute an element of the conservative Jacobian-vector products $D_{w_t}(\lambda)^\top y$ (in reverse mode) and $D_{w_t}(\lambda) x$ (in forward mode). Similarly AID can efficiently compute an element in $D^k_{w_t}(\lambda)^\top y$. Thanks to Automatic Differentiation, if $k = t$ the standard implementation of both AID-FP and ITD has a cost in time of the same order of that of computing $w_t(\lambda)$. However, while AID-FP only uses $w_t(\lambda)$, ITD has a larger $\Theta(t)$ memory cost, since it needs to store the entire optimization trajectory $(w_i(\lambda))_{0 \leqslant i \leqslant t}$.

Convergence Guarantees In the Lipschitz smooth case Grazzi et al. (2020) proved non-asymptotic linear convergence rates for both methods, revealing that AID-FP is slightly faster than ITD. We now extend this analysis to nonsmooth ITD and AID-FP, focusing on the convergence of the set-valued iterations in (14) and (15). Thanks to Lemma 3.2 and the properties of the excess, the proof (in Appendix C) can proceed similarly to the one for the smooth case.

Theorem 4.1 (nonsmooth ITD and AID-FP Rates)**.** *Let Assumption 3.1 hold. For every $\lambda \in \Lambda$, let R_λ and M_λ be the quantities defined in Lemma 3.2 and $B_\lambda := \|D_{\Phi,2}(w(\lambda), \lambda)\|_{\sup}$. For every $t, k \in \mathbb{N}$, let $\Delta_t = \|w_t(\lambda) - w(\lambda)\|$, $\delta_\lambda(t) := \mathbb{1}\{\Delta_t > R_\lambda\}$ and $\bar{\delta}_\lambda(t) = t^{-1} \sum_{i=0}^{t-1} \delta_\lambda(i)$. Then the following hold.*

(i) *The ITD iteration in* (14) *satisfies*

$$\begin{aligned} e(D_{w_t}(\lambda), D_w^{\text{fix}}(\lambda)) \leqslant \frac{B_\lambda}{1-q} q^t \\ + \frac{B_\lambda + 1}{1-q} \Big(L + \frac{M_\lambda}{R_\lambda} \bar{\delta}_\lambda(t) \Big) \Delta_0 \, t \, q^{t-1}. \end{aligned} \tag{16}$$

(ii) *The AID-FP iteration in* (15) *satisfies*

$$\begin{aligned} e(D^k_{w_t}(\lambda), D_w^{\text{fix}}(\lambda)) \leqslant \frac{B_\lambda}{1-q} q^k \\ + \frac{B_\lambda + 1}{1-q} \Big(L + \frac{M_\lambda}{R_\lambda} \delta_\lambda(t) \Big) \frac{1-q^k}{1-q} \Delta_t. \end{aligned} \tag{17}$$

Moreover, if $w_t(\lambda) = \Phi(w_{t-1}(\lambda), \lambda)$, then $\Delta_t \leqslant q\Delta_{t-1} \leqslant q^t \Delta_0$ and there exists $\tau_\lambda \in \mathbb{N}$ such that $\delta_\lambda(t) = \mathbb{1}\{t < \tau_\lambda\}$ and thus $\delta_\lambda(t) \leqslant \bar{\delta}_\lambda(t) \leqslant 1$.

To compare the two rates in Theorem 4.1, let $t = k$ and $w_t(\lambda) = \Phi(w_{t-1}(\lambda), \lambda)$, so that both AID-FP and ITD have time complexity of the order of computing $w_t(\lambda)$. In that situation, since $1 - q^k = 1 - q^t < q^{-1}(1-q)t$ and $\delta_\lambda(t) \leqslant \bar{\delta}_\lambda(t)$, the upper bound of AID-FP is always lower than that of ITD. Moreover, if we let $\kappa = (1-q)^{-1}$ to play a similar role to the condition number, we observe that both methods converge linearly: AID-FP as $O(\kappa^2 e^{-t/\kappa})$, while ITD slightly slower as $O(\kappa t e^{-t/\kappa})$. When $t \geqslant \tau_\lambda$, $\delta_\lambda(t) = 0$ while $\bar{\delta}_\lambda(t) = \tau_\lambda / t$, which might cause a wide difference between the two bounds if M_λ / R_λ is large, and such ratio can get arbitrarily large the closer $(w(\lambda), \lambda)$ is to regions where Φ is not differentiable. Finally, if we replace

Lemma 3.2 with the L-smoothness of Φ, we essentially recover the same bounds reported by Grazzi et al. (2020), where the terms $\delta_\lambda, \bar{\delta}_\lambda$ do not appear.

The work by (Bolte et al., 2022) also reports a rate for nonsmooth ITD of $O((\sqrt{q}+\epsilon)^t)$ for arbitrary $\epsilon > 0$. However, this rate does not match the best available rate for smooth ITD (Grazzi et al., 2020). Theorem 4.1 (in (16)) fills this gap since it achieves[3] an improved rate of $O((q+\epsilon)^t)$. Moreover, our rate is more explicit, since it does not involve any arbitrary ϵ.

We conclude the section by noting that Theorem 4.1 ensures that the sequence constructed by selecting one element at each iteration in (14) and (15), is guaranteed to converge, up to a subsequence, to the set $D_w^{\text{fix}}(\lambda)$.

5. Nonsmooth Stochastic Implicit Differentiation

In this section we study the stochastic fixed point formulation in (2) and present an algorithm that, given a random vector $y \in \mathbb{R}^d$ and an approximate solution $w_t(\lambda)$, efficiently approximates an element of $D_w^{\text{imp}}(\lambda)^\top y$ accessing only $\hat{T}_\xi$, G and fixed selections of their conservative derivatives. Similarly to deterministic AID, here we assume that $w_t(\lambda)$ is generated by a stochastic algorithm which converges in mean square to $w(\lambda)$. Several algorithms can ensure such convergence guarantees for the composite minimization problems in (8) (e.g, Rosasco et al. (2020) provide a proximal stochastic gradient algorithm with rate $O(1/t)$) and composite monotone inclusions (Rosasco et al., 2014). We recall that for a path differentiable function $F\colon U \subset \mathbb{R}^{p_1+p_2} \to \mathbb{R}^d$, we denote by F' an arbitrary selection of D_F and by $\partial_1 F(x) \in \mathbb{R}^{d\times p_1}$ and $\partial_2 F(x) \in \mathbb{R}^{d\times p_2}$ the first and second block component of $F'(x)$ respectively, so that we can write $F'(x) = [\partial_1 F(x), \partial_2 F(x)]$.

We consider the following assumptions

Assumption 5.1.

(i) T and G satisfy Assumption 3.1(i) individually, with constant L_T and L_G respectively. Let T' and G' be selections of the conservative derivatives D_T and D_G respectively. Also, $\Phi(u,\lambda) = G(T(u,\lambda),\lambda)$.

(ii) For every $(u,\lambda) \in \mathbb{R}^d \times \Lambda$, $\|D_{T,1}(u,\lambda)\|_{\sup} \leqslant 1$ and $\|D_{G,1}(u,\lambda)\|_{\sup} \leqslant 1$ and either T or G satisfies Assumption 3.1(ii).

(iii) $y \in \mathbb{R}^d$ is a random vector.

Assumption 5.2. The random variable ξ takes values in Ξ and for every $x \in \Xi$

(i) $\hat{T}_x\colon \mathbb{R}^d \times O_\Lambda \to \mathbb{R}^d$ and $\mathbb{E}[\hat{T}_\xi(u,\lambda)] = T(u,\lambda)$.

(ii) $\hat{T}_x$ is path differentiable and $\hat{T}'_x$ is a selection of its conservative derivative $D_{\hat{T}_x}$ and there exist $\sigma_1, \sigma_2, \sigma'_1, \sigma'_2 \geqslant 0$ such that for every $u \in \mathbb{R}^d$, $\lambda \in \Lambda$

$$\mathbb{E}[\hat{T}'_\xi(u,\lambda)] = T'(u,\lambda) \in D_T(u,\lambda),$$

$$\operatorname{Var}[\hat{T}_\xi(u,\lambda)] \leqslant \sigma_1 + \sigma_2 \|u - T(u,\lambda)\|^2,$$

$$\operatorname{Var}[\partial_1 \hat{T}_\xi(u,\lambda)] \leqslant \sigma'_1, \quad \operatorname{Var}[\partial_2 \hat{T}_\xi(u,\lambda)] \leqslant \sigma'_2.$$

where $\hat{T}'_x(u,\lambda) = [\partial_1 \hat{T}_x(u,\lambda), \partial_2 \hat{T}_x(u,\lambda)]$.

Remark 5.3. The above assumptions can be satisfied in the following situations: (1) G is nonsmooth, e.g., some proximity operator or the projection on some simple constraints, while T and $\hat{T}_x$ are smooth (e.g., one step of gradient descent of a twice differentiable loss); (2) in view of Remark 3.3, when $T = \frac{1}{n}\sum_{i=1}^n \hat{T}_i$, $T' = \frac{1}{n}\sum_{i=1}^n \hat{T}'_i$ with $\hat{T}' \in D^s_{\hat{T}_i}$ and ξ is uniformly distributed on $[n]$.

Assumption 5.1 ensures that D_Φ obtained via the chain rule for conservative derivatives in (Bolte & Pauwels, 2021) (see Appendix D) is a conservative derivative of Φ and that $\|D_{\Phi,1}(u,\lambda)\|_{\sup} \leqslant q < 1$. Thus, $w(\lambda)$ is well defined and it has conservative derivatives D_w^{imp} and D_w^{fix}. Assumption 5.2 is a nonsmooth generalization of the corresponding one in (Grazzi et al., 2021; 2023). Finally, recalling (11), if we set $\partial_2\Phi(u,\lambda) = \partial_1 G(T(u,\lambda),\lambda)\partial_2 T(u,\lambda) + \partial_2 G(T(u,\lambda),\lambda)$ then

$$\partial_2\Phi(w(\lambda),\lambda)^\top v(w(\lambda),\lambda) \in D_w^{\text{imp}}(\lambda)^\top y \tag{18}$$

where, for every $u \in \mathbb{R}^d$, $v(u,\lambda)$ is a solution of the linear system

$$(I - \partial_1 T(u,\lambda)^\top \partial_1 G(T(u,\lambda),\lambda)^\top) v = y. \tag{19}$$

Algorithm and convergence guarantees Our method is inspired by (18) and (19) but it uses mini-batch estimators of T and $\partial_2\Phi$. To that purpose we assume to have two independent sets of samples $\boldsymbol{\xi}^{(1)} = (\xi_j^{(1)})_{1\leqslant j\leqslant J}$ and $\boldsymbol{\xi}^{(2)} = (\xi_i^{(2)})_{1\leqslant i\leqslant k}$, being i.i.d. copies of the random variable ξ. Moreover, we define the path differentiable functions

$$\bar{T}(u,\lambda) = \frac{1}{J}\sum_{j=1}^J \hat{T}_{\xi_j^{(1)}}(u,\lambda), \quad \bar{\Phi}(u,\lambda) = G(\bar{T}(u,\lambda),\lambda).$$

In fact our approach first replaces the linear system (19) with

$$(I - \partial_1 T(w_t(\lambda),\lambda)^\top \partial_1 G(\bar{T}(w_t(\lambda),\lambda),\lambda)^\top) v = y, \tag{20}$$

where the solution is in turn approximated by a stochastic sequence $(v_k)_{k\in\mathbb{N}}$, which has access only to $\hat{T}_x, G$, and $w_t(\lambda)$. Second, it outputs $\partial_2\bar{\Phi}(w_t(\lambda),\lambda)^\top v_k$, where for any $u \in \mathbb{R}^d$, $\lambda \in O_\Lambda$,

$$\partial_2\bar{\Phi}(u,\lambda) = \partial_1 G(\bar{T}(u,\lambda),\lambda)\partial_2 \bar{T}(u,\lambda) + \partial_2 G(\bar{T}(u,\lambda),\lambda), \tag{21}$$

[3] For any $\epsilon \in [0, 1-q]$, $\exists C > 0$ such that $tq^{t-1} \leqslant C(q+\epsilon)^t$.

with $\bar{T}'(u,\lambda) := [\partial_1 \bar{T}(u,\lambda), \partial_2 \bar{T}(u,\lambda)] = \frac{1}{J}\sum_{j=1}^{J} \hat{T}'_{\xi_j^{(1)}}(u,\lambda)$, which thanks to the chain rule is an element of a partial conservative derivative of $\bar{\Phi}$ (see also Appendix D).

We now provide a general bound for the mean square error of an estimator of an element of the Jacobian vector product $D_w^{\mathrm{imp}}(\lambda)^\top y$, which is agnostic with respect to the algorithms solving the fixed point equation (1) and the linear system (20). The proof (in Appendix D) uses similar techniques as the one for the smooth case in (Grazzi et al., 2021; 2023).

Assumption 5.4. Let $\rho_\lambda \colon \mathbb{N} \to \mathbb{R}_+$, $\sigma_\lambda \colon \mathbb{N} \to \mathbb{R}_+$ be such that $\lim_{t\to+\infty} \rho_\lambda(t) = 0$, $\lim_{k\to+\infty} \sigma_\lambda(k) = 0$.

(i) $(w_t(\lambda))_{t\in\mathbb{N}}$ is a sequence of random vectors in $\mathbb{R}^d$ and

$$\mathbb{E}[\|w_t(\lambda) - w(\lambda)\|^2] \leqslant \rho_\lambda(t),$$

(ii) For every $(u_1,u_2) \in \mathbb{R}^d \times \mathbb{R}^d$, $(v_k(u_1,u_2))_{k\in\mathbb{N}}$ is a sequence of random vectors in $\mathbb{R}^d$ which is independent on $(w_t(\lambda))_{t\in\mathbb{N}}$ and such that

$$\mathbb{E}[\|v_k(u_1,u_2) - \bar{v}(u_1,u_2)\|^2 \mid y] \leqslant \|y\|^2 \sigma_\lambda(k),$$

where $\bar{v}(u_1,u_2)$ is the unique fixed point of the affine mapping $v \mapsto \partial_1 T(u_1,\lambda)^\top \partial_1 G(u_2,\lambda)^\top v + y$.

(iii) The r.v. y satisfies $\mathbb{E}[\|y\|^2 \mid w_t(\lambda)] \leqslant b^2$ a.s.

Theorem 5.5. *Under Assumption 5.1, 5.2, and 5.4, let $\kappa = (1-q)^{-1}$. We define the estimator*

$$(w'(\lambda)^\top y)\hat{} := \partial_2 \bar{\Phi}(w_t(\lambda),\lambda)^\top v_k\big(w_t(\lambda), \bar{T}(w_t(\lambda),\lambda)\big).$$

Then for every $t,k,J \in \mathbb{N}$, we have

$$\mathbb{E}\Big[e\big((w'(\lambda)^\top y)\hat{}, D_w^{\mathrm{imp}}(\lambda)^\top y\big)^2\Big] = b^2 \times O\left(\sigma_\lambda(k) + \kappa^4\left(J^{-1} + \rho_\lambda(t)\right)\right).$$

We preset the full procedure, named nonsmooth stochastic implicit differentiation (NSID), in Algorithm 1, where the sequence v_k considered in Assumption 5.4(ii) is generated by a simple stochastic fixed-point iteration algorithm (described in (Grazzi et al., 2021) and recalled in Appendix D) with step sizes $(\eta_i)_{1\leqslant i\leqslant k}$.

Note that all steps can be efficiently implemented via automatic differentiation by using only vector-valued function evaluations and conservative Jacobian-vector products without the expensive computation of the full matrix derivatives. Also, using a fixed selection for the conservative derivative of $\hat{T}_x$ and G corresponds to the standard implementation.

If $G(\cdot,\lambda)$ is the identity and T is smooth, NSID reduces to the same procedure given in (Grazzi et al., 2023), which also provide the bound $O\left(\sigma_\lambda(k) + \kappa^2 J^{-1} + \kappa^4 \rho_\lambda(t)\right)$ in Theorem 7. Compared to the bound given in Theorem 5.5, we note that the only difference is in the constant in front of the term J^{-1}, which we believe may be related to the term G. Indeed handling a general G provides an additional challenge since we do not have access anymore to an unbiased estimator of Φ. However, we could overcome this issue by using different samples sequences for the two factors occuring in $\hat{\Psi}$. Incidentally, one of those sequences can be the one used to compute a mini-batch estimator of $\partial_2\Phi$. Ultimately, this does not call for any additional samples compared to the smooth version, but it could worsen some constants in the bound.

Algorithm 1 NSID

1: **Input:** $k, J \in \mathbb{N}$, $w_t(\lambda), y \in \mathbb{R}^d$, $\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}$
2: $\bar{T}_t(\lambda) \leftarrow \bar{T}(w_t(\lambda),\lambda)$ *(using $\boldsymbol{\xi}^{(1)}$)*
3: $\hat{\Psi} \colon (v,x) \mapsto \partial_1 \hat{T}_x(w_t(\lambda),\lambda)^\top \partial_1 G(\bar{T}_t(\lambda),\lambda)^\top v + y$
4: **for** $i = 1$ **to** k **do**
5: $\quad v_i \leftarrow (1-\eta_i)v_{i-1} + \eta_i \hat{\Psi}(v_{i-1}, \xi_i^{(2)})$
6: **end for**
7: **Return** $(w'(\lambda)^\top y)\hat{} := \partial_2 \bar{\Phi}(w_t(\lambda),\lambda)^\top v_k$

Finally, we specialize the result of Theorem 5.5 to Algorithm 1. The proof is in Appendix D.

Theorem 5.6. *Under Assumption 5.1, 5.2, and 5.4(i)(iii), let $(w'(\lambda)^\top y)\hat{}$ be generated by Algorithm 1 with $\eta_i = \Theta(i^{-1})$ and assume that $\rho_\lambda(t) = O(\kappa^\alpha t^{-1})$, with $\alpha > 0$. Then*

$$\mathbb{E}\Big[e\big((w'(\lambda)^\top y)\hat{}, D_w^{\mathrm{imp}}(\lambda)^\top y\big)^2\Big] = O\left(\frac{\kappa^5}{k} + \frac{\kappa^4}{J} + \frac{\kappa^{4+\alpha}}{t}\right).$$

Hence if $J = O(t)$, $k = O(t)$, the mean square error is $\leqslant \epsilon$ after $O(\kappa^{5+\alpha}\epsilon^{-1})$ samples.

Note that the sample complexity $O(\epsilon^{-1})$ matches the performance of SGD for minimizing strongly convex and Lipschitz smooth functions (Bottou et al., 2018), which are a special cases of Problem (2). Furthermore it is the same one that the SID algorithm by Grazzi et al. (2021; 2023) attains when $G(v,\lambda) = v$ and Φ is Lipschitz smooth. A limitation is the choice of step-sizes (η_i), problematic in practice.

6. Application to Bilevel Optimization

In this section, we consider the following bilevel problem with the fixed point problem in (1) at the lower level

$$\min_{\lambda\in\Lambda} \{E(w(\lambda),\lambda) : \; w(\lambda) = \Phi(w(\lambda),\lambda)\}, \tag{22}$$

where $E \colon \mathbb{R}^d \times O_\Lambda \to \mathbb{R}$. We will show how we can use AID-FP, ITD and NSID to approximate an element of the conservative derivative of the bilevel objective $f(\lambda) := E(w(\lambda),\lambda)$ and retain the same convergence rates.

In addition to the requirement that Φ satisfies Assumption 3.1, we also make the hypothesis that E satisfies the first item of same assumption with corresponding conservative derivative $D_E = D_E^s$. Therefore, applying the usual chain rule, we have that for $* \in \{\text{imp}, \text{fix}\}$

$$D_f^*(\lambda) := D_E(w(\lambda), \lambda) \begin{bmatrix} D_w^*(\lambda) \\ I_m \end{bmatrix}$$

is a conservative derivative for f. We also let $f_t(\lambda) := E(w_t(\lambda), \lambda)$, where $w_t(\lambda)$ is an approximate solution for the fixed point problem.

Deterministic Case The approximate derivatives

$$\textbf{(BITD)} \quad D_{f_t}(\lambda) := D_E(w_t(\lambda), \lambda) \begin{bmatrix} D_{w_t}(\lambda) \\ I_m \end{bmatrix}$$

$$\textbf{(BAID-FP)} \quad D_{f_t}^k(\lambda) := D_E(w_t(\lambda), \lambda) \begin{bmatrix} D_{w_t}^k(\lambda) \\ I_m \end{bmatrix}$$

converge to $D_f^{\text{fix}}(\lambda)$ with the same rate as ITD and AID (Theorem E.3).

Stochastic Case We study the bilevel problem

$$\begin{aligned} &\min_{\lambda \in \Lambda} f(\lambda) := \mathbb{E}[\hat{E}_\zeta(w(\lambda), \lambda)], \\ &w(\lambda) = G\big(\mathbb{E}[\hat{T}_\xi(w(\lambda), \lambda)], \lambda\big). \end{aligned} \tag{23}$$

where ζ is a random variable. We consider Algorithm 2, which additionally computes $\bar{E}'(w_t(\lambda), \lambda) := J_1^{-1} \sum_{j=1}^{J_1} \hat{E}'_{\zeta_j^{(1)}}(w_t(\lambda), \lambda)$, a minibatch gradient estimator of $E' \in D_E$, using the sequence $\boldsymbol{\zeta}^{(1)} = (\zeta^{(1)})_{1 \leqslant j \leqslant J_1}$ of i.i.d. copies of ζ.

Algorithm 2 NSID-Bilevel

1: **Input:** $k, J_1, J_2 \in \mathbb{N}$, $w_t(\lambda) \in \mathbb{R}^d$, $\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}, \boldsymbol{\zeta}^{(1)}$
2: Compute $\bar{E}'(w_t(\lambda), \lambda)$ *(using $\boldsymbol{\zeta}^{(1)}$)*
3: $y \leftarrow \partial_1 \bar{E}(w_t(\lambda), \lambda)^\top$
4: $r(w_t(\lambda), \lambda) \leftarrow \text{NSID}(k, J_2, w_t(\lambda), y, \boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)})$
5: **Return** $\hat{\nabla} f(\lambda)^\top := r(w_t(\lambda), \lambda)^\top + \partial_2 \bar{E}(w_t(\lambda), \lambda)$

With additional mild assumptions on the variance of $\hat{E}$ and when $E(\cdot, \lambda)$ is Lipschitz, we recover the same convergence rates as NSID, but this time to $D_f^{\text{imp}}(\lambda)$ (Theorem E.6).

On the convergence of the bilevel problem Despite these encouraging results and the fact that in the smooth case several works provide convergence rates to a stationary point of the gradient of f (Ji et al., 2021; Arbel & Mairal, 2021; Grazzi et al., 2023, and others), proving such type of results or even asymptotic convergence (without rates) in our nonsmooth case is more challenging and we leave it for future work. One crucial issue is that in the analysis, the constant defined in Lemma 2.5, which we use in place of that of Lipschitz smoothness, cannot be properly controlled on the whole Λ as required in the smooth case: it becomes arbitrarily large when $(w(\lambda), \lambda)$ approaches nondifferentiable regions of Φ.

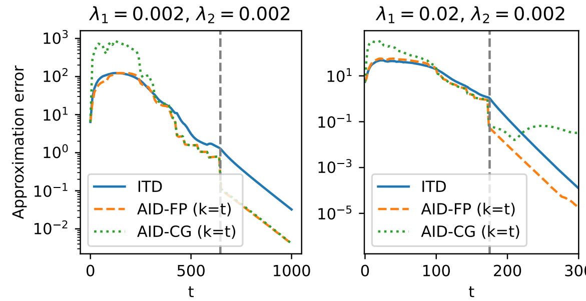

Figure 1. AID vs ITD for synthetic elastic-net. t corresponds to the number of steps to find an approximate fixed point and the dashed vertical line is the step where the support is identified. AID-FP converges faster than ITD; note that after support identification there is a wide gap between the methods, as anticipated by our theoretical bounds. AID-CG does not converge in plot on the right, probably due to sensitivity to numerical errors.

7. Experiments

The experiments aim to achieve two primary goals. Firstly, we aim to empirically demonstrate the practical manifestation of distinct behaviors between AID and ITD, as outlined in the theoretical findings of Section 4. Emphasis is placed on aspects specific to the nonsmooth analysis. Secondly, we intend to evaluate the empirical performance of our stochastic method NSID presented in Algorithm 1. We implement NSID by relying on PyTorch automatic differentiation for the computation of Jacobian-vector products. For AID and ITD, we use the existing PyTorch implementations[4]. We provide the code to reproduce our experiments at https://github.com/prolearner/nonsmooth_implicit_diff

Experimental Setup We consider two problems where we are interested in approximating an element of the conservative Jacobian-vector product of the solution map $D_w^{\text{fix}}(\lambda)^\top y$ for $y \in \mathbb{R}^d$. With a focus on bilevel optimization, we set y as the gradient of the validation loss in $w_t(\lambda)$, as explained in Section 6, while to compute the approximation error we use the procedure described in Appendix F.1.

Elastic Net Let $(X, y) \in \mathbb{R}^{n \times d} \times \mathbb{R}^n$ be a training regression dataset. The elastic net solution $w(\lambda)$ is the minimizer of the objective function $\frac{1}{n}\|Xw - y\|^2 + \lambda_1 \|w\|_1 + \frac{\lambda_2}{2}\|w\|_2^2$, where $\lambda = (\lambda_1, \lambda_2)$ are the regularization hyperparameters.

[4] https://github.com/prolearner/hypertorch

Data Poisoning We consider a data poisoning scenario similar to the one in (Xiao et al., 2015), where an attacker would like to corrupt part of the training dataset by adding noise in order to decrease the accuracy of an elastic-net regularized logistic regression model after training. In particular, let c be the number of classes and $(\tilde{X}, \tilde{y}) \in \mathbb{R}^{n' \times d} \times [c]^{n'}$ be the examples to corrupt while $(X, y) \in \mathbb{R}^{n \times d} \times [c]^n$ are the clean ones. Let also $\Gamma \in \mathbb{R}^{n' \times d}$ represent the noise and define the data poisoning elastic net solution as $w(\Gamma) = \arg\min_{w \in \mathbb{R}^d} f(\Gamma, w) + \lambda_1 \|w\|_1 + \frac{\lambda_2}{2}\|w\|_2^2$, where $f(\Gamma, w) = \ell(Xw, y)/2 + \ell((\tilde{X} + \Gamma)w, \tilde{y})/2$ and ℓ is the cross-entropy loss. A strategy to find Γ would be by approximating an element of the conservative Jacobian-vector product $D_w(\Gamma)^\top y$ where y is the gradient of the cross-entropy loss on an hold out set. This setting is of particular interest, since Γ is high dimensional and hence zero-order methods like grid or random search are less appropriate. For both settings and all considered methods, we find an approximate solution $w_t(\lambda)$ always by iterating the contraction map which describes the iterates of the deterministic iterative soft-thresholding algorithm (see e.g., (Combettes & Wajs, 2005)). Although this may be inefficient in the stochastic setup, it yields a fairer comparison, since both the stochastic and deterministic algorithms will have the same $w_t(\lambda)$ as input. Additional details are in the appendix.

AID and ITD We consider the Elastic Net scenario and construct a synthetic supervised linear regression problem with 100 training examples and 100 features, of which 30 are informative. As the fixed point map Φ we use one step of iterative soft-thresholding. The appropriate choice for the step-size guarantees that Φ is a contraction, in our case we set it equal to $2/(L + \mu + 2\lambda_2)$, where L and μ are the largest and smallest eigenvalues values of $n^{-1}X^\top X$.

We compare ITD, AID-FP, and AID-CG a variant of AID which uses conjugate gradient to solve the linear system (Grazzi et al., 2020), where the vector y for the Jacobian-vector product is the gradient of the square loss on a validation set, computed on the t-the iterate $(\nabla E(w_t(\lambda), \lambda)$ where E is defined in (22)). In Figure 1 we can see two runs, each one for two particular choices of λ which highlight a wide gap in performance after support identification, i.e. when both $w_t(\lambda)$ and $w(\lambda)$ have the same non-zero elements. This was predicted by Theorem 4.1, since support identification coincides with $\|w_t(\lambda) - w(\lambda)\| \leqslant R_\lambda$.

Stochastic Methods We compare our stochastic method NSID (Algorithm 1) against AID-FP and the algorithm SID in (Grazzi et al., 2023). In particular, for NSID $\hat{T}_x$ corresponds to one step of gradient descent on a minibatch of training points, while G is soft-thresholding. We implement SID by setting in NSID $G(u, \lambda) = u$ and using $\hat{\Phi}_\xi(u, \lambda) = G(\hat{T}_\xi(u, \lambda), \lambda)$ in place of $\hat{T}_\xi$. Note that although the theoretical convergence guarantee for SID do not hold due to $\hat{\Phi}_\xi$ being biased, the performance of SID still effectively measures the impact of such bias in practice.

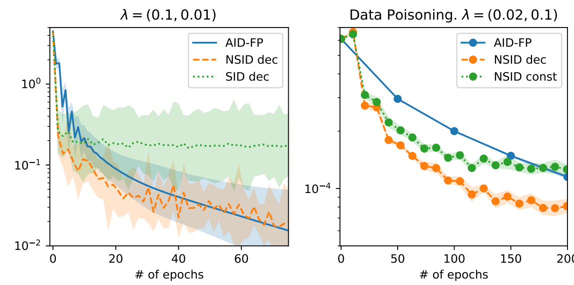

Figure 2. Stochastic implicit differentiation for elastic net (left) and data poisoning (right) with constant (const) and decreasing (dec) step sizes. Mean (solid line) and the geometric standard deviation (shaded region) of the approximation error over 10 runs. SID does not converge on elastic net for this specific choice of λ and diverges in data poisoning (hence we do not report it), while NSID converges faster (at the beginning) than the deterministic AID-FP. Note that decreasing step-sizes provide a favorable choice.

We consider both the elastic net and the data poisoning setups; see the appendix for more information. The results are shown in Figure 2. For elastic net, each run corresponds to a different sampling of the covariance matrix, training points, true solution vector and minibatches used by the stochastic algorithms. For Data poisoning, each run corresponds to different sampling of the noise Γ (sampled from a normal and then each component projected in $[-.1, .1]$) and the mini-batches used by the stochastic algorithms. For AID-FP, each epoch corresponds to one iteration, since it uses the entire dataset, while for NSID and SID the number of epochs is equal to $(k + J)(n' + n)/b$, where b is the minibatch size, which we set to 10% of the training set, i.e. $b = (n' + n)/10$. Note that for each point in the plots for NSID and SID, we need to start the algorithm from scratch since we increase both k and J simultaneously. In particular we set $k = J$ for elastic net and $J = \lceil k/20 \rceil$ for data poisoning.

8. Conclusions

We established convergence guarantees for nonsmooth implicit differentiation methods. Leveraging the foundation laid by (Bolte et al., 2022), we developed tools facilitating the translation of results from the smooth case. This allowed us to provided non-asymptotic linear convergence rates for AID-FP and ITD, focusing on deviations from their smooth analogs. Additionally, we introduced NSID, a principled stochastic algorithm. Numerical experiments underscored the distinctive behaviors of AID-FP and ITD, along with the good performance of NSID, which may be useful in large scale bilevel optimization problems in the future. Despite our results, establishing rates for solving nonsmooth bilevel problems is still challenging and we leave it for future work.

Acknowledgements

This work was supported in part from the PNRR MUR Project PE000013 CUP J53C22003010006 "Future Artificial Intelligence Research (FAIR)", funded by the European Union – NextGenerationEU, and EU Project ELSA under grant agreement No. 101070617.

Impact Statement

This paper presents work whose goal is to advance theoretical understanding of Machine learning techniques. There are several potential societal consequences of our work, none which we feel must be specifically highlighted here. Our principled algorithms can help make machine learning methods more reliable in practical scenarios.

References

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.

Arbel, M. and Mairal, J. Amortized implicit differentiation for stochastic bilevel optimization. In *International Conference on Learning Representations*, 2021.

Bai, S., Kolter, J. Z., and Koltun, V. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.

Bauschke, H. H. and Combettes, P. L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces. 2nd Edition*. Springer International Publishing, 2017.

Beer, G. *Topologies on Closed and Closed Convex Sets*, volume 268. Springer Science & Business Media, 1993.

Bertinetto, L., Henriques, J., Torr, P., and Vedaldi, A. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations (ICLR), 2019*, 2019.

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. *The Journal of Machine Learning Research*, 23(1):6680–6722, 2022.

Blondel, M., Berthet, Q., Cuturi, M., Frostig, R., Hoyer, S., Llinares-López, F., Pedregosa, F., and Vert, J.-P. Efficient and modular implicit differentiation. *Advances in Neural Information Processing Systems*, 35:5230–5242, 2022.

Bolte, J. and Pauwels, E. A mathematical model for automatic differentiation in machine learning. *Advances in Neural Information Processing Systems*, 33:10809–10819, 2020.

Bolte, J. and Pauwels, E. Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. *Mathematical Programming*, 188:19–51, 2021.

Bolte, J., Le, T., Pauwels, E., and Silveti-Falls, T. Nonsmooth implicit differentiation for machine-learning and optimization. *Advances in neural information processing systems*, 34:13537–13549, 2021.

Bolte, J., Pauwels, E., and Vaiter, S. Automatic differentiation of nonsmooth iterative algorithms. *Advances in Neural Information Processing Systems*, 35:26404–26417, 2022.

Bolte, J., Pauwels, E., and Silveti-Falls, A. Differentiating nonsmooth solutions to parametric monotone inclusion problems. *SIAM Journal of Optimization*, 34:71–97, 2024.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.

Chen, T., Sun, Y., and Yin, W. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Advances in Neural Information Processing Systems*, 34:25294–25307, 2021.

Combettes, P. L. and Wajs, V. R. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

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.

Frecon, J., Salzo, S., and Pontil, M. Bilevel learning of the group lasso structure. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Grazzi, R., Franceschi, L., Pontil, M., and Salzo, S. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pp. 3748–3758. PMLR, 2020.

Grazzi, R., Pontil, M., and Salzo, S. Convergence properties of stochastic hypergradients. In *International Conference on Artificial Intelligence and Statistics*, pp. 3826–3834. PMLR, 2021.

Grazzi, R., Pontil, M., and Salzo, S. Bilevel optimization with a lower-level contraction: Optimal sample complexity without warm-start. *Journal of Machine Learning Research*, 24(167):1–37, 2023.

Griewank, A. and Walther, A. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, volume 105. SIAM, 2008.

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.

Khanduri, P., Tsaknakis, I., Zhang, Y., Liu, J., Liu, S., Zhang, J., and Hong, M. Linearly constrained bilevel optimization: A smoothed implicit gradient approach. In *International Conference on Machine Learning*, pp. 16291–16325. PMLR, 2023.

Lee, K., Maji, S., Ravichandran, A., and Soatto, S. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10657–10665, 2019.

Liu, Y. and Liu, R. Boml: A modularized bilevel optimization library in python for meta learning. In *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–2. IEEE, 2021.

Lorraine, J., Vicol, P., and Duvenaud, D. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1540–1552. PMLR, 2020.

Maclaurin, D., Duvenaud, D., and Adams, R. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, pp. 2113–2122. PMLR, 2015.

Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C., and Roli, F. Towards poisoning of deep learning algorithms with back-gradient optimization. In *ACM Workshop on Artificial Intelligence and Security*, pp. 27–38, 2017.

Ochs, P., Ranftl, R., Brox, T., and Pock, T. Bilevel optimization with nonsmooth lower level problems. In *Scale Space and Variational Methods in Computer Vision: 5th International Conference*, pp. 654–665. Springer, 2015.

Pedregosa, F. Hyperparameter optimization with approximate gradient. In *International Conference on Machine Learning*, pp. 737–746, 2016.

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

Rosasco, L., Villa, S., and Vũ, B. C. A stochastic forward-backward splitting method for solving monotone inclusions in hilbert spaces. *arXiv preprint arXiv:1403.7999*, 2014.

Rosasco, L., Villa, S., and Vũ, B. C. Convergence of stochastic proximal gradient algorithm. *Applied Mathematics & Optimization*, 82:891–917, 2020.

Scholtes, S. *Introduction to Piecewise Differentiable Equations*. Springer Science & Business Media, 2012.

Xiao, H., Biggio, B., Brown, G., Fumera, G., Eckert, C., and Roli, F. Is feature selection secure against training data poisoning? In *International Conference on Machine Learning*, pp. 1689–1698. PMLR, 2015.

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.

Appendices

This supplementary material is organized as follows. In App. A we recall the notion of definable mappings. App. B gives some auxiliary results and proof of lemmas in the main body. In App. C we present the proof of Theorem 4.1. App. D gives the proof of Theorems 5.5 and 5.6. In App. E we address bilevel optimization. Finally, App. F contains more information on the numerical experiments.

A. Definable Mappings

The concept of definable sets and functions is part of the so called tame geometry. Here we give just a very brief account (additional details can be found in (Bolte & Pauwels, 2021)). An *o-minimal structure on* $(\mathbb{R}, +, \cdot)$ ('o' stands for 'ordinal') is a collection of sets $\mathcal{O} = (\mathcal{O}_p)_{p\in\mathbb{N}}$ such that, for each $p \in \mathbb{N}$,

(i) $\mathcal{O}_p$ is a *Boolean algebra*, meaning a nonempty family of subset of $\mathbb{R}^p$ which is stable by complementations and finite unions and intersections. Moreover, it contains the algebraic sets, that is, the sets of zeros of polynomial functions in p variables.

(ii) $\mathcal{O}_1$ is made exactly of finite unions of intervals.

(iii) $A \in \mathcal{O}_p \Rightarrow\; A \times \mathbb{R}, \mathbb{R} \times A \in \mathcal{O}_{p+1}$

(iv) if $\pi_p\colon \mathbb{R}^{p+1} \to \mathbb{R}^p$ is the canonical projection onto the first p components, then $A \in \mathcal{O}_{p+1} \;\Rightarrow\; \pi_p(A) \in \mathcal{O}_p$;

Subsets of $\mathbb{R}^p$ which belongs to an o-minimal structure $\mathcal{O}$ are called *definable in* $\mathcal{O}$ and set-valued mappings $F\colon \mathbb{R}^d \rightrightarrows \mathbb{R}^p$ are said *definable in* $\mathcal{O}$ if their graphs (as a subset of $\mathbb{R}^{d+p}$) is definable in $\mathcal{O}$.

There are several examples of o-minimal structures. The smallest one is that of real semialgebraic sets, meaning finite unions of sets which are solutions of a system of polynomial equations and inequalities. Here we consider the larger class of $\log - \exp$ structure, which additionally contains the graph of the exponential function and includes most of the functions considered in machine learning, including deep learning. So, in this paper definable is meant to be definable in the $\log - \exp$ o-minimal structure.

B. Auxiliary Lemmas

Lemma B.1 (Properties of the excess)**.** *Let $\mathcal{A}, \mathcal{B}, \mathcal{A}', \mathcal{B}' \subset \mathbb{R}^{n\times p}$ and $\mathcal{C} \subset \mathbb{R}^{d\times n}$, $\mathcal{D} \subset \mathbb{R}^{p\times d}$ be nonempty sets of matrices. The following hold true:*

(i) $e(\mathcal{A}, \mathcal{C}) \leqslant e(\mathcal{A}, \mathcal{B}) + e(\mathcal{B}, \mathcal{C})$

(ii) $e(\mathcal{A} + \mathcal{A}', \mathcal{B} + \mathcal{B}') \leqslant e(\mathcal{A}, \mathcal{B}) + e(\mathcal{A}', \mathcal{B}')$

(iii) $e(\mathcal{C}\mathcal{A}, \mathcal{C}\mathcal{B}) \leqslant \|\mathcal{C}\|_{\sup} e(\mathcal{A}, \mathcal{B})$ *and* $e(\mathcal{A}\mathcal{D}, \mathcal{B}\mathcal{D}) \leqslant \|\mathcal{D}\|_{\sup} e(\mathcal{A}, \mathcal{B})$

(iv) *If $\mathcal{B} \subset \mathcal{B}'$, then $e(\mathcal{A}, \mathcal{B}') \leqslant e(\mathcal{A}, \mathcal{B})$.*

(v) *Suppose that $n = p$ and that all the elements in $\mathcal{A}$ and $\mathcal{B}$ are invertible. Then*

$$e(\mathcal{A}^{-1}, \mathcal{B}^{-1}) \leqslant \|\mathcal{A}^{-1}\|_{\sup}\|\mathcal{B}^{-1}\|_{\sup}\, e(\mathcal{A}, \mathcal{B}).$$

(vi) *Suppose that $p = p_1 + p_2$ and set, for $k = 1, 2$ $\mathrm{pr}_1\colon \mathbb{R}^{n\times p} \to \mathbb{R}^{n\times p_k}$ be the canonical projections and*

$$\mathcal{A}_k = \mathrm{pr}_k(\mathcal{A}) = \{A_k \in \mathbb{R}^{n\times p_k} \,|\, [A_1, A_2] \in \mathcal{A}\}, \quad \mathcal{B}_k = \mathrm{pr}_k(\mathcal{B}) = \{B_k \in \mathbb{R}^{n\times p_k} \,|\, [B_1, B_2] \in \mathcal{A}\}.$$

Then $e(\mathcal{A}_k, \mathcal{B}_k) \leqslant e(\mathcal{A}, \mathcal{B})$.

(vii) *Suppose that $p = p_1 + p_2$. Then, for all $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^{p_1\times p_2}$, we have*

$$\|\mathcal{A}(\mathcal{X})\|_{\sup} \leqslant \|\mathcal{A}_1\|_{\sup}\|\mathcal{X}\|_{\sup} + \|\mathcal{A}_2\|_{\sup},$$
$$e(\mathcal{A}(\mathcal{X}), \mathcal{A}(\mathcal{Y})) \leqslant \|\mathcal{A}_1\|_{\sup}\, e(\mathcal{X}, \mathcal{Y}), \qquad e(\mathcal{A}(\mathcal{X}), \mathcal{B}(\mathcal{X})) \leqslant (1 + \|\mathcal{X}\|_{\sup})\, e(\mathcal{A}, \mathcal{B})$$

where we recall that $\mathcal{A}(\mathcal{X}) = \{A_1X + A_2 \,|\, [A_1, A_2] \in \mathcal{A}, X \in \mathcal{X}\}$.

Proof. In the following when A is a matrix and $\mathcal{B}$ is a set of matrices we set $d(A,\mathcal{B}) := \inf_{B\in\mathcal{B}}\|A-B\|$, which is the distance from A to the set $\mathcal{B}$.

(i): Let $A \in \mathcal{A}$ and $B \in \mathcal{B}$. Then

$$\begin{aligned}(\forall\, C \in \mathcal{C}) \quad d(A,\mathcal{C}) &\leqslant \|A-C\| \leqslant \|A-B\| + \|B-C\| \\ &\implies d(A,\mathcal{C}) - \|A-B\| \leqslant \|B-C\|.\end{aligned}$$

Thus

$$d(A,\mathcal{C}) - \|A-B\| \leqslant d(B,\mathcal{C}) \leqslant e(\mathcal{B},\mathcal{C})$$

and hence

$$(\forall\, B \in \mathcal{B}) \quad d(A,\mathcal{C}) - e(\mathcal{B},\mathcal{C}) \leqslant \|A-B\|.$$

So, $d(A,\mathcal{C}) - e(\mathcal{B},\mathcal{C}) \leqslant d(A,\mathcal{B}) \leqslant e(\mathcal{A},\mathcal{B}) \implies d(A,\mathcal{C}) \leqslant e(\mathcal{B},\mathcal{C}) + d(\mathcal{A},\mathcal{B})$. Taking the sup in $A \in \mathcal{A}$ the statement follows.

(ii): Let $A \in \mathcal{A}$, $A' \in \mathcal{A}$. Then,

$$\begin{aligned}(\forall\, B \in \mathcal{B})(\forall\, B' \in \mathcal{B}') \quad d(A+A', \mathcal{B}+\mathcal{B}') &\leqslant \|(A+A') - (B+B')\| \\ &\leqslant \|A-B\| + \|A'-B'\|.\end{aligned}$$

Thus,

$$d(A+A', \mathcal{B}+\mathcal{B}') \leqslant d(A,\mathcal{B}) + d(A',\mathcal{B}') \leqslant e(\mathcal{A},\mathcal{B}) + e(\mathcal{A}',\mathcal{B}').$$

Since A and A' are arbitrary in $\mathcal{A}$ and $\mathcal{A}'$ respectively, the statement follows.

(iii): Let $A \in \mathcal{A}$, $B \in \mathcal{B}$ and $C \in C$. Then

$$d(CA, \mathcal{C}\mathcal{B}) \leqslant \|CA - CB\| \leqslant \|C\|\|A-B\| \leqslant \|\mathcal{C}\|_{\sup}\|A-B\|.$$

Taking the infimum over $B \in \mathcal{B}$ we get

$$d(CA, \mathcal{C}\mathcal{B}) \leqslant \|\mathcal{C}\|_{\sup} \inf_{B\in\mathcal{B}}\|A-B\| \leqslant \|\mathcal{C}\|_{\sup}\, e(\mathcal{A},\mathcal{B}).$$

Now, taking the supremum over $C \in \mathcal{C}$ and $A \in \mathcal{A}$, the statement follows. A similar proof can be applied for the other case.

(v): Let $A \in \mathcal{A}$ and $B \in \mathcal{B}$. Then $A^{-1} - B^{-1} = A^{-1}(B-A)B^{-1}$ and hence

$$\|A^{-1} - B^{-1}\| \leqslant \|A^{-1}\|\|A-B\|\|B^{-1}\| \leqslant \|\mathcal{A}^{-1}\|_{\sup}\|\mathcal{B}^{-1}\|_{\sup}\|A-B\|.$$

Thus

$$\begin{aligned}\inf_{B\in\mathcal{B}}\|A^{-1} - B^{-1}\| &\leqslant \|\mathcal{A}^{-1}\|_{\sup}\|\mathcal{B}^{-1}\|_{\sup} \inf_{B\in\mathcal{B}}\|A-B\| \\ &\leqslant \|\mathcal{A}^{-1}\|_{\sup}\|\mathcal{B}^{-1}\|_{\sup}\, e(\mathcal{A},\mathcal{B}).\end{aligned}$$

Taking the supremum in $A \in \mathcal{A}$, the statement follows.

(vi): We first note that if $A = [A_1, A_2] \in \mathbb{R}^{d\times(p_1+p_2)}$ we have

$$\|A_1\| = \sup_{\|x\|\leqslant 1}\|A_1 x\| = \sup_{\|(x,0)\|\leqslant 1}\left\|[A_1 A_2]\begin{bmatrix}x\\0\end{bmatrix}\right\| \leqslant \|A\|$$

and similarly $\|A_2\| \leqslant \|A\|$. Now let $A_1 \in \mathcal{A}_1$ and $B = [B_1 B_2] \in \mathcal{B}$. Then there exists A_2 such that $A = [A_1 A_2] \in \mathcal{A}$ and hence

$$d(A_1, \mathcal{B}_1) \leqslant \|A_1 - B_1\| \leqslant \|A-B\|.$$

Since the above inequality holds for every $B \in \mathcal{B}$ we have

$$d(A_1, \mathcal{B}_1) \leqslant \inf_{B\in\mathcal{B}}\|A-B\| \leqslant e(\mathcal{A},\mathcal{B})$$

which in turns holds for every $A_1 \in \mathcal{A}_1$. Thus, taking the supremum in $A_1 \in \mathcal{A}_1$ the statement follows with $k = 1$. The other case is proved in the same manner.

(vii): For the first inequality we have

$$\begin{aligned}\|\mathcal{A}(\mathcal{X})\|_{\sup} &= \sup_{A\in\mathcal{A}, X\in\mathcal{X}} \|A_1 X + A_2\| \\ &\leqslant \sup_{A\in\mathcal{A}, X\in\mathcal{X}} (\|A_1\|\|X\| + \|A_2\|) \\ &\leqslant \sup_{A\in\mathcal{A}} \|A_1\| \sup_{A\in\mathcal{X}} \|X\| + \sup_{A'\in\mathcal{A}} \|A_2'\| = \|\mathcal{A}_1\|_{\sup}\|\mathcal{X}\|_{\sup} + \|\mathcal{A}_2\|_{\sup}.\end{aligned}$$

For the second inequality we have

$$\begin{aligned}e(\mathcal{A}(\mathcal{X}), \mathcal{A}(\mathcal{Y})) &= \sup_{A\in\mathcal{A}, X\in\mathcal{X}} \inf_{A'\in\mathcal{A}, Y\in\mathcal{Y}} \|A_1 X - A_2 - A_1' Y + A_2'\| \\ &\leqslant \sup_{A\in\mathcal{A}, X\in\mathcal{X}} \inf_{Y\in\mathcal{Y}} \|A_1(X - Y)\| \\ &\leqslant \sup_{A\in\mathcal{A}} \|A_1\| \sup_{X\in\mathcal{X}} \inf_{Y\in\mathcal{Y}} \|X - Y\| = \|\mathcal{A}_1\|_{\sup}\, e(\mathcal{X}, \mathcal{Y}).\end{aligned}$$

For the third inequality we have

$$\begin{aligned}e(\mathcal{A}(\mathcal{X}), \mathcal{B}(\mathcal{X})) &= \sup_{A\in\mathcal{A}, X\in\mathcal{X}} \inf_{B\in\mathcal{B}, X'\in\mathcal{X}} \|A_1 X - A_2 - B_1 X' + B_2\| \\ &\leqslant \sup_{A\in\mathcal{A}, X\in\mathcal{X}} \inf_{B\in\mathcal{B}} \|(A_1 - B_1)X - A_2 + B_2\| \\ &\leqslant \sup_{A\in\mathcal{A}, X\in\mathcal{X}} \inf_{B\in\mathcal{B}} (\|A_1 - B_1\|\|X\| + \|A_2 - B_2\|) \\ &\leqslant \sup_{A\in\mathcal{A}, X\in\mathcal{X}} \inf_{B\in\mathcal{B}} (\|A - B\|\|X\| + \|A - B\|) \\ &\leqslant \sup_{X\in\mathcal{X}} (1 + \|X\|) \sup_{A\in\mathcal{A}} \inf_{B\in\mathcal{B}} \|A - B\| = (1 + \|\mathcal{X}\|_{\sup})\, e(\mathcal{A}, \mathcal{B}).\end{aligned}$$

The proof is complete. □

We now recall the following result from (Bolte et al., 2022) (Lemma 4 in the Appendices), which is stated in a slightly more general form.

Theorem B.2. *Let $F\colon U \subset \mathbb{R}^p \to \mathbb{R}^d$ be a continuous selection of the definable Lipschitz smooth mappings $F_1, \ldots, F_r\colon U \subset \mathbb{R}^p \to \mathbb{R}^d$. Let L_i be the Lipschitz constant of F_i' and set $L = \max_{1\leqslant i\leqslant r} L_i$. Then, for any $x \in U$ there exists $R_x > 0$ such that*

$$\forall\, x' \in U \text{ with } \|x' - x\| \leqslant R_x\colon \quad e(D_F^s(x'), D_F^s(x)) \leqslant L\|x' - x\|.$$

Proof. Similarly to (Bolte et al., 2022) we define

$$g\colon\ ]0, +\infty[\ \rightrightarrows [r] \quad \text{such that } g(\rho) = I_F(B_\rho(x)),$$

where $B_\rho(x)$ is the closed ball of radius $\rho > 0$ centered at x. Now, we note that g is the composition of the maps

$$\varphi\colon\ ]0, +\infty[\ \rightrightarrows \mathbb{R}^p\colon \rho \rightrightarrows B_\rho(x), \quad \text{and} \quad I_F\colon \mathbb{R}^p \rightrightarrows [r].$$

The first one is clearly semialgebraic and hence definable and the second map is definable by definition (since the F_i's are definable, it is easy to see that F is definable if and only if I_F is definable). Thus, being g composition of definable set-valued mappings it is definable. Then, for every $I \subset [r]$, we have that the set $[g = I] = \{\rho \in\ ]0, +\infty[\ \mid g(\rho) = I\}$ is definable and setting $\mathcal{J} = \{g(\rho) \mid \rho \in\ ]0, +\infty[\} \subset 2^{[r]}$, we have that $([g = I])_{I\in\mathcal{J}}$, is a finite partition of $]0, +\infty[$ made of definable sets of the real line. Thus, each one of them must be finite unions of disjoints intervals, which shows that g is piecewise constant. It follows that there exists $R_x > 0$ and $I \subset [r]$ such that for every $\rho \in\ ]0, R_x]$ $g(\rho) = I$. The proof continues as in Lemma 4 in (Bolte et al., 2022). □

Proof of Lemma 2.5. Let $x \in U$. Let $\Delta_r = \{\alpha \in \mathbb{R}^r_+ \mid \sum_{i=1}^r \alpha_i = 1\}$ be the unit simplex of $\mathbb{R}^r$ and $\Delta_r^x = \{\alpha \in \Delta_r \mid \forall\, i \in [r] \backslash I_F(x) \colon \alpha_i = 0\}$ (which is essentially the unit simplex of $\mathbb{R}^{I_F(x)}$). Set $\mathcal{A} = \operatorname{co}(\{\partial F_i(x) \mid i \in I_F(x')\})$. Then, using the property of the excess in Lemma B.1(i)

$$e(D_F^s(x'), D_F^s(x)) \leqslant \underbrace{e(D_F^s(x'), \mathcal{A})}_{(1)} + \underbrace{e(\mathcal{A}, D_F^s(x))}_{(2)}.$$

We will bound the two terms (1) and (2) separately. We recall that

$$D_F^s(x') = \operatorname{co}(\{F_i'(x') \mid i \in I_F(x')\}) \quad \text{and} \quad D_F^s(x) = \operatorname{co}(\{F_i'(x) \mid i \in I_F(x)\}).$$

Then

$$\begin{aligned}
(1) &= \sup_{\alpha \in \Delta_r^{x'}} \inf_{\beta \in \Delta_r^{x'}} \Big\| \sum_{i \in I(x')} \alpha_i F_i'(x') - \sum_{i \in I(x')} \beta_i F_i'(x) \Big\| \\
&\leqslant \sup_{\alpha \in \Delta_r^{x'}} \Big\| \sum_{i \in I(x')} \alpha_i (F_i'(x') - F_i'(x)) \Big\| \\
&\leqslant \sup_{\alpha \in \Delta_r^{x'}} \sum_{i \in I(x')} \alpha_i \| F_i'(x') - F_i'(x) \| \leqslant \sup_{\alpha \in \Delta_r^{x'}} \sum_{i \in I(x')} \alpha_i L \|x - x'\| = L\|x - x'\|.
\end{aligned}$$

Moreover,

$$\begin{aligned}
(2) &= \sup_{\alpha \in \Delta_r^{x'}} \inf_{\beta \in \Delta_r^{x}} \Big\| \sum_{i \in I(x')} \alpha_i F_i'(x) - \sum_{i \in I(x)} \beta_i F_i'(x) \Big\| = \sup_{\alpha \in \Delta_r^{x'}} \inf_{\beta \in \Delta_r^{x}} \Big\| \sum_{i=1}^r (\alpha_i - \beta_i) F_i'(x) \Big\| \\
&\leqslant \sup_{\alpha \in \Delta_r} \inf_{\beta \in \Delta_r^{x}} \Big\| \sum_{i=1}^r (\alpha_i - \beta_i) F_i'(x) \Big\| =: (*).
\end{aligned}$$

Now we note that

$$\varphi(\alpha, \beta) = \Big\| \sum_{i=1}^r (\alpha_i - \beta_i) F_i'(x) \Big\| + \iota_{\Delta_r}(\alpha) + \iota_{\Delta_r^x}(\beta)$$

is jointly convex, hence $\alpha \mapsto \inf_\beta \varphi(\alpha, \beta)$ is convex and its maximum is achieved at the vertices of Δ_r. Thus, if we set $e_i = (\delta_j^i)_{1 \leqslant j \leqslant r}$ the canonical basis of $\mathbb{R}^r$, we have

$$\begin{aligned}
(*) &= \max_{1 \leqslant i \leqslant r} \inf_{\beta \in \Delta_r^x} \Big\| \sum_{j=1}^r (\delta_j^i - \beta_j) F_j'(x) \Big\| = \max_{1 \leqslant i \leqslant r} \inf_{\beta \in \Delta_r^x} \Big\| F_i'(x) - \sum_{j=1}^r \beta_j F_j'(x) \Big\| \\
&\leqslant \max_{1 \leqslant i \leqslant r} \inf_{j \in I(x)} \Big\| F_i'(x) - F_j'(x) \Big\| = M_x.
\end{aligned}$$

In the end

$$e(D_F^s(x'), D_F^s(x)) \leqslant M_x + L\|x' - x\|.$$

Now, let $R_x > 0$ be as in Theorem B.2. Then if $\|x' - x\| > R_x$ we have $\|x' - x\|/R_x > 1$ and hence

$$e(D_F^s(x'), D_F^s(x)) \leqslant \frac{M_x}{R_x}\|x' - x\| + L\|x' - x\| = \left(\frac{M_x}{R_x} + L\right)\|x' - x\|,$$

otherwise, if $\|x - x'\| \leqslant R_x$, then by Theorem B.2, we have

$$e(D_F^s(x'), D_F^s(x)) \leqslant L\|x' - x\| \leqslant \left(\frac{M_x}{R_x} + L\right)\|x' - x\|.$$

The statement follows. $\square$

Lemma B.3. *Under Assumption 3.1(ii), for every $(u, \lambda) \in \mathbb{R}^p \times \Lambda$,*

$$\|(I - D_{\Phi,1}(u, \lambda))^{-1}\|_{\sup} \leqslant \frac{1}{1-q}, \qquad \|D_w^{\mathrm{imp}}(\lambda)\|_{\sup} \leqslant \|D_w^{\mathrm{fix}}(\lambda)\|_{\sup} \leqslant \frac{\|D_{\Phi,2}(w(\lambda), \lambda)\|_{\sup}}{1-q}.$$

Proof. As for the first inequality, we recall that for any matrix A such that $\|A\| \leqslant q < 1$, we have $(I - A)^{-1} = \sum_{n=0}^{\infty} A^n$ and hence $\|I - A\| \leqslant \sum_{n=0}^{+\infty} \|A\|^n \leqslant \sum_{n=0}^{+\infty} q^n = 1/(1-q)$. Thus, if we let $\mathcal{A} = D_\Phi(u, \lambda)$ we have that

$$\|(I - \mathcal{A}_1)^{-1}\|_{\sup} = \sup_{A_1 \in \mathcal{A}_1} \|(I - A_1)^{-1}\| \leqslant \frac{1}{1-q}.$$

The second inequality holds since $D_w^{\mathrm{imp}}(\lambda) \subset D_w^{\mathrm{fix}}(\lambda)$. For the last inequality we note that if we let $\mathcal{B} = D_\Phi(w(\lambda), \lambda)$, it follows from the definition of D_w^{fix} that

$$D_w^{\mathrm{fix}}(\lambda) = \mathcal{B}(D_w^{\mathrm{fix}}(\lambda)).$$

Thus, applying Lemma B.1(vii) and recalling that $\|D_{\Phi,1}(w(\lambda), \lambda)\|_{\sup} \leqslant q < 1$ we have

$$\begin{aligned}
\|D_w^{\mathrm{fix}}(\lambda)\|_{\sup} &\leqslant \|D_{\Phi,1}(w(\lambda), \lambda)\|_{\sup} \|D_w^{\mathrm{fix}}(\lambda)\|_{\sup} + \|D_{\Phi,2}(w(\lambda), \lambda)\|_{\sup} \\
&\leqslant q \|D_w^{\mathrm{fix}}(\lambda)\|_{\sup} + \|D_{\Phi,2}(w(\lambda), \lambda)\|_{\sup}
\end{aligned}$$

which implies the last inequality, after rearranging the terms. □

C. Iterative and Approximate Implicit Differentiation

Note that if $\kappa = 1/(1-q)$, then $q^t = \exp(-\log(1/q)t) \leqslant \exp(-t/\kappa)$.

Proof of Theorem 4.1. Let $\lambda \in \Lambda$ and $t \in \mathbb{N}$, $t \geqslant 1$. For the sake of brevity, we set

$$\begin{aligned}
b_{\lambda,t} &= \big(\|D_w^{\mathrm{fix}}(\lambda)\|_{\sup} + 1\big) C_\lambda(w_t(\lambda)), \\
\mathcal{A}_t &= D_\Phi(w_t(\lambda), \lambda), \quad \mathcal{A}_{t,1} = D_{\Phi,1}(w_t(\lambda), \lambda), \quad \mathcal{B} = D_\Phi(w(\lambda), \lambda),
\end{aligned}$$

where C_λ is defined in Lemma 3.2. We recall that

$$D_{w_t}(\lambda) = \mathcal{A}_{t-1}(D_{w_{t-1}}(\lambda)), \qquad D_w^{\mathrm{fix}}(\lambda) = \mathcal{B}(D_w^{\mathrm{fix}}(\lambda)).$$

We also recall that $\delta_\lambda(t) = \mathbb{1}\{\|w_t(\lambda) - w(\lambda)\| > R_\lambda\} \in \{0, 1\}$ and hence

$$C_\lambda(w_t(\lambda)) = L + \frac{M_\lambda}{R_\lambda} \delta_\lambda(t).$$

ITD (16): Let $\Delta'_t := e(D_{w_t}(\lambda), D_w^{\mathrm{fix}}(\lambda))$. Using the properties in Lemma B.1(i)(vii) we have

$$\begin{aligned}
\Delta'_t &= e(\mathcal{A}_{t-1}(D_{w_{t-1}}(\lambda)), \mathcal{B}(D_w^{\mathrm{fix}}(\lambda))) \\
&\leqslant e(\mathcal{A}_{t-1}(D_{w_{t-1}}(\lambda)), \mathcal{A}_{t-1}(D_w^{\mathrm{fix}}(\lambda))) + e(\mathcal{A}_{t-1}(D_w^{\mathrm{fix}}(\lambda)), \mathcal{B}(D_w^{\mathrm{fix}}(\lambda))) \\
&\leqslant \|\mathcal{A}_{t-1,1}\|_{\sup} \Delta'_{t-1} + (1 + \|D_w^{\mathrm{fix}}(\lambda)\|_{\sup})\, e(\mathcal{A}_{t-1}, \mathcal{B}) \\
&\leqslant q \Delta'_{t-1} + b_{\lambda,t-1} \Delta_{t-1},
\end{aligned}$$

where for the last inequality we used that for any $u \in \mathbb{R}^d$, $\|D_{\Phi,1}(u, \lambda)\| < q$ and Lemma 3.2. By unrolling the recursive inequality and using the inequality $\Delta_i \leqslant q^i \Delta_0$ we obtain

$$\begin{aligned}
\Delta'_t &\leqslant q^t \Delta'_0 + \sum_{i=0}^{t-1} q^{t-1-i} b_{\lambda,i} \Delta_i \leqslant q^t \Delta'_0 + q^{t-1} \Delta_0 \sum_{i=0}^{t-1} b_{\lambda,i} \\
&\leqslant q^t \|D_w^{\mathrm{fix}}(\lambda)\|_{\sup} + \Delta_0 q^{t-1} t (\|D_w^{\mathrm{fix}}(\lambda)\|_{\sup} + 1)(L + M_\lambda R_\lambda^{-1} \bar{\delta}_\lambda(t)),
\end{aligned}$$

where, in the last inequality, we used $\Delta'_0 \leqslant \|D_w^{\mathrm{fix}}(\lambda)\|_{\sup}$ and the definitions of $\bar{\delta}_\lambda(t)$ and $C_\lambda(w_t(\lambda))$. Applying Lemma B.3, factoring out t and using the definition of B_λ gives the final result.

AID-FP (17): In this case we have

$$D^k_{w_t}(\lambda) = \mathcal{A}_t(D^{k-1}_{w_t}(\lambda)).$$

Set $\Delta'_k := e(D^k_{w_t}(\lambda), D_w^{\mathrm{fix}}(\lambda))$. Then using again Lemma B.1(i)(vii) we have

$$\begin{aligned}
\Delta'_k &= e(\mathcal{A}_t(D^{k-1}_{w_t}(\lambda)), \mathcal{B}(D_w^{\mathrm{fix}}(\lambda))) \\
&\leqslant e(\mathcal{A}_t(D^{k-1}_{w_t}(\lambda)), \mathcal{A}_t(D_w^{\mathrm{fix}}(\lambda))) + e(\mathcal{A}_t(D_w^{\mathrm{fix}}(\lambda)), \mathcal{B}(D_w^{\mathrm{fix}}(\lambda))) \\
&\leqslant \|\mathcal{A}_{t,1}\|_{\sup}\Delta'_{k-1} + (1 + \|D_w^{\mathrm{fix}}(\lambda)\|_{\sup})\, e(\mathcal{A}_t, \mathcal{B}) \\
&\leqslant q\Delta'_{k-1} + b_{\lambda,t}\Delta_t,
\end{aligned}$$

where for the last inequality we used Assumption 3.1(ii) and Lemma 3.2. By unrolling the inequality recursion we obtain

$$\Delta'_k \leqslant q^k \Delta'_0 + b_{\lambda,t}\Delta_t \sum_{i=0}^{k-1} q^i = q^k \|D_w^{\mathrm{fix}}(\lambda)\|_{\sup} + b_{\lambda,t}\frac{1-q^k}{1-q}\Delta_t.$$

Applying Lemma B.3 and using the definition of $b_{\lambda,t}$, C_λ and δ_λ gives the final result.

For the final comment, if $w_t(\lambda) = \Phi(w_{t-1}(\lambda), \lambda)$, due to the contraction property of Φ, $\Delta_t = q\Delta_{t-1} < \Delta_{t-1}$ and there exist $\tau_\lambda \in \{0, \dots, t'_\lambda\}$ with $t'_\lambda := \lceil \log(\Delta_0/R_\lambda)/\log(1/q) \rceil$, such that $\|w_{\tau_\lambda}(\lambda) - w(\lambda)\| \leqslant R_\lambda$, and if $\tau_\lambda \neq 0$, $\|w_{\tau_\lambda - 1}(\lambda) - w(\lambda)\| > R_\lambda$. Thus, for every $i \in \mathbb{N}$ $\delta_\lambda(w_i) = \mathbb{1}\{i < \tau_\lambda\}$ and therefore for every t, $\delta_\lambda(w_t) \leqslant \bar{\delta}_\lambda(t) \leqslant 1$. ☐

D. Stochastic Implicit Differentiation

For simplicity let for every $u \in \mathbb{R}^d$, $\lambda \in O_\Lambda$

$$\Psi(u, \lambda) = (T(u, \lambda), \lambda), \qquad D_\Psi(u, \lambda) = \left\{ \begin{bmatrix} C_1 & C_2 \\ 0 & I_m \end{bmatrix} \,\middle|\, [C_1, C_2] \in D_T(u, \lambda) \right\}$$

From Lemma 3 in (Bolte & Pauwels, 2021) and Assumption 5.1(i) it follows that D_Ψ is a conservative derivative of Ψ. Moreover, we can write $\Phi(u, \lambda) = G(\Psi(u, \lambda))$ and thanks to the chain rule of conservative derivatives we have that

$$D_\Phi(u, \lambda) := D_G(\Psi(u, \lambda)) D_\Psi(u, \lambda) = D_G(T(u, \lambda), \lambda) \begin{bmatrix} D_T(u, \lambda) \\ 0 \;\; I_m \end{bmatrix} \tag{24}$$

is a conservative derivative for Φ. Furthermore, if Assumption 5.1(ii) is satisfied, then $\|D_{\Phi,1}(u, \lambda)\|_{\sup} \leqslant q < 1$ and D_w^{fix} and D_w^{imp} in (12) and (11) are well defined and conservative derivatives of w. Similarly, a conservative derivative of $\bar{\Phi}$ can be obtained as

$$D_{\bar{\Phi}}(u, \lambda) := D_G(\bar{T}(u, \lambda), \lambda) \begin{bmatrix} D_{\bar{T}}(u, \lambda) \\ 0 \;\; I_m \end{bmatrix}, \quad \textit{with} \quad D_{\bar{T}}(u, \lambda) = \frac{1}{J} \sum_{j=1}^J D_{\hat{T}_{\xi_j^{(1)}}}(u, \lambda). \tag{25}$$

Note that $\partial_2 \bar{\Phi}(u, \lambda)$ as defined in (21) is an element of $D_{\bar{\Phi},2}$.

The following result is similar to Lemma 3.2 and follows directly from Lemma 2.5. The only difference is that the constants are majorized so to be independent on u. This is done only to simplify the analysis.

Lemma D.1. *Under Assumption 5.1, for every $\lambda \in \Lambda$, there exist $R_{G,\lambda}, R_{T,\lambda} > 0$ such that for every $u \in \mathbb{R}^d$ and*

$$\begin{aligned}
e(D_G(u, \lambda), D_G(T(w(\lambda), \lambda), \lambda) &\leqslant C_{G,\lambda}\|u - T(w(\lambda), \lambda)\| \\
e(D_T(u, \lambda), D_T(w(\lambda), \lambda)) &\leqslant C_{T,\lambda}\|u - w(\lambda)\|,
\end{aligned}$$

where $C_{G,\lambda} := L_G + M_{G,\lambda}/R_{G,\lambda}$, $C_{T,\lambda} := L_T + M_{T,\lambda}/R_{T,\lambda}$, with

$$\begin{aligned}
M_{T,\lambda} &:= \max_{i \in \{1,\dots,r\}} \min_{j \in I_T(w(\lambda),\lambda)} \|T'_i(w(\lambda), \lambda) - T'_j(w(\lambda), \lambda)\| \\
M_{G,\lambda} &:= \max_{i \in \{1,\dots,r\}} \min_{j \in I_G(T(w(\lambda),\lambda),\lambda)} \|G'_i(T(w(\lambda), \lambda), \lambda) - G'_j(T(w(\lambda), \lambda), \lambda)\|
\end{aligned}$$

and L_T, L_G satisfying Assumption 3.1(i).

We now present the proof of Theorem 5.5.

Proof of Theorem 5.5. Set, for the sake of brevity, $z_t = (w_t(\lambda), \lambda)$ and $z = (w(\lambda), \lambda)$. We also set $v_k = v_k(w_t(\lambda), \bar{T}_t(\lambda))$, $\bar{v} = \bar{v}(w_t(\lambda), \bar{T}_t(\lambda))$, and $a_\lambda = \|w(\lambda) - T(z)\|$. From Assumption 5.2 on the variance of $\hat{T}$ and since $T(\cdot, \lambda)$ is 1-Lipschitz we have

$$\begin{aligned} \mathrm{Var}[\hat{T}_\xi(z_t) \mid w_t(\lambda), y] &\leqslant \sigma_1 + \sigma_2 \|w_t(\lambda) \mp w(\lambda) - T(z_t) \pm T(z)\|^2 \\ &\leqslant \sigma_1 + 3\sigma_2(2\Delta_t^2 + a_\lambda^2) =: b_\lambda(\Delta_t^2). \end{aligned} \tag{26}$$

Now, recall that $G'(\bar{T}(z_t), \lambda) = [\partial_1 G(\bar{T}(z_t), \lambda), \partial_2 G(\bar{T}(z_t), \lambda)]$, $T'(z_t) = [\partial_1 T(z_t), \partial_2 T(z_t)]$ and set

$$\begin{aligned} B := [B_1, B_2] &= \underset{B' \in D_G(T(z), \lambda)}{\arg\min} \|G'(\bar{T}(z_t), \lambda) - B'\| \\ C := [C_1, C_2] &= \underset{C' \in D_T(z)}{\arg\min} \|T'(z_t) - C'\| \end{aligned}$$

with $B_1, C_1 \in \mathbb{R}^{d\times d}$, $B_2, C_2 \in \mathbb{R}^{d\times m}$, which is valid since the $\arg\min$ is over compact convex sets. Then, recalling the definition of excess and applying Lemma D.1 we have that for $j = 1, 2$

$$\begin{aligned} \|\partial_j G(\bar{T}(z_t), \lambda) - B_j\| &\leqslant \|G'(\bar{T}(z_t), \lambda) - B\| = e(G'(\bar{T}(z_t), \lambda), D_G(T(z), \lambda)) \leqslant C_{G,\lambda}\|\bar{T}(z_t) - T(z)\|, \\ \|\partial_j T(z_t) - C_j\| &\leqslant \|T'(z_t) - C\| = e(T'(z_t), D_G(z)) \leqslant C_{G,\lambda}\|z_t - z\|. \end{aligned} \tag{27}$$

Let also $A := [A_1, A_2]$ with $A_1 := B_1 C_1$ and $A_2 := B_1 C_2 + B_2$. Since $B \in D_G(T(z), \lambda)$, $C \in D_T(z)$ we have that $A \in D_\Phi(z)$ (from the definition of D_Φ in (24)) and consequently that $(I - A_1)^{-1} A_2 \in D_w^{\mathrm{imp}}(\lambda)$. Hence, recalling the definition of excess we can write

$$e(\partial_2 \bar{\Phi}(z_t)^\top v_k, D_w^{\mathrm{imp}}(\lambda)^\top y) \leqslant \|\partial_2 \bar{\Phi}(z_t)^\top v_k - A_2^\top (I - A_1)^{-\top} y\|.$$

To prove the result, it is therefore sufficient to appropriately control the distance to a particular element of $D_w^{\mathrm{imp}}(\lambda)^\top y$, namely $A_2^\top (I - A_1)^{-\top} y$, which is a random variable depending on $y, w_t(\lambda), \boldsymbol{\xi}^{(1)}$ (from the definition of B and C). We have the following error decomposition

$$\begin{aligned} e(\partial_2 \bar{\Phi}(z_t)^\top v_k, D_w^{\mathrm{imp}}(\lambda)^\top y) &\leqslant \|\partial_2 \bar{\Phi}(z_t)^\top v_k - A_2^\top v_k\| + \|A_2^\top v_k - A_2^\top (I - A_1^\top)^{-1} y\|) \\ &\leqslant \|\partial_2 \bar{\Phi}(z_t) - A_2\| \|v_k\| + \|D_{\Phi,2}(z)\|_{\sup} \|v_k - (I - A_1^\top)^{-1} y\|, \end{aligned}$$

where we used that $A_2 \in D_{\Phi,2}(z)$. Hence, squaring and taking the conditional expectation of both sides yields

$$\begin{aligned} \mathbb{E}[e(\partial_2 \bar{\Phi}(z_t)^\top v_k, D_w^{\mathrm{imp}}(\lambda)^\top y)^2] \mid w_t(\lambda), y, \boldsymbol{\xi}^{(1)}] &\leqslant \underbrace{2\mathbb{E}[\|v_k\|^2 \mid w_t(\lambda), y, \boldsymbol{\xi}^{(1)}] \cdot \|\partial_2 \bar{\Phi}(z_t) - A_2\|^2}_{(1)} \\ &\quad + \underbrace{2\|D_{\Phi,2}(z)\|_{\sup}^2 \mathbb{E}[\|v_k - (I - A_1^\top)^{-1} y\|^2 \mid w_t(\lambda), y, \boldsymbol{\xi}^{(1)}]}_{(2)}. \end{aligned} \tag{28}$$

Bound for term (1) of (28) We have that

$$\begin{aligned} \mathbb{E}[\|v_k\|^2 \mid w_t(\lambda), y, \boldsymbol{\xi}^{(1)}] &\leqslant 2\mathbb{E}[\|v_k - \bar{v}\|^2 + \|\bar{v}\|^2 \mid w_t(\lambda), y, \boldsymbol{\xi}^{(1)}] \\ &\leqslant 2\big(\mathbb{E}[\|v_k - \bar{v}\|^2 \mid w_t(\lambda), y, \boldsymbol{\xi}^{(1)}] + \|y\|^2/(1-q)^2\big) \\ &\leqslant 2\|y\|^2\big(\sigma_\lambda(k) + 1/(1-q)^2\big). \end{aligned}$$

where in the second last inequality we used Assumption 5.4(ii). Hence

$$(1) \leqslant 4\|y\|^2\big(\sigma_\lambda(k) + 1/(1-q)^2\big)\|\partial_2 \bar{\Phi}(z_t) - A_2\|^2.$$

Now recall that

$$\partial_2 \bar{\Phi}(z_t) = \partial_1 G(\bar{T}(z_t), \lambda)\partial_2 \bar{T}(z_t) + \partial_2 G(\bar{T}(z_t), \lambda), \quad A_2 = B_1 C_2 + B_2,$$

therefore we have

$$
\begin{aligned}
\|\partial_2\bar{\Phi}(z_t) - A_2\| &\leqslant \|\partial_1 G(\bar{T}(z_t),\lambda)\partial_2\bar{T}(z_t) - \partial_1 G(\bar{T}(z_t),\lambda)C_2\| \\
&\qquad + \|\partial_1 G(\bar{T}(z_t),\lambda)C_2 - B_1C_2\| + \|\partial_2 G(\bar{T}(z_t),\lambda) - B_2\| \\
&\leqslant \|\partial_1 G(\bar{T}(z_t),\lambda)\|\|\partial_2\bar{T}(z_t) - C_2\| \\
&\qquad + \|C_2\|\|\partial_1 G(\bar{T}(z_t),\lambda) - B_1\| + \|\partial_2 G(\bar{T}(z_t),\lambda) - B_2\| \\
&\leqslant \|\partial_2\bar{T}(z_t) - \partial_2 T(z_t)\| + \|\partial_2 T(z_t) - C_2\| \\
&\qquad + \|B_2\|\|\partial_1 G(\bar{T}(z_t),\lambda) - B_1\| + \|\partial_2 G(\bar{T}(z_t),\lambda) - B_2\| \\
&\overset{(*)}{\leqslant} \|\partial_2\bar{T}(z_t) - \partial_2 T(z_t)\| + C_{T,\lambda}\|w_t(\lambda) - w(\lambda)\| \\
&\qquad + C_{G,\lambda}(1 + \|C_2\|)(\|\bar{T}(z_t) - T(z_t)\| + \|T(z_t) - T(z)\|) \\
&\leqslant \|\partial_2\bar{T}(z_t) - \partial_2 T(z_t)\| + [C_{T,\lambda} + C_{G,\lambda}(1 + \|D_{T,2}(z)\|_{\sup})]\Delta_t \\
&\qquad + C_{G,\lambda}(1 + \|D_{T,2}(z)\|_{\sup})\|\bar{T}(z_t) - T(z_t)\|,
\end{aligned}
$$

where in $(*)$ we used (27) and in the last inequality the the fact that $C_2 \in D_{T,2}(z)$. Hence, from Assumption 5.2 and (26), we have

$$
\begin{aligned}
\mathbb{E}\big[\|\partial_2\bar{\Phi}(z_t) - C_2\|^2 \,\big|\, w_t(\lambda), y\big] &\leqslant 3\operatorname{Var}[\partial_2\bar{T}(z_t) \,|\, w_t(\lambda), y] + 3[C_{T,\lambda} + C_{G,\lambda}(1 + \|D_{T,2}(z)\|_{\sup})]^2\Delta_t^2 \\
&\qquad + 3C_{G,\lambda}^2(1 + \|D_{T,2}(z)\|_{\sup})^2 \operatorname{Var}[\bar{T}(z_t) \,|\, w_t(\lambda), y] \\
&\leqslant \frac{3\sigma_2'}{J} + 3[C_{T,\lambda} + C_{G,\lambda}(1 + \|D_{T,2}(z)\|_{\sup})]^2\Delta_t^2 \\
&\qquad + 3C_{G,\lambda}^2(1 + \|D_{T,2}(z)\|_{\sup})^2\frac{b_\lambda(\Delta_t^2)}{J}.
\end{aligned}
$$

In the end we have

$$
\mathbb{E}[(1) \,|\, w_t(\lambda), y] \leqslant 12\|y\|^2\left(\sigma_\lambda(k) + \frac{1}{(1-q)^2}\right)\left(\frac{\sigma_2'}{J} + [C_{T,\lambda} + C_{G,\lambda}M_{T,\lambda}]^2\Delta_t^2 + C_{G,\lambda}^2 M_{T,\lambda}^2\frac{b_\lambda(\Delta_t^2)}{J}\right),
$$

where we set $M_{T,\lambda} = 1 + \|D_{T,2}(w(\lambda),\lambda)\|_{\sup}$.

Bound for term (2) of (28) We have

$$
\|v_k - (I - A_1^\top)^{-1}y\| \leqslant \|v_k - \bar{v}\| + \|\bar{v} - (I - A_1)^{-\top}y\|.
$$

Let $\hat{B}_1 = \partial_1 G(\bar{T}(z_t),\lambda)$ and $\hat{C}_1 = \partial_1 T(z_t)$ and $\hat{A}_1 = \hat{B}_1\hat{C}_1$ and recall that $\bar{v} = (I - \hat{A}_1^\top)^{-1}y$ and $A_1 \in D_{\Phi,1}(z)$. Noting that $\max\{\|B_1\|, \|C_1\|, \|\hat{B}_1\|, \|\hat{C}_1\|)\} \leqslant 1$, $\max\{\|\hat{A}_1\|, \|A_1\|\} \leqslant q$ and hence $\max\{\|(I - \hat{A}_1^\top)^{-1}\|, \|(I - A_1^\top)^{-1}\|\} \leqslant 1/(1-q)$, we obtain

$$
\begin{aligned}
&\|v_k - (I - A_1^\top)^{-1}y\| \\
&\quad \leqslant \|v_k - \bar{v}\| + \|y\|\|(I - \hat{A}_1^\top)^{-1}\|\|(I - A_1^\top)^{-1}\|\|\hat{A}_1 - A_1\| \\
&\quad \leqslant \|v_k - \bar{v}\| + \frac{\|y\|}{(1-q)^2}\|\partial_1 G(\bar{T}(z_t),\lambda)\partial_1 T(z_t) - B_1C_1\| \\
&\quad \leqslant \|v_k - \bar{v}\| + \frac{\|y\|}{(1-q)^2}\Big[\|\partial_1 G(\bar{T}(z_t),\lambda)\partial_1 T(z_t) - B_1\partial_1 T(z_t)\| + \|B_1\partial_1 T(z_t) - B_1C_1)\|\Big] \\
&\quad \leqslant \|v_k - \bar{v}\| + \frac{\|y\|}{(1-q)^2}\Big[\|\partial_1 T(z_t)\|\|\partial_1 G(\bar{T}(z_t),\lambda) - B_1\| + \|B_1\|\|\partial_1 T(z_t) - C_1\|\Big] \\
&\quad \leqslant \|v_k - \bar{v}\| + \frac{\|y\|}{(1-q)^2}\Big[\|\partial_1 G(\bar{T}(z_t),\lambda) - B_1\| + \|\partial_1 T(z_t) - C_1\|\Big]
\end{aligned}
$$

$$
\begin{aligned}
&\overset{(*)}{\leqslant} \|v_k - \bar{v}\| + \frac{\|y\|}{(1-q)^2}\big[C_{G,\lambda}(\|\bar{T}(z_t) - T(z_t)\| + \|T(z_t) - T(z)\|) + C_{T,\lambda}\|w_t(\lambda) - w(\lambda)\|\big] \\
&\leqslant \|v_k - \bar{v}\| + \frac{\|y\|}{(1-q)^2}\big[C_{G,\lambda}\|\bar{T}(z_t) - T(z_t)\| + (C_{G,\lambda} + C_{T,\lambda})\Delta_t\big],
\end{aligned}
$$

where in $(*)$ we used (27). Therefore,

$$
\begin{aligned}
&\mathbb{E}\big[e(v_k,(I - A_1^\top)^{-1}y)^2 \,\big|\, w_t(\lambda), y, \boldsymbol{\xi}^{(1)}\big] \\
&\qquad \leqslant 3\bigg(\|y\|^2\sigma_\lambda(k) + \frac{\|y\|^2}{(1-q)^4}\big[C_{G,\lambda}^2\|\bar{T}(z_t) - T(z_t)\|^2 + (C_{G,\lambda} + C_{T,\lambda})^2\Delta_t^2\big]\bigg)
\end{aligned}
$$

and hence, taking the expectation over $\boldsymbol{\xi}^{(1)}$ we obtain

$$
\begin{aligned}
&\mathbb{E}\big[e(v_k,(I - A_1^\top)^{-1}y)^2 \,\big|\, w_t(\lambda), y\big] \\
&\qquad \leqslant 3\|y\|^2\bigg(\sigma_\lambda(k) + \frac{1}{(1-q)^4}\big[C_{G,\lambda}^2 \operatorname{Var}[\bar{T}(z_t) \,|\, w_t(\lambda), y] + (C_{G,\lambda} + C_{T,\lambda})^2\Delta_t^2\big]\bigg) \\
&\qquad = 3\|y\|^2\bigg(\sigma_\lambda(k) + \frac{1}{(1-q)^4}\Big(C_{G,\lambda}^2 \frac{\operatorname{Var}[\hat{T}_\xi(z_t) \,|\, w_t(\lambda), y]}{J} + (C_{G,\lambda} + C_{T,\lambda})^2\Delta_t^2\Big)\bigg) \\
&\qquad \leqslant 3\|y\|^2\bigg(\sigma_\lambda(k) + \frac{1}{(1-q)^4}\Big(C_{G,\lambda}^2 \frac{b_\lambda(\Delta_t^2)}{J} + (C_{G,\lambda} + C_{T,\lambda})^2\Delta_t^2\Big)\bigg).
\end{aligned}
$$

In the end we have

$$
\mathbb{E}[(2) \,|\, w_t(\lambda), y] \leqslant 6\|D_{\Phi,2}(w(\lambda),\lambda)\|_{\sup}^2\|y\|^2\bigg(\sigma_\lambda(k) + \frac{1}{(1-q)^4}\Big(C_{G,\lambda}^2 \frac{b_\lambda(\Delta_t^2)}{J} + (C_{G,\lambda} + C_{T,\lambda})^2\Delta_t^2\Big)\bigg).
$$

Combined bound By combining the above results we finally obtain

$$
\begin{aligned}
&\mathbb{E}[e(\partial_2\bar{\Phi}(z_t)^\top v_k, D_w^{\mathrm{imp}}(\lambda)^\top y)^2] \,|\, w_t(\lambda), y] \\
&\leqslant 12\|y\|^2\big(\sigma_\lambda(k) + \kappa^2\big)\bigg(\frac{\sigma_2'}{J} + [C_{T,\lambda} + C_{G,\lambda}M_{T,\lambda}]^2\Delta_t^2 + C_{G,\lambda}^2 M_{T,\lambda}^2 \frac{\sigma_1 + 3\sigma_2(2\Delta_t^2 + a_\lambda^2)}{J}\bigg) \\
&\quad + 6\|y\|^2\|D_{\Phi,2}(w(\lambda),\lambda)\|_{\sup}^2\bigg(\sigma_\lambda(k) + \kappa^4\Big(C_{G,\lambda}^2 \frac{\sigma_1 + 3\sigma_2(2\Delta_t^2 + a_\lambda^2)}{J} + (C_{G,\lambda} + C_{T,\lambda})^2\Delta_t^2\Big)\bigg),
\end{aligned}
$$

where we used the expression for $b_\lambda(\Delta_t^2)$ and $\kappa = (1-q)^{-1}$. Taking the expectation $\mathbb{E}[\cdot \,|\, w_t(\lambda)]$ and recalling the hypothesis on $\|y\|^2$ and Δ_t^2 in Assumption 5.4(i)(iii), the statement follows. □

Before reporting the proof for the linear system rate, we rewrite for reader's convenience the following result from (Grazzi et al., 2021), which establishes a convergence rate for stochastic fixed-point iterations with a decreasing step size.

Lemma D.2. *(Grazzi et al., 2021, Theorem 4.2) Let $\Psi\colon \mathbb{R}^d \to \mathbb{R}^d$ be a q-contraction ($0 \leqslant q < 1$), ξ a random variable with values in Ξ and $\hat{\Psi}\colon \mathbb{R}^d \times \Xi \to \mathbb{R}^d$ be such that for every $v \in \mathbb{R}^d$*

$$
\mathbb{E}[\hat{\Psi}(v,\xi)] = \Psi(v) \quad \textit{and} \quad \operatorname{Var}[\hat{\Psi}(v,\xi)] \leqslant \hat{\sigma}_1 + \hat{\sigma}_2\|\Psi(v) - v\|^2,
$$

for some $\hat{\sigma}_1, \hat{\sigma}_2 > 0$. Let $\eta_i = \beta/(\gamma + i)$, with $\beta > 1/(1-q^2)$ and $\gamma \geqslant \beta(1 + \hat{\sigma}_2)$. Let $(\xi_i)_{i\in\mathbb{N}}$ be a sequence of i.i.d copies of ξ and let $(v_i)_{i\in\mathbb{N}}$ be such that $v_0 = 0$ and for $i = 0, 1, \ldots$

$$
v_{i+1} = v_i + \eta_i(\hat{\Psi}(v_i,\xi_i) - v_i).
$$

Then for every $i \in \mathbb{N}$

$$
\mathbb{E}[\|v_i - \bar{v}\|^2] \leqslant \frac{1}{\gamma + i}\max\Big\{\gamma\|\bar{v}\|^2, \frac{\beta^2\hat{\sigma}_1}{\beta(1-q^2) - 1}\Big\},
$$

where $\bar{v}$ is the (unique) fixed point of Ψ.

Algorithm 3 Stochastic fixed point iterations

1: **Input:** $k \in \mathbb{N}$, $u_1, u_2, y \in \mathbb{R}^d$, $\boldsymbol{\xi}^{(2)} = (\xi_i^{(2)})_{1 \leqslant i \leqslant k}$.
2: $\hat{\Psi} \colon (v, x) \mapsto \partial_1 \hat{T}(u_1, \lambda, x)^\top \partial_1 G(u_2, \lambda)^\top v + y$
3: $v_0 = 0$
4: **for** $i = 1$ **to** k **do**
5: $\quad v_i \leftarrow (1 - \eta_i) v_{i-1} + \eta_i \hat{\Psi}(v_{i-1}, \xi_i^{(2)})$
6: **end for**
7: **Return** v_k

We now present the rate for the algorithm used to solve the linear system in Algorithm 1. Consider the procedure in Algorithm 3

Note that v_k in Algorithm 1 is exactly the output of Algorithm 3 with $u_1 = w_t(\lambda)$, $u_2 = \bar{T}_t(\lambda)$. Moreover, we obtain the following convergence rate which is completely independent from the inputs u_1 and u_2.

Lemma D.3 (Linear system rate)**.** *Under Assumption 5.1 and 5.2, let $\hat{\sigma}_2 = 2\sigma_1'(1-q)^{-2}$, $\hat{\sigma}_1 = \hat{\sigma}_2 \|y\|^2$, and consider the stochastic fixed point iterations in Algorithm 3 with $\eta_i = \beta/(\gamma + i)$, with $\beta > 1/\left(1 - q^2\right)$ and $\gamma \geqslant \beta\,(1 + \hat{\sigma}_2)$. For any $u_2, u_1, y \in \mathbb{R}^d$ let the solution of the linear system be*

$$\bar{v} := (I - \partial_1 T(u_1, \lambda)^\top \partial_1 G(u_2, \lambda)^\top)^{-1} y.$$

Then we have

$$\mathbb{E}[\|v_k - \bar{v}\|^2] \leqslant \frac{\|y\|^2}{\gamma + k} \max\left\{\frac{\gamma}{(1-q)^2}, \frac{\beta^2 \hat{\sigma}_2}{\beta\,(1-q^2) - 1}\right\}. \tag{29}$$

In particular, if we set $\beta = 2/(1-q^2), \gamma = 2(1+\hat{\sigma}_2)/(1-q^2)$, we obtain

$$\mathbb{E}[\|v_k - \bar{v}\|^2] \leqslant \frac{1}{k} \cdot \frac{2\|y\|^2(1 + 4\sigma_1')}{(1-q)^5}.$$

Proof. Let

$$\Psi(v) := \mathbb{E}[\hat{\Psi}(v, \xi)] = \partial_1 T(u_1, \lambda)^\top \partial_1 G(u_2, \lambda)^\top v + y.$$

Since $\|\partial_1 T(u_1, \lambda)^\top \partial_1 G(u_2, \lambda)^\top\| \leqslant q$, Ψ is a q-contraction with fixed point $\bar{v}$. It is immediate to see that

$$\mathrm{Var}[\hat{\Psi}(v, \xi)] \leqslant \mathrm{Var}[\partial_1 \hat{T}_\xi(u_1, \lambda)]\|v\|^2.$$

Moreover, we have

$$\|v\| \leqslant \|v - \Psi(v)\| + \|\Psi(v) - \Psi(0)\| + \|\Psi(0)\| \leqslant \|v - \Psi(v)\| + q\|v\| + \|y\|$$

and hence

$$(1-q)\|v\| \leqslant \|v - \Psi(v)\| + \|y\|, \tag{30}$$

which, recalling Assumption 5.2 on the variance of T_1', ultimately yields

$$\mathrm{Var}[\hat{\Psi}(v, \xi)] \leqslant \frac{2}{(1-q)^2}\,\mathrm{Var}[\partial_1 \hat{T}_\xi(u_1, \lambda)]\big(\|v - \Psi(v)\|^2 + \|y\|^2\big) \leqslant \frac{2\sigma_1'}{(1-q)^2}\|\Psi(v) - v\|^2 + \frac{2\sigma_1'\|y\|^2}{(1-q)^2}.$$

Therefore, the first part of the statement follows from Lemma D.2 and from $\|\bar{v}\| \leqslant \|y\|(1-q)^{-1}$ (a consequence of (30)). The last part follows by (29), the equations

$$\gamma = \frac{2}{1-q^2}\Big(1 + \frac{2\sigma_1'}{(1-q)^2}\Big) \leqslant \frac{2(1 + 2\sigma_1')}{(1-q^2)(1-q)^2} \quad \text{and} \quad \beta^2 \hat{\sigma}_2 = \frac{8\sigma_1'}{(1-q)^2(1-q^2)^2}$$

and the fact that $(1-q^2)^{-1} \leqslant (1-q)^{-1}$ when $q < 1$. ☐

Proof of Theorem 5.6. By applying Lemma D.3 with $u_1 = w_t(\lambda)$ and $u_2 = \bar{T}(w_t(\lambda), \lambda)$ we obtain that Assumption 5.4(ii) (the rate on the mean square error of v_k) is satisfied with $\sigma_\lambda(k) = O(\kappa^5 k^{-1})$. The statement follows by applying Theorem 5.5 and substituting the rates $\rho_\lambda(t)$ and $\sigma_\lambda(k)$. ☐

E. Bilevel Optimization

In this section we consider Problem (22) and we make the following assumption.

Assumption E.1. The map E satisfies Assumption 3.1(i) with constant L_E and corresponding conservative derivative D_E.

Note that similarly to Φ, since E satisfies Assumption E.1, a direct application of Lemma 2.5 to the map E yields

Lemma E.2. *Under Assumption E.1, for every $\lambda \in \Lambda$, there exist $R_{E,\lambda} > 0$ such that for every $u \in \mathbb{R}^d$*

$$e(D_E(u, \lambda), D_E(w(\lambda), \lambda)) \leqslant C_{E,\lambda} \|u - w(\lambda)\|,$$

where $C_{E,\lambda} := L_E + M_{E,\lambda}/R_{E,\lambda}$, with $M_{E,\lambda} := \max_{i \in \{1,\ldots,m\}} \min_{j \in I_E(w(\lambda),\lambda)} \|E_i'(w(\lambda), \lambda) - E_j'(w(\lambda), \lambda)\|$.

E.1. Deterministic Case

Theorem E.3. *Let Assumption 3.1 and E.1 hold. Then for every $\lambda \in \Lambda$ and every $t, k \in \mathbb{N}$ we have that for BAID-FP we get*

$$e(D_{f_t}^k(\lambda), D_f^{\text{fix}}(\lambda)) = O(\kappa e^{-k/\kappa} + \kappa^2 \Delta_t)$$

while if $w_t(\lambda) = \Phi(w_{t-1}(\lambda), \lambda)$, for BITD we get

$$e(D_{f_t}(\lambda), D_f^{\text{fix}}(\lambda)) = O(\kappa t e^{-\kappa/t}).$$

Proof. For simplicity, let $\mathcal{A} = D_E(w(\lambda), \lambda)$, $\mathcal{A}_t = D_E(w_t(\lambda), \lambda)$ and recall that

$$D_{f_t}(\lambda) = \mathcal{A}_t(D_{w_t}(\lambda)), \qquad D_f^{\text{fix}}(\lambda) = \mathcal{A}(D_w^{\text{fix}}(\lambda)).$$

Using the properties of excess in Lemma B.1 we obtain, for BITD:

$$\begin{aligned}
e(D_{f_t}(\lambda), D_f^{\text{fix}}(\lambda)) &\leqslant e(\mathcal{A}_t(D_{w_t}(\lambda)), \mathcal{A}_t(D_w^{\text{fix}}(\lambda))) + e(\mathcal{A}_t(D_w^{\text{fix}}(\lambda)), \mathcal{A}(D_w^{\text{fix}}(\lambda))) \\
&\leqslant \|D_{E,1}(w_t(\lambda), \lambda)\|_{\sup} \, e(D_{w_t}(\lambda), D_w^{\text{fix}}(\lambda)) \\
&\quad + \big(1 + \|D_w^{\text{fix}}(\lambda)\|_{\sup}\big) \, e(D_E(w_t(\lambda), \lambda), D_E(w(\lambda), \lambda)) \\
&\leqslant (\|D_{E,1}(w(\lambda), \lambda)\|_{\sup} + C_{E,\lambda}\Delta_t) \, e(D_{w_t}(\lambda), D_w^{\text{fix}}(\lambda)) \\
&+ \Big(\frac{\|D_{\Phi,2}(w(\lambda), \lambda)\|_{\sup}}{1-q} + 1\Big) C_{E,\lambda}\Delta_t \\
&\leqslant (\|D_{E,1}(w(\lambda), \lambda)\|_{\sup} + C_{E,\lambda}\Delta_0) \times \underbrace{O(\kappa t e^{-t/\kappa})}_{(*)} \\
&+ \Big(\frac{\|D_{\Phi,2}(w(\lambda), \lambda)\|_{\sup}}{1-q} + 1\Big) C_{E,\lambda}\Delta_0 e^{-t/\kappa},
\end{aligned}$$

where we used $\Delta_t \leqslant \Delta_0 e^{-t/\kappa} < \Delta_0$, the ITD bound in Theorem 4.1 and Lemma E.2. A very similar proof can be done for AID-FP by changing the $(*)$ term to $O(\kappa e^{-k/\kappa} + \kappa^2 e^{-t/\kappa})$. ☐

E.2. Stochastic Case

We consider the special case of Problem (22) with

$$E(w, \lambda) = \mathbb{E}[\hat{E}_\zeta(w, \lambda)], \;\; \Phi(w, \lambda) = G(\mathbb{E}[\hat{T}_\xi(w, \lambda)], \lambda).$$

In addition to Assumption E.4, as for the smooth case in (Grazzi et al., 2023), we consider the following assumption on E

Assumption E.4. For any $\lambda \in \Lambda$ there exists $B_{E,\lambda} \geqslant 0$ such that

$$\forall\, w \in \mathbb{R}^d\colon \;\; \|D_{E,1}(w, \lambda)\|_{\sup} \leqslant B_{E,\lambda}.$$

The assumption above is verified e.g., for the logistic and for the cross-entropy loss. Moreover, the assumptions on $\hat{E}$ are the following.

Assumption E.5. ζ is a random variable with values in $\mathcal{Z}$ and for every $z \in \mathcal{Z}$

(i) $\hat{E}_z\colon \mathbb{R}^d \times O_\Lambda \to \mathbb{R}^d$, $\mathbb{E}[\hat{E}_\zeta(u,\lambda)] = E(u,\lambda)$.

(ii) $\hat{E}_z$ is path differentiable with conservative derivative $D_{\hat{E}}$ and E'_z is a selection of $D_{\hat{E}}$ such that $\hat{E}'_z(u,\lambda) = [\partial_1 \hat{E}_z(u,\lambda), \partial_2 \hat{E}_\zeta(u,\lambda)]$ and there exist $\sigma_{E,1}, \sigma_{E,2} \geqslant 0$ such that for every $u \in \mathbb{R}^d$ and $\lambda \in \Lambda$

$$\mathbb{E}\big[E'_\zeta(u,\lambda)\big] = E'(u,\lambda) \in D_E(u,\lambda), \quad \mathrm{Var}[\partial_1 \hat{E}_\zeta(u,\lambda)] \leqslant \sigma_{E,1}, \quad \mathrm{Var}[\partial_2 \hat{E}_\zeta(u,\lambda)] \leqslant \sigma_{E,2}.$$

Theorem E.6. *Let Assumption 5.1, 5.2, E.1, E.4, E.5 hold and let $\kappa = (1-q)^{-1}$. Also, suppose that $\mathbb{E}[\|w_t(\lambda) - w(\lambda)\|] \leqslant \rho_\lambda(t)$, for every $t \in \mathbb{N}$. Then the output $\hat{\nabla} f(\lambda)^\top$ of NSID-Bilevel (Algorithm 2) where NSID uses step sizes $\eta_i = \Theta(i^{-1})$ satisfies*

$$\mathbb{E}[e(\hat{\nabla} f(\lambda)^\top, D_f^{\mathrm{imp}}(\lambda))^2] = O\left(\frac{\kappa^5}{k} + \kappa^4\left(\frac{1}{J_2} + \rho_\lambda(t)\right) + \frac{\kappa^2}{J_1}\right).$$

Furthermore, if $\rho_\lambda(t) = O(\kappa^\alpha t^{-1})$ ($\alpha > 0$), then

$$\mathbb{E}[e(\hat{\nabla} f(\lambda)^\top, D_f^{\mathrm{imp}}(\lambda))^2] = O\big(\kappa^2 J_1^{-1} + \kappa^5 (k^{-1} + J_2^{-1} + \kappa^\alpha t^{-1})\big).$$

Therefore, by setting e.g., $t = k = J_1 = J_2$ we have

$$\mathbb{E}[e(\hat{\nabla} f(\lambda)^\top, D_f^{\mathrm{imp}}(\lambda))^2] = O(\kappa^{5+\alpha} t^{-1})$$

which has the same dependency on t as stochastic gradient descent on strongly convex and Lipschitz smooth objectives (Bottou et al., 2018).

Proof. For simplicity, let $z_t = (w_t(\lambda), \lambda)$, $z = (w(\lambda), \lambda)$, $\mathcal{A} = D_E(w(\lambda), \lambda)$, $\mathcal{B}_t = \{\bar{E}'(z_t)\}$. We also recall that

$$D_f^{\mathrm{imp}}(\lambda) := \mathcal{A}(D_w^{\mathrm{imp}}(\lambda)), \qquad \hat{\nabla} f(\lambda)^\top = r(z_t)^\top + \partial_2 \bar{E}(z_t),$$

with $r(z_t)$ which is an estimator of $D_w^{\mathrm{imp}}(\lambda)^\top \partial_1 \bar{E}(z_t)$. Then, using the properties in Lemma B.1 and noting that $\mathcal{B}_t(D_w^{\mathrm{imp}}(\lambda)) = \partial_1 \bar{E}(z_t) D_w^{\mathrm{imp}}(\lambda) + \partial_2 \bar{E}(z_t)$, we have

$$\begin{aligned}
& e(\hat{\nabla} f(\lambda)^\top, D_f^{\mathrm{imp}}(\lambda)) \\
&\leqslant e(\hat{\nabla} f(\lambda)^\top, \mathcal{B}_t(D_w^{\mathrm{imp}}(\lambda)) + e(\mathcal{B}_t(D_w^{\mathrm{imp}}(\lambda)), \mathcal{A}(D_w^{\mathrm{imp}}(\lambda))) \\
&\leqslant e(r(z_t), D_w^{\mathrm{imp}}(\lambda)^\top\, \partial_1 \bar{E}(z_t)) + (1 + \|D_w^{\mathrm{imp}}(\lambda)\|_{\sup})\, e(\bar{E}'(z_t), D_E(z)) \\
&\leqslant e(r(z_t), D_w^{\mathrm{imp}}(\lambda)^\top\, \partial_1 \bar{E}(z_t)) + (1 + \|D_w^{\mathrm{imp}}(\lambda)\|_{\sup})(\|\bar{E}'(z_t) - E'(z_t)\| + e(E'(z_t), D_E(z))) \\
&\leqslant e(r(z_t), D_w^{\mathrm{imp}}(\lambda)^\top\, \partial_1 \bar{E}(z_t)) + (1 + \|D_w^{\mathrm{imp}}(\lambda)\|_{\sup})(\|\bar{E}'(z_t) - E'(z_t)\| + C_{E,\lambda}\Delta_t)
\end{aligned}$$

Moreover, let $\tilde{\mathbb{E}} = \mathbb{E}[\,\cdot \mid w_t(\lambda)]$, we have that

$$\begin{aligned}
\tilde{\mathbb{E}}[\|\bar{E}'(z_t) - E'(z_t)\|^2] &= \tilde{\mathbb{E}}[\|\partial_1 \bar{E}(z_t) - \partial_1 E(z_t)\|^2] + \tilde{\mathbb{E}}[\|\partial_2 \bar{E}(z_t) - \partial_2 E(z_t)\|^2] \\
&\leqslant \frac{\mathrm{Var}[\partial_1 \hat{E}_\zeta(z_t) \mid w_t(\lambda)] + \mathrm{Var}[\partial_2 \hat{E}_\zeta(z_t) \mid w_t(\lambda)]}{J_1} \leqslant \frac{\sigma_{E,1} + \sigma_{E,2}}{J_1}.
\end{aligned}$$

Hence

$$\begin{aligned}
& \tilde{\mathbb{E}}[\, e(\hat{\nabla} f(\lambda)^\top, D_f^{\mathrm{imp}}(\lambda))^2] \\
&\leqslant 3\left(\tilde{\mathbb{E}}[e(r(z_t), D_w^{\mathrm{imp}}(\lambda)^\top\, \partial_1 \bar{E}(z_t))^2] + (1 + \|D_w^{\mathrm{imp}}(\lambda)\|_{\sup})^2\Big(C_{E,\lambda}^2 \Delta_t^2 + \frac{\sigma_{E,1} + \sigma_{E,2}}{J_1}\Big)\right). \qquad (31)
\end{aligned}$$

We also note that $\|D_w^{\mathrm{imp}}(\lambda)\|_{\sup} \leqslant \|D_{\Phi,2}(w(\lambda),\lambda)\|_{\sup}/(1-q)$ and that

$$\tilde{\mathbb{E}}\big[\|\partial_1 \bar{E}(z_t)\|^2\big] \leqslant 2\tilde{\mathbb{E}}\big[\|\partial_1 \bar{E}(z_t) - \partial_1 E(z_t)\|^2\big] + 2\|D_{E,1}(z_t)\|_{\sup}^2 \leqslant \frac{2\sigma_{E,1}}{J_1} + 2B_{E,\lambda}. \tag{32}$$

Therefore, taking the total expectation in (31) and applying Theorem 5.5 with $y = \partial_1 \bar{E}(z_t)$ we get

$$\begin{aligned}
&\mathbb{E}\big[e(\hat{\nabla} f(\lambda)^\top, D_f^{\mathrm{imp}}(\lambda))^2\big] \\
&\quad \leqslant O\left(\sigma_\lambda(k) + \kappa^4\left(\frac{2\sigma_{E,1}}{J_1} + 2B_{E,\lambda}\right)\left(\frac{1}{J_2} + \rho_\lambda(t)\right)\right) \\
&\qquad + 3(1+\kappa\|D_{\Phi,2}(w(\lambda),\lambda)\|_{\sup})^2\left(C_{E,\lambda}^2\Delta_t^2 + \frac{\sigma_{E,1}+\sigma_{E,2}}{J_1}\right)\Big) \\
&\quad = O\left(\sigma_\lambda(k) + \kappa^4\left(\frac{1}{J_2} + \rho_\lambda(t)\right) + \frac{\kappa^2}{J_1}\right).
\end{aligned}$$

The first part of the statement follows by noting that for NSID we have $\sigma_\lambda(k) = O(\kappa^5/k)$, where $\kappa = 1/(1-q)$. The second and last result are immediate. □

F. Experimental Details

We give more information on the numerical experiments in Section 7.

F.1. Computing the approximation Error.

Let $c \in \mathbb{R}^m$, be the output of an algorithm approximating the jacobian vector product $D_w^{\mathrm{fix}}(\lambda)^\top y$. We call approximation error the quantity

$$e(c, D_w^{\mathrm{fix}}(\lambda)^\top y).$$

Since $D_w^{\mathrm{fix}}(\lambda)^\top y$ is set valued and each element is not available in closed form, we instead approximate an upper bound to this quantity using AID-FP for enough iterations k, which as we mention in Section 4, generates a subsequence linearly converging to an element of $D_w^{\mathrm{fix}}(\lambda)^\top y$. Also, as a starting point to AID-FP we use $w_t(\lambda) = \Phi(w_{t-1}(\lambda), \lambda)$, with sufficiently large t and starting from $w_0(\lambda) = 0$, so to be sufficiently close to the fixed point solution $w(\lambda)$, also not available in closed form.

F.2. Constructing the fixed-point map.

In all the experiments, we consider composite minimization problems in the form

$$\min_u f_\lambda(u) + g_\lambda(u).$$

To convert it to fixed point we set a step size $\eta_\lambda > 0$ and set

$$\Phi(u,\lambda) = G(T(u,\lambda),\lambda),$$

with

$$G(u,\lambda) = \mathrm{Prox}_{\eta_\lambda g_\lambda}(u) \qquad T(w,\lambda) = u - \eta_\lambda \nabla f_\lambda(u)$$

In particular, since in our case g_λ is always the an elastic net regularization, Prox is the soft-thresholding and we set

$$\eta_\lambda = \frac{2}{c(L+\mu) + 2\lambda_2},$$

where λ_2 is the L2 regularization parameter and L, μ are the largest and smallest eigenvalues of $n^{-1}X^\top X$, where X is the design matrix of the training set (which contains the corrupted points in the case of data poisoning). We set $c = 1$ for the elastic net experiments with synthetic data, while we set $c = 0.1$ for the poisoning experiments. The choice $c = 1$ yields the optimal theoretical value for the step-size when using the square loss, while we used a difference choice for data poisoning (using the cross-entropy loss) since we found the optimal theoretical value for the step-size too conservative. To set the stochastic step-size schedules we also set the (estimated) contraction constant as $q = \max\{|1 - \eta_\lambda(cL + \lambda_2)|, |1 - \eta_\lambda(c\mu + \lambda_2)|\}$.

F.3. Details for the AID and ITD Experiments

We construct the synthetic dataset by sampling each element of the matrix $X \in \mathbb{R}^{n \times d}$ and the vector w from a normal distribution. Subsequently, we set the non-informative features of w to zero and we compute the vector y as $y = Xw + \epsilon$, where ϵ_i is Gaussian noise with mean 0.1 and unit variance. For this experiment we set $n = 100$ and $p = 100$ of which 30 are informative. We use 200 hold-out examples for the validation set.

F.4. Details for the Stochastic Experiments

We start by noting that for each point in the plots for the stochastic experiments in Figure 2 $w_t(\lambda)$ is fixed as the last iterate of deterministic iterative soft-thresholding, so that the focus lies entirely on the computation of the derivative.

For elastic net, we enhance the setup used for the deterministic methods by sampling the population covariance matrix randomly for the informative features. To do so, we first sample a matrix A_1 from a standard normal, then we normalize all eigenvalues by diving all of them by their maximum obtaining A_2, finally we use the normalized $A_2^\top A_2$ as the covariance matrix of a Gaussian distribution for the informative features. This introduces correlations among the features, thereby increasing the complexity of the problem. We also increase the number of training points from 100 to $10K$ and the number of validation points from 200 to $20K$.

For the data poisoning setup we use the MNIST dataset. We split the MNIST original train set into $30K$ example for training and $30K$ examples for validation. Additionally, we perform a random split of the training set into $X \in \mathbb{R}^{n \times d}$ and $\tilde{X} \in \mathbb{R}^{n' \times p}$, with $p = 784$ representing the number of features for MNIST images. Notably, $n' = 9K$ denotes the number of corrupted examples. It is essential to highlight that $\Gamma \in \mathbb{R}^{n' \times p}$ and $n'p$ is approximately 7 million, posing a significant challenge for derivative estimation using zero-order methods. We set the regularization parameters $\lambda = (0.02, 0.1)$ since with this setup, the final uncorrputed linear model achieves a validation accuracy of around 80% with around 90% of components set to zero.

We note that NSID and SID require to choose the step sizes (η_i), which we found to be difficult, since the theoretical values are often conservative estimates for this problem. We try two policies: constant and decreasing (as $\Theta(1/i)$) step sizes, indicated with "const" and "dec" after the method name respectively. Note that only when the step sizes are decreasing NSID is guaranteed to converge. To simplify the setup, we always have the same η_0 for both constant and decreasing step-size policies. Moreover, we set the step size of SID equal to that of NSID, when they use the same step sizes policy. More speifically, we set $\eta_i = a_1/(a_2 + i)$ for (N)SID dec and $\eta = a_1/a_2$ for (N)SID const, where $a_1 = b_1\beta$ and $a_2 = b_2\beta$, where beta is set to the theoretical value suggested in Lemma D.3 $(2/(1-q^2))$. We tuned a_1, a_2 manually for each setting. In particular we set $a_1 = 0.5, a_2 = 2$ for the synthetic Elastic net experiment and $a_1 = 2, a_2 = 0.01$ for Data poisoning.