Nonsmooth Implicit Differentiation for Machine Learning and Optimization

Jérôme Bolte
Toulouse School of Economics
Univ. Toulouse
Toulouse, France

Tam Le
Toulouse School of Economics
Univ. Toulouse
Toulouse, France

Edouard Pauwels
IRIT, CNRS
Univ. Toulouse
Toulouse, France

Antonio Silveti-Falls
Toulouse School of Economics
Univ. Toulouse
Toulouse, France

Abstract

In view of training increasingly complex learning architectures, we establish a nonsmooth implicit function theorem with an operational calculus. Our result applies to most practical problems (i.e., definable problems) provided that a nonsmooth form of the classical invertibility condition is fulfilled. This approach allows for *formal subdifferentiation*: for instance, replacing derivatives by Clarke Jacobians in the usual differentiation formulas is fully justified for a wide class of nonsmooth problems. Moreover this calculus is entirely compatible with algorithmic differentiation (e.g., backpropagation). We provide several applications such as training deep equilibrium networks, training neural nets with conic optimization layers, or hyperparameter-tuning for nonsmooth Lasso-type models. To show the sharpness of our assumptions, we present numerical experiments showcasing the extremely pathological gradient dynamics one can encounter when applying implicit algorithmic differentiation without any hypothesis.

1 Introduction

Differentiable programming. The recent introduction of deep equilibrium networks [7], the increasing importance of bilevel programming (e.g., hyperparameter optimization) [41] and the ubiquity of differentiable programming (e.g., TensorFlow [1], PyTorch [40], JAX [16]) in modern optimization call for the development of a versatile theory of nonsmooth differentiation. Our focus is on nonsmooth implicit differentiation. There are currently two practices lying at the crossroads of mathematics and computer science: on the one hand the use of the standard smooth implicit function theorem "almost everywhere" [29, 28] and on the other hand the development of algorithmic differentiation tools [2, 3, 51]. The empirical use of the latter in the nonsmooth world has shown surprisingly efficient results [51], but the current theories cannot explain this success. We bridge this gap by providing nonsmooth implicit differentiation results and illustrating their impact on the training of neural networks and hyperparameter optimization.

Backpropagation: a formal differentiation approach. Let us consider z implicitly defined through $F(z(x)) = h(x)$ where F and h have full domain and adequate dimensions. How does autograd apply to evaluating the "derivative" of the implicitly defined function z? Regardless of differentiability or nonsmoothness, and provided that inversion is possible, one commonly uses (or dynamically approximates) this derivative by

$$(\mathrm{backprop}_F(z(x)))^{-1}\,\mathrm{backprop}_h x,$$

Preprint. Under review.

where backprop outputs the result of formal backpropagation, see e.g., [43]. This identity[1] is used to provide efficient training despite the fact that the rules of classical nonsmooth calculus are transgressed [7, 51]. Note that spurious outputs may be created by this approach, but on a negligible set. Consider for example the simple implicit problem $x = f(z(x))$ where $f(z) := \tanh(z) + \mathrm{relu}(\quad z) + z \quad \mathrm{relu}(z)$, whose solution is $z(x) = \tanh x$. Yet

applying the implicit differentiation framework of [7] using JAX library, as presented in [51], provides inconsistency of the derivative at the origin, see Figure 1. As mentioned above, despite these unpredictable outputs, propagating derivatives leads to an undeniable efficiency. But can we parallel these propagation ideas with a simple mathematical counterpart? Is there a rigorous theory backing up *formal (sub)differentiation* or *formal propagation*? The answer is positive and was initiated in [14, 15] through conservative Jacobians (see also [35, 23]).

A mathematical model for propagating derivatives. Conservative calculus models nonsmooth algorithmic differentiation faithfully and allows for a sharp study of training methods in Deep Learning [14, 15]. It involves a new class of derivatives, generalizing Clarke Jacobians [19]. A distinctive feature of conservative calculus is that it is preserved by Jacobian multiplication. Consider for example a feed forward network combining analytic or relu activations and max pooling. A conservative Jacobian for this network can be obtained by using Clarke Jacobians formally as classical Jacobians, regardless of qualification conditions. For instance, Figure 1 depicts a selection in a conservative Jacobian. This approach is general enough to handle spurious points such as in Figure 1 while keeping the essence of the properties one expects from a derivative. It was proved in [14] that backprop, applied to any reasonable program of a function, is a conservative Jacobian for this function; in contrast, backprop cannot be modelled by some subdifferential operator. For instance for the fixed point problem above, given conservative Jacobians J_F and J_h (e.g., Clarke Jacobians) for F and h one obtains a new conservative Jacobian J_z implicitly defined through

$$J_F(z(x))J_z(x) = J_h(x).$$

This property exactly parallels the idea of "propagating derivatives" in practice. It gives a strong meaning to the formal use of Jacobians proposed in [7], and many empirical approaches [30, 2, 29, 28].

Main contributions:

— We establish a nonsmooth conservative implicit function theorem that comes with an *implicit calculus* which is the central focus of this paper. Our calculus amounts somehow to *formal subdifferentiation with Clarke Jacobians*. This approach cannot rely on classical tools like the inverse of a Clarke Jacobian or a composition of Clarke Jacobians, which are not in general Clarke Jacobians. Indeed, a surprising example (Example 1) shows that an "inverse function theorem with Clarke calculus" is not possible.

— We study a wide range of applications of our implicit differentiation theorem, covering deep equilibrium problems [7], conic optimization layers [2], and hyperparameter optimization for the Lasso [9]. Each case is detailed and its specificities are discussed.

— As a consequence, we obtain convergence guarantees for mini-batched stochastic algorithms with vanishing step size for training wide classes of Neural Nets, or for Lasso hyperparameter selection. The assumptions needed for our results are mild and fulfilled by most losses occurring in ML in the spirit of [15, 34]: elementary log-exp functions [15], semialgebraic functions [12], all being subclasses of definable functions [22, 47]. The use of such structural classes has become standard in nonsmooth optimization and is more and more common in ML (see, e.g., [18, 15, 34, 32]).

— As in the smooth implicit function theorem, the invertibility condition is not avoidable in general. We provide various examples for which the assumption is not satisfied; this results in severe failures for the corresponding gradient methods. In Figure 1, one sees how lack of invertibility on an otherwise ordinary problem may provide totally unpredictable behavior for smooth quadratic optimization.

Definitions and Notations. A function $F : \mathrm{R}^n \; ! \; \mathrm{R}^m$ is *locally Lipschitz* if, for each $x \; 2 \; \mathrm{R}^n$, there exists a neighborhood U of x such that F is Lipschitz on U. Given matrices $A \; 2 \; \mathrm{R}^{n\times m}$ and

[1] The notation $\mathrm{backprop}_z$ instead of $\mathrm{backprop}(z)$ is indicative of the fact that backprop is an operator that does not act on functions themselves but rather on the program used to represent them, see [15].

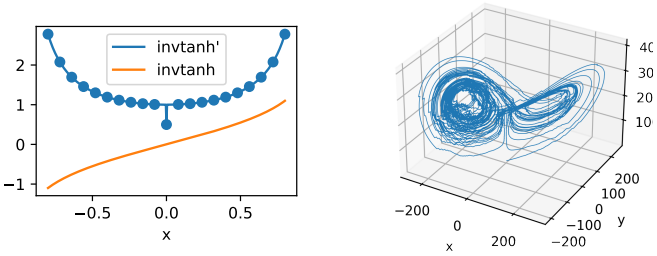

Figure 1: Left: Inconsistencies due to combination of implicit differentiation and algorithmic differentiation. Right: A gradient trajectory of an implicitly defined quadratic function.

B 2 $\mathrm{R}^{n\times p}$, $[A\ B]$ 2 $\mathrm{R}^{n\times(m+p)}$ denotes their concatenation; Id_n denotes the n n identity matrix. For q 2 R^n, $\mathrm{diag}\,(q)$ 2 $\mathrm{R}^{n\times n}$ denotes the diagonal matrix whose diagonal entries are given by the q_i; $\mathrm{sign}\,(q)$ 2 f $1, 0, 1g^n$ denotes the componentwise sign function. The *convex hull* of U is denoted $\mathrm{conv}\ U$. The projection onto a closed convex set C 2 R^n is given, for each x 2 R^n, by $P_C\,(x) := \mathrm{argmin} f\frac{1}{2}\,ku$ $xk^2 : u$ 2 Cg. Given a convex proper lower semicontinuous function f : R^n ! R [$f+1\,g$, we define its proximal operator through x 2 R^n, $\mathrm{prox}_f\,(x) := \mathrm{argmin} f f\,(u) +$ $\frac{1}{2}\,ku$ $xk^2\,u$ 2 $\mathrm{R}^n g$. Set-valued maps are denoted by , for example the subgradient ∂f: R^n R^n. Additional details and notations are provided in Appendix A.

2 Implicit Differentiation with Conservative Jacobians

Definitions and conservativity. Conservative Jacobians are generalized forms of Jacobians well suited for automatic differentiation, introduced in [14]. Given a locally Lipschitz continuous function F : R^n ! R^m, we say that J_F : R^n $\mathrm{R}^{n\times m}$ is a *conservative mapping* or a *conservative Jacobian* for F if J_F has a closed graph, is locally bounded, and is nonempty with

$$\frac{\mathrm{d}}{\mathrm{d}t}F(\gamma(t))\ 2\ J_F(\gamma(t))\dot{\gamma}(t) \text{ a.e.} \tag{1}$$

whenever γ is an absolutely continuous curve in R^n. When $m = 1$, the corresponding vectors are called *conservative gradient fields*. Note that when J_F is conservative, so is its pointwise convexified extension $\mathrm{conv}\ J_F$.

A locally Lipschitz function is called *path differentiable* if it has a conservative Jacobian. Recall that the *Clarke Jacobian* is defined as

$$\mathrm{Jac}^c\,F(x) = \mathrm{conv} \lim_{k\to+\infty} \mathrm{Jac}\,F(x_k) : x_k\ 2\ \mathrm{diff}_F, x_k \underset{k\to+\infty}{!} x$$

where diff_F is the full measure set of points where F is differentiable and $\mathrm{Jac}\,F$ is the standard Jacobian of F. Path differentiability is equivalent to having a chain rule as in (1) for the Clarke subdifferential, see [14, 24].

Examples of path differentiable functions and conservative Jacobians. (a) Convex functions and concave functions are path differentiable, see [14]. This implies that their subdifferential in the sense of convex analysis is a conservative field.
(b) The vast class of definable functions are path differentiable [24, 14]. As a result, the Clarke Jacobian of a Lipschitz definable mapping is a conservative Jacobian. Definable functions (see [5, 24, 18, 14] for an optimization context and [22] for a foundational work) encompass semialgebraic functions [12], elementary log-exp selection [14], PAP [34] (restricted to analytic functions with full domain), and many others, see [47] and references therein. This includes networks with common nonlinearities: for example analytic with full domain (e.g., square, exponential, logistic loss, hyperbolic tangent, sigmoid), relu, max pooling, sort, (see Appendix A.2 for more detail).
(c) The backpropagation can be seen as an oracle (in the optimization sense) for a conservative Jacobian. Let P_F be a numerical program for a function F, aggregating elementary functions, for instance, relu, max pooling, affine mappings, polynomials (in general, any definable

function). Then the backpropagation algorithm applied to P_F, which we denote (abusively) by $\mathrm{backprop}\, P_F := \mathrm{backprop}_F$, outputs an element of a conservative Jacobian [14, Theorem 8] which depends on P_F and can be constructed by a closure procedure [15, definition 5]. As described in [15], due to spurious behaviors, $\mathrm{backprop}_F$ is not in general an element of the Clarke Jacobian of F.

The structure of conservative Jacobians. As established in [35] in a semialgebraic context, the discrepancy between conservative gradients and Clarke subdifferentials is somehow negligible. Let us provide a version of that result matching our concerns. We call conservative mappings of the null function *residual* or *residual conservative*. Such a mapping R has the property that $R(x+tv)v = 0$ for almost all t in R and all x, v in $\mathrm{R}^n \quad \mathrm{R}^n$. The following theorem and proposition (partially) extend results from [14] and [35], their proof is given in Appendix B.

Theorem 1 (The Clarke Jacobian is a minimal conservative Jacobian) *Given a nonempty open subset U of* R^n *and* $F : U \quad \mathrm{R}^n \; ! \; \mathrm{R}^m$ *locally Lipschitz, let* J_F *be a convex-valued conservative Jacobian for* F*. Then for almost all* $x \; 2 \; U, J_F(x) = f\mathrm{Jac}\, Fg$ *and, for all* $x \; 2 \; U$, $\mathrm{Jac}^{\,c} F(x)$ $J_F(x)$.

Proposition 1 (Decomposition of conservative fields) *Let* J_F *be a conservative Jacobian for* F*, then there is a residual* R *such that*

$$J_F \quad \mathrm{Jac}^{\,c} F + R.$$

Note that the above may not hold with equality. Consider $F(x) = jxj$ and $J_F(0) = [\;\; 1, 1] \; [\; [2, 3]$, $J_F(x) = \mathrm{sign}\,(x)$ otherwise. One cannot write $J_F = \mathrm{Jac}^{\,c} F + R$ with a residual operator R.

Formal subdifferentiation in a nonsmooth setting. Propagating derivatives within a nonsmooth function finds its justification in the following:

Proposition 2 (Stability by composition, [14]) *Let* $F : \mathrm{R}^n \; ! \; \mathrm{R}^m$ *and* $G : \mathrm{R}^m \; ! \; \mathrm{R}^l$ *be two locally path differentiable functions having respective conservative Jacobians* J_F *and* J_G*. Then* $F \;\; G$ *is path differentiable and the point-to-set matrix-valued* $x \quad J_F(G(x))J_G(x)$ *is conservative.*

A conservative Implicit Function Theorem. There is already a long tradition of nonsmooth implicit function theorems, e.g., [19, 6, 42, 25]. What makes the following theorem useful is that it comes with a qualification-free calculus. The proofs are given in Appendix B.

Theorem 2 (Implicit differentiation) *Let* $F : \mathrm{R}^n \quad \mathrm{R}^m \; ! \; \mathrm{R}^m$ *be path differentiable on* $U \quad V$ $\mathrm{R}^n \quad \mathrm{R}^m$ *an open set and* $G : U \; ! \; V$ *a locally Lipschitz function such that, for each* $x \; 2 \; U$*,*

$$F(x, G(x)) = 0. \tag{2}$$

Furthermore, assume that for each $x \; 2 \; U$*, for each* $[A \; B] \; 2 \; J_F(x, G(x))$*, the matrix* B *is invertible where* J_F *is a conservative Jacobian for* F*. Then,* $G : U \; ! \; V$ *is path differentiable with conservative Jacobian given, for each* $x \; 2 \; U$*, by*

$$J_G : x \qquad B^{-1}A : [A \; B] \; 2 \; J_F(x, G(x)) \quad .$$

Corollary 1 (Path differentiable implicit function theorem) *Let* $F : \mathrm{R}^n \quad \mathrm{R}^m \; ! \; \mathrm{R}^m$ *be path differentiable with conservative Jacobian* J_F*. Let* $(\hat{x}, \hat{y}) \; 2 \; \mathrm{R}^n \quad \mathrm{R}^m$ *be such that* $F(\hat{x}, \hat{y}) = 0$*. Assume that* $J_F(\hat{x}, \hat{y})$ *is convex and that, for each* $[A \; B] \; 2 \; J_F(\hat{x}, \hat{y})$*, the matrix* B *is invertible. Then, there exists an open neighborhood* $U \quad V \quad \mathrm{R}^n \quad \mathrm{R}^m$ *of* $(\hat{x}, \hat{y})$ *and a path differentiable function* $G : U \; ! \; V$ *such that the conclusion of Theorem 2 holds.*

Corollary 2 (Path differentiable inverse function theorem) *Let* U *and* V *be open neighborhoods of* 0 *in* R^n *and* $\Phi : U \; ! \; V$ *path differentiable with* $\Phi(0) = 0$*. Assume that* Φ *has a conservative Jacobian* J_Φ *such that* $J_\Phi(0)$ *contains only invertible matrices. Then, locally,* Φ *has a path differentiable inverse* Ψ *with a conservative Jacobian given by*

$$J_\Psi(y) = \quad A^{-1} : A \; 2 \; J_\Phi(\Psi(y)) \quad .$$

Remark 1 (a) **(On the necessity of conservativity)** Example 1 in Appendix B shows that one cannot hope for the formulas in Corollaries 1 & 2 to provide Clarke Jacobians in general, even if the input(s) are Clarke Jacobians themselves.
(b) (**Lipschitz definable implicit and inverse function theorems)** See Theorem 4 and 5 in the appendix

3 Nonsmooth implicit differentiation in Machine Learning

Detailed proof arguments for all considered models are given in Appendix C.

Monotone deep equilibrium networks. Deep Equilibrium Networks (DEQs) [7] are specific neural network architectures including layers whose input-output relation is implicitly defined through a fixed point equation of the form

$$z = f(z, x) \tag{3}$$

where $x \ 2 \ \mathrm{R}^p$ is a given input and $z \ 2 \ \mathrm{R}^m$ is the corresponding output. We may consider that the variable x represents both the input layer and layer parameters. Assuming that, for each $x \ 2 \ \mathrm{R}^p$, there is a unique $z \ 2 \ \mathrm{R}^m$ satisfying the relation (3), this defines an input-output relation $z\colon \mathrm{R}^p \ ! \ \mathrm{R}^m$. Furthermore, if f is path differentiable with convex-valued conservative Jacobian $J_f\colon \mathrm{R}^m \quad \mathrm{R}^p \quad \mathrm{R}^{m\times(m+p)}$ whose projection on the first m columns are all invertible, then the function z itself admits a conservative Jacobian which can be computed from Theorem 2.

We now focus on monotone operator implicit layers [50] for which assumptions are easily stated. Our method applies to other similar architectures, e.g., DEQs [7] or implicit graph neural networks [30]. Let $\sigma\colon \mathrm{R}^m \ ! \ \mathrm{R}^m$ be the proximal operator of a convex function and assume σ is path differentiable with conservative Jacobian $J_\sigma\colon \mathrm{R}^m \quad \mathrm{R}^{m\times m}$, assumed to be convex-valued. This encompasses the majority of activation functions used in practice [20]. Let $W \ 2 \ \mathrm{R}^{m\times m}$ be a matrix such that $W + W^T \quad 2\theta I$ with $\theta > 0$. Under these assumptions the implicit equation

$$z = \sigma(Wz + b) \tag{4}$$

has a unique output $z(W, b)$ [50, Theorem 2]. The transformation $(W, b) \ 7! \ z(W, b)$ is a *monotone implicit layer.*

The set-valued mapping obtained from Theorem 2 provides a conservative Jacobian for $(W, z) \ 7! \ z(W, z)$. A similar expression was described in [50, Theorem 2], without using conservativity and using the Clarke Jacobian formally as a classical Jacobian. The proposition below provides a full justification of this heuristic and ensures convergence of algorithmic differentiation based training.

Proposition 3 (Path differentiation through monotone layers) *Assume that J_σ is convex-valued and that, for all $J \ 2 \ J_\sigma(Wz(W, b) + b)$, the matrix $(\mathrm{Id}_m \quad JW)$ is invertible. Consider a loss-like function $\ell\colon \mathrm{R}^m \ ! \ \mathrm{R}$ with conservative gradient $D_\ell\colon \mathrm{R}^m \quad \mathrm{R}^m$, then $g : (W, z) \ 7! \ \ell(z(W, b))$ is path differentiable and has a conservative gradient D_g defined through*

$$D_g\colon (W, b) \quad J^T(\mathrm{Id}_m \quad JW)^{-T}vz^T, J^T(\mathrm{Id}_m \quad JW)^{-T}v) : \ J \ 2 \ J_\sigma(Wz + b), \ v \ 2 \ D_\ell(z) \ .$$

Remark 2 Convexity and invertibility assumptions are satisfied when J_σ is the Clarke Jacobian [50].

Optimization layers: the conic program case. Optimization layers in deep learning may take many forms; we consider here those based on conic programming [17, 3, 2, 4]. We follow [3], simplifying the analysis by ignoring infeasability certificates, which correspond to the absence of a primal-dual solution [17], in line with the implementation described in [2, Appendix B]. Consider a conic problem (P) and its dual (D):

$$\begin{array}{lll} \text{(P)} & \inf & c^T x \\ & \text{subject to} & Ax + s = b \\ & & s \ 2 \ K \end{array} \qquad\qquad \begin{array}{lll} \text{(D)} & \inf & b^T y \\ & \text{subject to} & A^T y + c = 0 \\ & & y \ 2 \ K^*, \end{array} \tag{5}$$

with primal variable $x \in \mathbb{R}^n$, dual variable $y \in \mathbb{R}^m$, and primal slack variable $s \in \mathbb{R}^m$. The set $K \subset \mathbb{R}^m$ is a nonempty closed convex cone and K^* is its dual cone. The problem parameters are the matrix $A \in \mathbb{R}^{m \times n}$ and the vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$; the cone K is fixed. Under the assumption that there is a unique primal-dual solution (x, y, s), we study the path differentiability of the solution mapping as a function of its parameters:

$$(A, b, c) \mapsto \text{sol}(A, b, c) = (x, y, s).$$

For this, let us interpret the solution mapping as a composition mapping involving equation-like implicit formulations. Set $N = n + m$, given $A, b, c \in \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n$, define

$$Q(A, b, c) = \begin{bmatrix} 0 & A^T \\ A & 0 \end{bmatrix} \in \mathbb{R}^{N \times N} \qquad V(b, c) = \begin{bmatrix} c \\ b \end{bmatrix} \in \mathbb{R}^N.$$

Consider a vector $z = (u, v) \in \mathbb{R}^n \times \mathbb{R}^m$, denote by Π the projection onto $\mathbb{R}^n \times K^*$ and define the *residual map* $N : \mathbb{R}^N \times \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^N$ as

$$N(z, A, b, c) = (Q(A, b, c) - \text{Id}_N)\Pi z + V(b, c) + z.$$

The mapping N is a synthetic form of optimality measure for (P) and (D), capturing KKT conditions. To simplify the presentation, we ignore the extreme cases of infeasibility and unboundedness which correspond to an absence of solution in [17].

Define the function $\phi : \mathbb{R}^N \to \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$ through $\phi(u, v) := (u, P_{\mathcal{K}}(v), P_{\mathcal{K}^*}(v) - v)$. As shown in Appendix C.2, $\phi(u, v)$ provides a primal-dual KKT solution of problems (P) and (D) if and only if $N(z, A, b, c) = 0$. When we assume that, for fixed A, b, and c, there is a unique $z \in \mathbb{R}^N$ such that $N(z, A, b, c) = 0$, we have an implicitly defined a function $z = \nu(A, b, c)$, such that

$$\text{sol}(A, b, c) = [\phi \circ \nu](A, b, c). \tag{6}$$

The following result extends the discussion in [17, 3], limited to situations where Π is differentiable at the proposed solution z, to a fully nonsmooth setting; its proof is postponed to Appendix C.2.

Proposition 4 (Path differentiation through cone programming layers) *Assume that $P_{\mathcal{K}}$, N are path differentiable, denote respectively by J_{P_K}, $J_{\mathcal{N}}$ corresponding convex-valued conservative Jacobians. Assume that, for all $A, b, c \in \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n$, $z = \nu(A, b, c) \in \mathbb{R}^n \times \mathbb{R}^m$ is the unique solution to $N(z, A, b, c) = 0$ and that all matrices formed from the N first columns of $J_{\mathcal{N}}(z, A, b, c)$ are invertible. Then, ϕ, ν, and* sol *are path differentiable functions with conservative Jacobians:*

$$\begin{aligned}
J_\nu(A, b, c) &:= -U^{-1}V : [U\ V] \in J_{\mathcal{N}}(\nu(A, b, c), A, b, c), \\
J_\phi(z) &:= \begin{bmatrix} \text{Id}_n & 0 \\ 0 & J_{P_K}(v) \\ 0 & (J_{P_K}(v) - \text{Id}_m) \end{bmatrix}, \\
J_{\text{sol}}(A, b, c) &:= J_\phi(\nu(A, b, c)) J_\nu(A, b, c).
\end{aligned}$$

In practice, the path differentiability of conic projections is pervasive since they are generally semi-algebraic (orthant, second-order cone, PSD cone). See [31, 37, 33, 37] for the computations of the corresponding Clarke Jacobians (which are conservative). Note that a conservative Jacobian for N may be obtained from J_{P_K} using Proposition 2.

Hyperparameter selection for Lasso type problems. Implicit differentiation can be used to tune hyperparameters via first-order methods optimizing some measure of task performance, see [10] and references therein. In a nonsmooth context, we recall the formulation in [9] of the general hyperparameter optimization problem as a bi-level optimization problem:

$$\min_{\lambda \in \mathbb{R}^m} C(\hat{\beta}(\lambda)) \quad \text{such that} \quad \hat{\beta}(\lambda) \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \psi(\beta, \lambda)$$

where $C : \mathbb{R}^p \to \mathbb{R}$ is continuously differentiable (e.g., test loss) and $\psi : \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}$ is a possibly nonsmooth training loss, convex in β, with hyperparameter $\lambda \in \mathbb{R}^m$. We seek a subgradient type

method for this problem with convergence guaranties; our nonsmooth implicit differentiation results can be used for this purpose. We demonstrate this approach on the Lasso problem [45]

$$\hat{\beta}(\lambda) \in \operatorname{argmin} \quad \frac{1}{2}\|y - X\beta\|_2^2 + e^{\lambda}\|\beta\|_1 : \beta \in \mathrm{R}^p \tag{7}$$

where $y \in \mathrm{R}^n$ is the vector of observations, $X = [X_1, \ldots, X_p] \in \mathrm{R}^{n \times p}$ is the design matrix with columns $X_j \in \mathrm{R}^n$, $j \in \{1, \ldots p\}$, and $\lambda \in \mathrm{R}$ is the hyperparameter. Define $F : \mathrm{R} \times \mathrm{R}^p \to \mathrm{R}^p$ to be

$$F(\lambda, \beta) := \beta - \operatorname{prox}_{e^{\lambda}\|\cdot\|_1}(\beta) - X^T(X\beta - y)$$

and recall that, for each $i \in \{1, \ldots, p\}$, $[\operatorname{prox}_{e^{\lambda}\|\cdot\|_1}(\beta)]_i = \operatorname{sign}(\beta_i)\max\{|\beta_i| - e^{\lambda}, 0\}$. The function $F(\lambda, \beta)$ is thus nonsmooth but locally Lipschitz on $\mathrm{R} \times \mathrm{R}^p$. An optimal $\hat{\beta}(\lambda)$ for (7) must satisfy $F(\lambda, \hat{\beta}(\lambda)) = 0$ [21, Prop. 3.1]. For a given solution $\hat{\beta}(\lambda)$, we introduce the equicorrelation set by $E := \{j \in \{1, \ldots, p\} : |X_j^T(y - X\hat{\beta}(\lambda))| = e^{\lambda}\}$ which contains the support set $\operatorname{supp}\hat{\beta} := \{i \in \{1, \ldots, p\} : \hat{\beta}_i \neq 0\}$. In fact, E does not depend on the choice of the solution $\hat{\beta}$, see [46, Lemma 1]. The proof of the following result is given in Appendix C.3.

Proposition 5 (Conservative Jacobian for the solution mapping) *For all $\lambda \in \mathrm{R}$, assume $X_{\mathcal{E}}^T X_{\mathcal{E}}$ is invertible where $X_{\mathcal{E}}$ is the submatrix of X formed by taking the columns indexed by E. Then $\hat{\beta}(\lambda)$ is single-valued, path differentiable with conservative Jacobian, $J_{\hat{\beta}}(\lambda)$, given for all λ as*

$$\left\{ \begin{bmatrix} e^{\lambda}\operatorname{Id}_p & \operatorname{diag}(q)\operatorname{Id}_p \end{bmatrix} \begin{bmatrix} X^T X & \\ & \end{bmatrix}^{-1} \operatorname{diag}(q)\operatorname{sign}\begin{bmatrix} \hat{\beta} \\ X^T(X\hat{\beta} - y) \end{bmatrix} : q \in M(\lambda) \right\}$$

where $M(\lambda) \subset \mathrm{R}^p$ is the set of vectors q such that $q_i = 1$ if $i \in \operatorname{supp}\hat{\beta}$, $q_i = 0$ if $i \notin E$ and $q_i \in [0, 1]$ if $i \in E \setminus \operatorname{supp}\hat{\beta}$.

Taking, in Proposition 5, $q_i = 1$ for all $i \in E$ corresponds to the directional derivative given by LARS algorithm [27], see also [36]. Alternatively, taking $q_i = 0$ for $i \notin \operatorname{supp}\hat{\beta}$ gives the weak derivative described by [9]. Both are particular selections in $J_{\hat{\beta}}$, which is the underlying conservative field.

4 Optimizing implicit problems with gradient descent

We establish the convergence of gradient descent algorithms for compositional learning problems involving implicitly defined functions. The result follows from the previous section and the general convergence results of [15].

The minimization problem. The applications considered in the previous section all yield minimization problems of the type

$$\min_{w \in \mathrm{R}^p} \ell(w) := \frac{1}{N}\sum_{i=1}^{N} \ell_i(w) \text{ with } \ell_i = g_{i,L} \circ g_{i,L-1} \circ \cdots \circ g_{i,1} \tag{8}$$

where, for each $i \in \{1, \ldots, N\}$, $\ell_i : \mathrm{R}^p \to \mathrm{R}$ is a composition of functions having appropriate input and output dimensions. The indices i correspond in practice to learning samples while the loss ℓ embodies an empirical expectation, as for instance in deep learning. We will enforce the following structural condition.

Assumption 1 *For $i \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, L\}$, the function $g_{i,j}$ is locally Lipschitz with conservative Jacobian $J_{i,j}$ and one of the following holds*

- *$g_{i,j}$ and $J_{i,j}$ are semialgebraic (or, more generally, definable).*
- *$g_{i,j}$ is defined as G in Theorem 2, with F and J_F semialgebraic (or, more generally, definable).*

Actually, in Assumption 1 the second point implies the first point; we list both for clarity. More details on semialgebraicity and definability are given in Appendix A.2. Let us stress that virtually all

elements entering the definition of neural networks are semialgebraic or, more generally, definable, see for example [15] for a constructive model. In particular, beyond classical networks with usual nonlinearities (e.g., relu, sigmoid, max pooling ...), this setting encompasses (through Corollary 1):

(a) Deep equilibrium networks: each $g_{i,j}$ may correspond to usual explicit layers or an implicit layer involving a fixed point mapping and a learning sample i as in (4) or (3).

(b) Training with optimization layers: similarly, the inner maps $g_{i,j}$ may also be solution mapping to convex conic programs and related to the sol function (6) of conic problems.

(c) One may assume that $N = 1, L = 2$ and retrieve the hyperparameter tuning for Lasso in its implicit formulation.

SGD with backpropagation. Algorithmic differentiation (AD) is an automated application of the chain rule of differential calculus. When applied to ℓ_i, it amounts to computing one element of the product $J_i := \prod_{j=1}^{L} J_{i,j}$ by choosing one element in each $J_{i,j}$ with appropriate inputs given by intermediate results kept in memory during a forward computation of the composition.

In this context AD stochastic gradient descent requires an initial $w_0 \in \mathrm{R}^p$ and a sequence of *i.i.d.* random indices uniform in $\{1, \dots, N\}$, $(I_k)_{k \in \mathbb{N}}$. It gives:

$$w_{k+1} = w_k - s\alpha_k v_k \tag{9}$$

$$v_k \in J_{I_k}(w_k), \qquad \text{(given by backprop)}, \tag{10}$$

where $(\alpha_k)_{k \in \mathbb{N}}$ is a sequence of positive step sizes and $s \in (s_{\min}, s_{\max})$ is a scaling factor where $s_{\max} > s_{\min} > 0$. A simpler choice could be $v_k \in \partial^c \ell_{I_k}(w_k)$, however, the chain rule used within algorithmic differentiation routines does not produce subgradients (see, e.g., Figure 1). In contrast, conservative Jacobians are faithful models of AD outputs. The asymptotic behavior of the above algorithm depends on the variational properties of the conservative Jacobian $J := \frac{1}{N} \sum_{i=1}^{N} J_i$.

Theorem 3 (Convergence result) *Consider minimizing ℓ given in* (8) *using algorithm* (10) *under Assumption 1. Assume furthermore the following*

- ***Step size:*** $\sum_{k=1}^{+\infty} \alpha_k = +\infty$ *and* $\alpha_k = o(1/\log(k))$.
- ***Boundedness:*** *there exists $M > 0$, and $K \subset \mathrm{R}^p$ open and bounded, such that, for all $s \in (s_{\min}, s_{\max})$ and $w_0 \in \operatorname{cl} K$, $\|w_k\| \leq M$ almost surely.*

For almost all $w_0 \in K$ and $s \in (s_{\min}, s_{\max})$, the objective value $\ell(w_k)$ converges and all accumulation points $\bar{w}$ of w_k are Clarke-critical in the sense that $0 \in \partial^c \ell(\bar{w})$.

This result shows that AD SGD may be applied successfully to all problems described in Section 3, combining algorithmic differentiation with implicit differentiation. Its proof may be adapted directly from [14, 11]; details are given in Appendix D.

5 Numerical experiments

Using implicit differentiation when the invertibility condition in Theorem 2 does not hold can result in absurd training dynamics.

A cyclic gradient dynamics via fixed-point/optimization layer. Consider the bilevel problem:

$$\min_{x,y,s} \quad \ell(x, y, s) := (x - s_1)^2 + 4(y - s_2)^2 \tag{11}$$

$$\text{s.t.} \quad s \in s(x, y) := \arg\max \{(a + b)(-3x + y + 2) : a \in [0, 3], b \in [0, 5]\}.$$

Problem (11) has an equivalent fixed-point formulation using projected gradient descent on the inner problem (Appendix E.1.1). Backpropagation applied to (11) associates to (x, y) the following:

$$\nabla_{(x,y)} \ell(x, y, s(x)) + \tilde{J}_s(x, y)^T \nabla_s \ell(x, y, s(x)) \tag{12}$$

where $\tilde{J}_s$ is piecewise derivative.

We implement gradient descent for (11), evaluating (12) either using `cvxpylayers` [2] or the `JAX` tutorial [51] for fixed-point layers. In both cases, the invertibility condition in Theorem 2 fails when $3x + y + 2 = 0$, resulting in discontinuity of s, affecting the dynamics globally: the gradient trajectory converges to a limit cycle of non critical points (Figure **??**); see Appendix E.1 for details.

Persistence under small perturbations: For different initial points the gradient flow converges to the same limit cycle (Figure 5a). The cycle persists even if we perturb the coefficients in the problem (11) (see for more details Appendix E.1.2) .

(a) (b)

Figure 2: (a) Gradient flows for several initializations. (b) Gradient flows for 20 perturbed experiments with $\sigma^2 = 0.4$.

A Lorenz-like dynamics via implicit differentiation. The Lorenz Ordinary Differential Equation (ODE) writes:

$$\dot{x} = \sigma(y \quad x), \qquad \dot{y} \quad = x(\rho \quad z) \quad y, \quad \text{and} \quad \dot{z} = xy \quad \beta z. \tag{13}$$

It is well-known that taking $(\sigma, \rho, \beta) = (10, 28, 8/3)$, and $(x(0), y(0), z(0)) = (0, 1, 1.05)$ gives a chaotic trajectory, displayed in Figure 3a. Denoting $F : (x, y, z) \mapsto (\sigma(y \quad x), x(\rho \quad z) \quad y, xy \quad \beta z)$ the vector field of the Lorenz system (13), consider the optimization problem:

$$\max_{u \in \mathbb{R}^3} \quad u^T z \qquad \text{s.t.} \qquad z \in \operatorname*{arg\,min}_{s \in \mathbb{R}^3} \|s\| \quad F(u)\|^4 \tag{14}$$

which is obviously equivalent to

$$\max_{u \in \mathbb{R}^3} \quad u^T F(u). \tag{15}$$

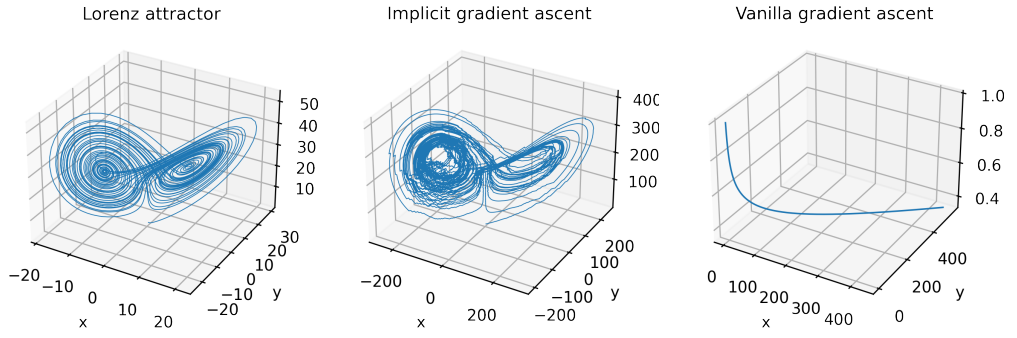

Figure 3: Implicit gradient ascent (b) outputs a pathological curve with some qualitative aspects of the Lorenz dynamics (a) and really different from a classical gradient (c).

The function $g : u \mapsto u^T F(u)$ is a nondegenerate quadratic function whose expression can be found in Appendix E.2.1. The function g has for unique critical point $(0, 0, 0)$ which is a strict saddle-point.

We perform gradient ascent with implicit differentiation using `cvxpylayers` on (14), and the classical gradient ascent on the equivalent problem (15). The path obtained by implicit differentiation (Figure 3b) resembles the Lorenz attractor (Figure 3a), in stark contrast to the conventional method (Figure 3c). The chaotic dynamics are a consequence of the lack of invertibility, due to the power 4 in (14), and various numerical approximations related to optimization and implicit differentiation.

6 Conclusion and future work

This article provides a rigorous framework and calculus rules for nonsmooth implicit differentiation using the theory of conservative Jacobians. In particular, it describes precise conditions under which implicit differentiation can be used, in a way that is compatible with backpropagation and first-order algorithms.

We show the applicability of our results on practical machine learning problems including training of neural networks involving layers with implicitly defined outputs (deep equilibrium nets, networks with optimization layers) and nonsmooth hyperparameter optimization (Lasso-type models).

Finally, we demonstrate the necessity of a rigorous theory of nonsmooth implicit differentiation through multiple numerical experiments. These illustrate the range of extremely pathological gradient dynamics that can occur when algorithmic differentiation is combined with nonsmooth implicit differentiation outside the scope of our theorem, i.e., without satisfying the invertibility condition we specify.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[2] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[3] A. Agrawal, S. Barratt, S. Boyd, E. Busseti, and W. M. Moursi. Differentiating through a cone program. *J. Appl. Numer. Optim*, 1(2):107–115, 2019.

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

[5] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-łojasiewicz inequality. *Mathematics of operations research*, 35(2):438–457, 2010.

[6] J.-P. Aubin and H. Frankowska. On inverse function theorems for set-valued maps. *Journal de mathématiques pures et appliquées*, 66(1):71–89, 1987.

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

[8] M. Benaïm, J. Hofbauer, and S. Sorin. Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, 44(1):328–348, 2005.

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

[10] Q. Bertrand, Q. Klopfenstein, M. Massias, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *arXiv preprint arXiv:2105.01637*, 2021.

[11] P. Bianchi, W. Hachem, and S. Schechtman. Convergence of constant step stochastic gradient descent for non-smooth non-convex functions. *arXiv preprint arXiv:2005.08513*, 2020.

[12] J. Bochnak, M. Coste, and M.-F. Roy. *Real algebraic geometry*, volume 36. Springer Science & Business Media, 2013.

[13] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18(2):556–572, 2007.

[14] J. Bolte and E. Pauwels. Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. *Mathematical Programming*, pages 1–33, 2020.

[15] J. Bolte and E. Pauwels. A mathematical model for automatic differentiation in machine learning. In *Conference on Neural Information Processing Systems*, 2020.

[16] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[17] E. Busseti, W. M. Moursi, and S. Boyd. Solution refinement at regular points of conic problems. *Computational Optimization and Applications*, 74(3):627–643, 2019.

[18] C. Castera, J. Bolte, C. Févotte, and E. Pauwels. An inertial newton algorithm for deep learning. *arXiv preprint arXiv:1905.12278*, 2019.

[19] F. H. Clarke. *Optimization and nonsmooth analysis*. SIAM, 1983.

[20] P. L. Combettes and J.-C. Pesquet. Deep neural network structures solving variational inequalities. *Set-Valued and Variational Analysis*, 28(3):491–518, 2020.

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

[22] M. Coste. *An introduction to o-minimal geometry*. Istituti editoriali e poligrafici internazionali Pisa, 2000.

[23] D. Davis and D. Drusvyatskiy. Conservative and semismooth derivatives are equivalent for semialgebraic maps. *arXiv preprint arXiv:2102.08484*, 2021.

[24] D. Davis, D. Drusvyatskiy, S. Kakade, and J. D. Lee. Stochastic subgradient method converges on tame functions. *Foundations of computational mathematics*, 20(1):119–154, 2020.

[25] A. L. Dontchev and R. T. Rockafellar. *Implicit functions and solution mappings*, volume 543. Springer, 2009.

[26] L. Dries and C. Miller. On the real exponential field with restricted analytic functions. *Israel Journal of Mathematics*, 92:427, 1995.

[27] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *Annals of statistics*, 32(2):407–499, 2004.

[28] L. E. Ghaoui, F. Gu, B. Travacca, and A. Askari. Implicit deep learning. *CoRR*, abs/1908.06315, 2019.

[29] S. Gould, R. Hartley, and D. J. Campbell. Deep declarative networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[30] F. Gu, H. Chang, W. Zhu, S. Sojoudi, and L. El Ghaoui. Implicit graph neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[31] S. Hayashi, N. Yamashita, and M. Fukushima. A combined smoothing and regularization method for monotone second-order cone complementarity problems. *SIAM Journal on Optimization*, 15(2):593–615, 2005.

[32] Z. Ji and M. Telgarsky. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[33] L. Kong, L. Tunçel, and N. Xiu. Clarke generalized jacobian of the projection onto symmetric cones. *Set-Valued and Variational Analysis*, 17(2):135–151, 2009.

[34] W. Lee, H. Yu, X. Rival, and H. Yang. On correctness of automatic differentiation for non-differentiable functions. In *NeurIPS 2020-34th Conference on Neural Information Processing Systems*, 2020.

[35] A. Lewis and T. Tian. The structure of conservative gradient fields. *arXiv preprint arXiv:2101.00699*, 2021.

[36] J. Mairal and B. Yu. Complexity analysis of the lasso regularization path. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1835–1842, 2012.

[37] J. Malick and H. S. Sendov. Clarke generalized jacobian of the projection onto the cone of positive semidefinite matrices. *Set-Valued Analysis*, 14(3):273–293, 2006.

[38] J. J. Moreau. Décomposition orthogonale d'un espace hilbertien selon deux cônes mutuellement polaires. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 255:238–240, 1962.

[39] M. R. Osborne, B. Presnell, and B. A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9(2):319–337, 2000.

[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

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

[42] S. M. Robinson. An implicit-function theorem for a class of nonsmooth functions. *Mathematics of operations research*, 16(2):292–309, 1991.

[43] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536, 1986.

[44] M. Shiota. *Geometry of subanalytic and semialgebraic sets*. De Gruyter, 2011.

[45] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[46] R. J. Tibshirani. The lasso problem and uniqueness. *Electronic Journal of statistics*, 7:1456–1490, 2013.

[47] L. van den Dries and C. Miller. Geometric categories and o-minimal structures. *Duke Math. J*, 84(2):497–540, 1996.

[48] J. Warga. Fat homeomorphisms and unbounded derivate containers. *Journal of Mathematical Analysis and Applications*, 81:545–560, 1981.

[49] A. Wilkie. A theorem of the complement and some new o-minimal structures. *Selecta Mathematica*, 5:397–421, 1999.

[50] E. Winston and J. Z. Kolter. Monotone operator equilibrium networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[51] M. J. Zico Kolter, David Duvenaud. Deep implicit layers - neural odes, deep equilibirum models, and beyond, 2020. NeurIPS Tutorial.

Appendices

A Lexicon

A.1 Conservative fields

We first collect the necessary definitions to define a conservative set-valued field, introduced in [14], and by extension conservative Jacobians. Recall from multivariable calculus that the *Jacobian* of a differentiable function $f\colon \mathrm{R}^n \to \mathrm{R}^m$ is given by

$$\operatorname{Jac} f := \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}.$$

Definition 1 (Absolutely continuous curve) *A continuous function $\gamma : \mathrm{R} \to \mathrm{R}^n$ is an absolutely continuous curve if it has a derivative $\dot{\gamma}(t)$, for almost all $t \in \mathrm{R}$, which furthermore satisfies*

$$\gamma(t) \quad \gamma(0) = \int_0^t \dot{\gamma}(\tau) d\tau$$

for all $t \in \mathrm{R}$.

The *graph* of a set-valued mapping $D : \mathrm{R}^n \rightrightarrows \mathrm{R}^m$ is the set $\operatorname{graph} D := \{(x, z) : x \in \mathrm{R}^n, z \in D(x)\}$.

Definition 2 (Closed graph) *A set-valued mapping $D : \mathrm{R}^n \rightrightarrows \mathrm{R}^m$ has closed graph or is graph closed if $\operatorname{graph} D$ is a closed subset of R^{n+m} or, equivalently, if, for any convergent sequences $(x_k)_{k\in\mathbb{N}}$ and $(z_k)_{k\in\mathbb{N}}$ with $z_k \in D(x_k)$ for all $k \in \mathrm{N}$, it holds*

$$\lim_{k\to\infty} z_k \in D \quad \lim_{k\to\infty} x_k \quad .$$

Definition 3 (Locally bounded) *A set-valued mapping $D : \mathrm{R}^n \rightrightarrows \mathrm{R}^m$ is locally bounded if for all $x \in \mathrm{R}^n$, there exists a neighborhood U of x and $M > 0$ such that, for all $u \in \mathrm{U}$, for all $y \in D(u)$, $\|y\| < M$.*

Definition 4 (Conservative set-valued field) *A set-valued mapping $D : \mathrm{R}^n \rightrightarrows \mathrm{R}^m$ is a conservative field if the following conditions hold:*

1. *For all $x \in \mathrm{R}^n$, $D(x)$ is nonempty.*
2. *D has a closed graph and is locally bounded.*
3. *For any absolutely continuous curve $\gamma : [0, 1] \to \mathrm{R}^n$ with $\gamma(0) = \gamma(1)$,*

$$\int_0^1 \max_{z\in D(\gamma(t))} \langle \dot{\gamma}(t), z\rangle dt = 0.$$

Although conservative fields are not assumed to be locally bounded in [14], we add this restriction here to ensure they are upper semicontinuous. This will allow us to use a nonsmooth Lyapunov method [8] to prove convergence of first-order algorithms.

Definition 5 (Monotone operator) *A set-valued mapping $D : \mathrm{R}^n \quad \mathrm{R}^m$ is called a monotone operator if, for all $x, y \ 2 \ \mathrm{R}^n$, $u \ 2 \ D(x)$, and $v \ 2 \ D(y)$,*

$$hx \quad y, u \quad vi \quad 0.$$

A.2 A simpler and more operational view on definability

We recall basic definitions and results on definable sets and functions used in this work. More details on this theory can be found in [47, 22].

We make a specific attempt to provide a new simple view on this subject by using dictionaries, in the hope that machine learning users consider utilizing these wonderful tools.

The archetypal o-minimal structure is the collection of *semialgebraic* sets. Recall that a set $A \quad \mathrm{R}^n$ is semialgebraic if it can be written as

$$A = \bigcup_{i=1}^{\infty} \bigcap_{j=1}^{\infty} fx \ 2 \ \mathrm{R}^n : P_{ij}(x) < 0, \ Q_{ij}(x) = 0g$$

where, for $i \ 2 \ f1, ..., Ig$ and $j \ 2 \ f1, ..., Jg$, P_{ij} and Q_{ij} are polynomials. The stability properties of semialgebraic sets may be axiomatized [44, 47] to give rise to the general notion of an o-minimal structure:

Definition 6 (o-minimal structure) *Let $O = (O_p)_{p\in\mathrm{N}}$ be a collection of sets such that, for all $p \ 2 \ \mathrm{N}$, O_p is a set of subsets of R^p. O is an o-minimal structure on $(\mathrm{R}, +, \ )$ if it satisfies the following axioms:*

1. *For all $p \ 2 \ \mathrm{N}$, O_p is stable by finite intersection and union, complementation, and contains R^p.*
2. *If $A \ 2 \ O_p$ then $A \quad \mathrm{R}$ and $\mathrm{R} \quad A$ belong to O_{p+1}.*
3. *Denoting by π the projection on the p first coordinates, if $A \ 2 \ O_{p+1}$ then $\pi(A) \ 2 \ O_p$.*
4. *For all $p \ 2 \ \mathrm{N}$, O_p contains the algebraic subsets of R^p, i.e., sets of the form $fx \ 2 \ \mathrm{R}^p : P(x) = 0g$, where $P : \mathrm{R}^p \ ! \ \mathrm{R}$ is a polynomial function.*
5. *The elements of O_1 are exactly the finite unions of intervals.*

A subset $A \quad \mathrm{R}^n$ is said to be definable in an o-minimal structure $O = (O_p)_{p\in\mathrm{N}}$ if O_n contains A. A function $f : \mathrm{R}^n \ ! \ \mathrm{R}^m$ is said to be definable if its graph, a subset of R^{n+m}, is definable.

Note that the collection of semialgebraic sets verifies 3 in Definition 6 according to the Tarski-Seidenberg theorem.

There are several major structures which have been explored [49, 47, 26]. But rather than relying on traditional description of these structures, we provide instead classes of functions that are contained in an o-minimal structure. The goals achieved are twofold:

- The classes we provide are o-minimal and thus all the results provided in the main text apply to functions in these classes.
- It is very easy to verify that a function belongs to one of the classes. Everything boils down to checking that the problem under consideration can be expressed in one of the dictionaries we provide.

Note however that we do not aim at providing neither a comprehensive nor a sharp picture of what could be done with o-minimal structures.

We consider first a collection of functions which will serve to establish dictionaries:

(a) Analytic functions restricted to semialgebraic compact domains (contained in their natural open domain), examples are $\cos$ and $\sin$ restricted to compact intervals.

(b) "Globally subanalytic functions": $\arctan, \tan_{]-\pi/2,\pi/2[}$ or any functions in (a) (see [26] for a precise definition of global subanalyticity).

(c) The $\log$ and $\exp$ functions.

(d) Functions of the form $x \mapsto x^r$ with r a real constant and x a positive real number. These can be represented as $x \mapsto \exp(r\log(x))$ which is definable in $(\mathrm{R}, \exp)$.

(e) Implicitly defined semialgebraic functions. That is, functions $G : \Omega \to \mathrm{R}^m$, with Ω open, which are maximal solutions (i.e., the domain Ω cannot be chosen to be bigger) to nonlinear equations of the type
$$F(x, G(x)) = 0$$
where F is a semialgebraic function.

With this collection of functions we may build *elementary dictionaries*. To demonstrate, we consider the following dictionaries

$$\mathrm{Dic(a)} = \{\text{functions satisfying (a)}\}$$
$$\mathrm{Dic(d, e)} = \{\text{functions satisfying (d) or (e)}\}$$
$$\mathrm{Dic(a, b, c, d, e)} = \{\text{functions satisfying (a) or (b) or (c) or (d) or (e)}\}$$

The last dictionary describes a larger class of functions, we shall come back on this later on.

Consider the dictionary $D = \mathrm{Dic}(\,)$ based on the properties (a)-(e) described above.

Then, in the spirit of [15], we can extend the idea of piecewise selection functions with the following three definitions.

Definition 7 (Elementary D-function) *An elementary D-function is a C^2 function described by a finite compositional expression involving the basic operations $\cdot, +, /$, multiplication by a constant, and the functions of D inside their domain of definition.*

Any elementary D-function is definable in $\mathrm{R}_{\mathrm{an,exp}}$ by stability of definable functions by composition. We shall denote $\mathfrak{S}D$ the set of elementary D-functions. For instance, the following functions belong to $\mathfrak{S}D$:

- $x \mapsto \frac{1}{1+\exp(-x)}$.
- $x \mapsto \log(1 + \exp(x))$.
- $(\beta, \lambda) \mapsto \|X\beta - Y\|_2 + e^{\lambda}\|\beta\|_1$.

Definition 8 (Elementary D-index) *Consider $r \in \mathrm{N}^*$, and $s : \mathrm{R}^n \to \{1, \ldots, r\}$. Then s is said to be an elementary D-index if, for $i \in \{1, \ldots, r\}$, each of the pre-images $s^{-1}(i)$ (i.e., the points in R^n such that s selects the index i) can be written as*
$$\bigcup_{i=1}^{I} \bigcap_{j=1}^{J} \{x \in \mathrm{R}^n : g_{ij}(x) < 0,\ h_{ij}(x) = 0\}$$
where, for $i \in \{1, \ldots, I\}$ and $j \in \{1, \ldots, J\}$, the g_{ij} and h_{ij} are elementary D-functions.

Definition 9 (Piecewise D-function) *A function $f : \mathrm{R}^n \to \mathrm{R}^m$ is a piecewise D-function if there exist $r \in \mathrm{N}^*$, elementary D-functions $f_1, \ldots, f_r$, and an elementary D-index $s : \mathrm{R}^n \to \{1, \ldots, r\}$ such that for all $x \in \mathrm{R}^n$,*
$$f(x) = f_{s(x)}(x).$$

We denote PD the set of piecewise D-functions. With the assumptions we have on the dictionary D, the piecewise selections we consider are all definable (it 's not always the case in general). Notice that piecewise log-exp functions [15] are a specific case of D-functions with the dictionary $D = \mathrm{Dic(c)} = \{\log, \exp\}$. It is easy to see that the following functions are in PD and thus definable:

- $x \mapsto \max(0, x)$ (relu).

– $x \mapsto \max(x_1, ..., x_n)$.
– sort function.
– $x \mapsto \begin{cases} \frac{1}{2}x^2 & \text{for } |x| \leq \delta, \\ \delta(|x| - \frac{1}{2}\delta), & \text{otherwise,} \end{cases}$ with $\delta > 0$ (Huber loss).

Moreover, composition of functions from PD are definable. This allows to say that if $\rho(w, x)$ is the output of a neural network built with usual elementary blocks (for instance Dense, Max Pooling or Conv layers), or even implicit layers involving functions in PD, with input x and weights w, then the empirical risk $\frac{1}{N}\sum_{i=1}^{N} \ell(\rho(w, x_i), y_i)$ is definable with respect to w provided that ℓ is also in PD.

Remark 3 (a) (Small and big dictionaries) It may be puzzling for the reader to see that there is a dictionary that contains all the others. A major comment is in order: bigger is not always better. The bigger the dictionary is, the weaker some properties are. For instance, any piecewise selection $f : \mathrm{R}^n \to \mathrm{R}$ built upon $\mathrm{Dic}(a, b)$ satisfies $|f(x)| \leq c\|x\|^N$ for some $c > 0, N > 0$, which may have consequences in terms of convergence rates, see e.g., [5]. Thus in practice using the smallest dictionary possible may lead to sharper results. On top of this, there are no universal dictionaries [26].
(b) (PAP functions and definability) Recently PAP functions were introduced in order to deal with automatic differentiation matters [34]. To deal with such types of functions in our framework and have guarantees in terms of automatic differentiation, implicit differentiation or convergence properties, we need to view them through the dictionary paradigm. For this we consider the dictionary of analytic functions defined on R^p for some p. In that case, piecewise functions are not necessarily definable but their restrictions to any ball (or any compact semialgebraic subset) are definable.

B Results from Section 2

Theorem 1 (The Clarke Jacobian is a minimal conservative Jacobian) *Given a nonempty open subset U of R^n and $F : U \subset \mathrm{R}^n \to \mathrm{R}^m$ locally Lipschitz, let J_F be a convex-valued conservative Jacobian for F. Then for almost all $x \in U, J_F(x) = \{\mathrm{Jac}\, F\}$ and for all $x \in U$, $\mathrm{Jac}^c F(x) \subset J_F(x)$.*

Proof: Using [14, Lemma 4] for $i \in \{1, \ldots, m\}$, $[J_F]_i$ is a conservative map for F_i on U and it is equal to ∇F_i on a set of full measure $S_i \subset U$. Hence for all $x \in S := \bigcap_{i=1}^{m} S_i$, which is of full measure in U, $J_F(x) = \mathrm{Jac}\, F(x)$. Since S has full measure within U, [48] gives the representation

$$\mathrm{Jac}^c F(x) = \mathrm{conv} \left\{ \lim_{k \to +\infty} \mathrm{Jac}\, F(x_k) : x_k \in S, x_k \xrightarrow[k \to +\infty]{} x \right\}, \text{ for any } x \in U.$$

But since J_F coincides with $\mathrm{Jac}\, F$ throughout S, we have

$$\mathrm{Jac}^c F(x) = \mathrm{conv} \left\{ \lim_{k \to +\infty} J_F(x_k) : x_k \in S, x_k \xrightarrow[k \to +\infty]{} x \right\}$$

for each $x \in U$. Finally, by graph closedness and convexity of J_F we get, for each $x \in U$,

$$\mathrm{Jac}^c F(x) \subset \mathrm{conv} \left\{ J_F \left(\lim_{k \to +\infty} x_k \right) : x_k \in S, x_k \xrightarrow[k \to +\infty]{} x \right\} = J_F(x).$$

Proposition 1 (Decomposition of conservative fields) *Let J_F be a conservative Jacobian for F, then there is a residual R such that*

$$J_F \subset \mathrm{Jac}^c F + R.$$

Proof: We have obviously the inclusion

$$J_F \subset \mathrm{Jac}^c F + (J_F - \mathrm{Jac}^c F),$$

so it suffices to remark that $(J_F - \mathrm{Jac}^c F)$ is residual due to the conservativity properties of both J_F and $\mathrm{Jac}^c F$.

Theorem 2 (Implicit differentiation) *Let $F : \mathrm{R}^n \times \mathrm{R}^m \to \mathrm{R}^m$ be path differentiable on $U \times V \subset \mathrm{R}^n \times \mathrm{R}^m$ an open set and $G : U \to V$ a locally Lipschitz function such that, for each $x \in U$,*

$$F(x, G(x)) = 0. \tag{16}$$

Furthermore, assume that for each $x \in U$, for each $[A\ B] \in J_F(x, G(x))$, the matrix B is invertible where J_F is a conservative Jacobian for F. Then, $G : U \to V$ is path differentiable with conservative Jacobian given, for each $x \in U$, by

$$J_G : x \rightrightarrows B^{-1}A : [A\ B] \in J_F(x, G(x)) \quad .$$

Proof: Let $\gamma : [0, 1] \to U$ be absolutely continuous, then the composition $G \circ \gamma$ is also absolutely continuous since G is locally Lipschitz. By (16) we have, for all $t \in [0, 1]$,

$$F(\gamma(t), G(t))) = 0$$

which we can differentiate almost everywhere; for almost every $t \in [0, 1]$, for any $[A\ B] \in J_F(\gamma(t), G(\gamma(t)))$,

$$[A\ B] \begin{bmatrix} \dot\gamma(t) \\ \frac{\mathrm{d}}{\mathrm{d}t} G(\gamma(t)) \end{bmatrix} = 0 \implies A\dot\gamma(t) = B\frac{\mathrm{d}}{\mathrm{d}t} G(\gamma(t)).$$

Since B is assumed to be invertible, we have, for almost every $t \in [0, 1]$,

$$B^{-1}A\dot\gamma(t) = \frac{\mathrm{d}}{\mathrm{d}t} G(\gamma(t)).$$

The set-valued mapping $J_G : x \rightrightarrows B^{-1}A : [A\ B] \in J_F(x, G(x))$ is nonempty, locally bounded, and has a closed graph for each $x \in U$ since $J_F(x, G(x))$ is a conservative Jacobian and B is invertible . We conclude that G is path differentiable on U with conservative Jacobian J_G.

Corollary 1 (Path differentiable implicit function theorem) *Let $F : \mathrm{R}^n \times \mathrm{R}^m \to \mathrm{R}^m$ be path differentiable with conservative Jacobian J_F. Let $(\hat x, \hat y) \in \mathrm{R}^n \times \mathrm{R}^m$ be such that $F(\hat x, \hat y) = 0$. Assume that $J_F(\hat x, \hat y)$ is convex and that, for each $[A\ B] \in J_F(\hat x, \hat y)$, the matrix B is invertible. Then, there exists an open neighborhood $U \times V \subset \mathrm{R}^n \times \mathrm{R}^m$ of $(\hat x, \hat y)$ and a path differentiable function $G : U \to V$ such that the conclusion of Theorem 2 holds.*

Proof: Since $J_F(\hat x, \hat y)$ is convex, it follows from Theorem 1 that $\mathrm{Jac}^c F(\hat x, \hat y) = J_F(\hat x, \hat y)$ and thus, for any $[A\ B] \in \mathrm{Jac}^c F(\hat x, \hat y)$, B is invertible, i.e., the conditions to apply [19, 7.1 Corollary] to F are satisfied. Therefore there exists an open neighborhood $U_1 \times V_1 \subset \mathrm{R}^n \times \mathrm{R}^m$ of $(\hat x, \hat y)$ and a locally Lipschitz function $G : U_1 \to V_1$ such that, for all $x \in U_1$,

$$F(x, G(x)) = 0.$$

By the continuity of the determinant and the fact that J_F has a closed graph, there exists an open neighborhood $U_2 \times V_2 \subset \mathrm{R}^n \times \mathrm{R}^m$ of $(\hat x, \hat y)$ such that, for all $(x, y) \in U_2 \times V_2$, for all $[A\ B] \in J_F(x, y)$, the matrix B is invertible. Let $U \times V := (U_1 \setminus U_2) \times (V_1 \setminus V_2)$, which is an open neighborhood of $(\hat x, \hat y)$. Then the requirements of Theorem 2 are met for F, J_F, and G on $U \times V$ and the desired claims follow.

Corollary 2 (Path differentiable inverse function theorem) *Let U and V be open neighborhoods of 0 in R^n and $\Phi : U \to V$ path differentiable with $\Phi(0) = 0$. Assume that Φ has a conservative Jacobian J_Φ such that $J_\Phi(0)$ contains only invertible matrices. Then, locally, Φ has a path differentiable inverse Ψ with a conservative Jacobian given by*

$$J_\Psi(y) = \{A^{-1} : A \in J_\Phi(\Psi(y))\} \quad .$$

Proof: Consider the function $F(x, y) = x - \Phi(y)$ and observe that it satisfies the assumptions of Corollary 1, so that we obtain a function G which is exactly the desired inverse.

It is tempting to think that Corollary 2 should come with a formula of the type

$$\mathrm{Jac}^c \Psi(z) = [\mathrm{Jac}^c \Phi(\Psi(z))]^{-1},$$

for all z in a neighborhood of 0. This happens to be false, making the use of the notion of conservativity necessary to catpure the artifacts resulting from application of ordinary calculus rules to nonsmooth inverse functions. Note that since the inverse function theorem is a special case of the implicit function theorem, this also rules out a Clarke calculus for implicit functions.

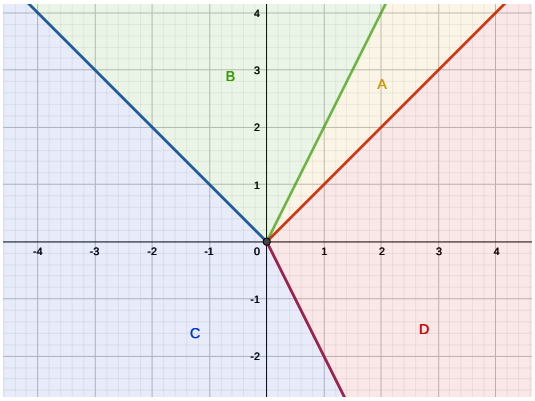

Figure 4: Illustration of the four different sets in the explicit piecewise affine representation of = $^{-1}$.

Example 1 (Counterexample to a potential "Clarke implicit differential calculus") We follow the example given by Clarke [19, Remark 7.1.2]. Consider the mapping $\Phi : \mathrm{R}^2 \; ! \; \mathrm{R}^2$ given by

$$\Phi(x, y) = (jxj + y, 2x + jyj) .$$

It is locally Lipschitz and semialgebraic and thus path differentiable with its Clarke Jacobian a conservative Jacobian. We have the following explicit piecewise linear representation

$$\Phi(x, y) = \begin{cases} (x + y, 2x + y) & \text{if } x \quad 0 \text{ and } y \quad 0, \\ (x + y, 2x \quad y) & \text{if } x \quad 0 \text{ and } y \quad 0, \\ (\; x + y, 2x \quad y) & \text{if } x \quad 0 \text{ and } y \quad 0, \\ (\; x + y, 2x + y) & \text{if } x \quad 0 \text{ and } y \quad 0 \end{cases}$$

from which we deduce that the Clarke Jacobian of Φ has the following structure

$$\mathrm{Jac}^c\, \Phi(0) = \mathrm{conv} \quad \begin{matrix} 1 & 1 \\ 2 & 1 \end{matrix} , \begin{matrix} 1 & 1 \\ 2 & 1 \end{matrix} , \begin{matrix} 1 & 1 \\ 2 & 1 \end{matrix} , \begin{matrix} 1 & 1 \\ 2 & 1 \end{matrix}$$

where the matrices correspond to linear maps in the explicit definition of Φ. Therefore $\mathrm{Jac}^c\, \Phi(0)$ is an affine set whose dimension is 2. In addition, it contains only invertible matrices [19, Remark 7.1.2]. We will use the following explicit matrix inverses:

$$\begin{matrix} 1 & 1 \\ 2 & 1 \end{matrix}^{-1} = \begin{matrix} 1 & 1 \\ 2 & 1 \end{matrix} , \begin{matrix} 1 & 1 \\ 2 & 1 \end{matrix}^{-1} = \frac{1}{3} \begin{matrix} 1 & 1 \\ 2 & 1 \end{matrix} , \begin{matrix} 1 & 1 \\ 2 & 1 \end{matrix}^{-1} = \frac{1}{3} \begin{matrix} 1 & 1 \\ 2 & 1 \end{matrix} .$$

Using the above, one can verify that Φ is a homeomorphism whose inverse is also piecewise linear. We set $\Psi = \Phi^{-1}$; it is given by

$$\begin{aligned} \Psi(u, v) &= (v \quad u, 2u \quad v) && \text{for } (u, v) \; 2 \; A, \\ \Psi(u, v) &= \frac{1}{3} (u + v, 2u \quad v) && \text{for } (u, v) \; 2 \; B, \\ \Psi(u, v) &= (u + v, 2u + v) && \text{for } (u, v) \; 2 \; C, \\ \Psi(u, v) &= \frac{1}{3} (v \quad u, 2u + v) && \text{for } (u, v) \; 2 \; D, \end{aligned}$$

where the subsets A, B, C, D form a "partition"[2] of R^2

$$\begin{aligned} A = \; & (u, v) \; 2 \; \mathrm{R}^2 : v \quad u \quad 0, 2u \quad v \quad 0 && \text{(corresponding to } x \quad 0, y \quad 0), \\ B = \; & (u, v) \; 2 \; \mathrm{R}^2 : u + v \quad 0, 2u \quad v \quad 0 && \text{(corresponding to } x \quad 0, y \quad 0), \\ C = \; & (u, v) \; 2 \; \mathrm{R}^2 : u + v \quad 0, 2u + v \quad 0 && \text{(corresponding to } x \quad 0, y \quad 0), \\ D = \; & (u, v) \; 2 \; \mathrm{R}^2 : v \quad u \quad 0, 2u + v \quad 0 && \text{(corresponding to } x \quad 0, y \quad 0). \end{aligned}$$

[2]Each piece having two half lines in common with other pieces.

A graphical representation of these sets is given in Figure 4.

From this explicit piecewise linear representation of Ψ, we deduce that its Clarke Jacobian at 0 is the following

$$\mathrm{Jac}^c\,\Psi(0) = \mathrm{conv}\quad \begin{matrix}1 & 1\\ 2 & 1\end{matrix},\ \begin{matrix}1 & 1\\ 2 & 1\end{matrix},\ \begin{matrix}\frac{1}{3} & 1 & 1\\ 2 & 1\end{matrix},\ \begin{matrix}\frac{1}{3} & 1 & 1\\ 2 & 1\end{matrix}\quad .$$

For a given subset of linear space we denote by $\mathrm{aff}\,F$ the affine span of F. It is easy to see that $\dim \mathrm{aff}[\,\mathrm{Jac}^c\,\Phi(0)] = 2$ while $\dim \mathrm{aff}\,[\mathrm{Jac}^c\,\Psi(0)] = 3$. More concretely, vectorialize the set $\mathrm{Jac}^c\,\Psi(0)$ at $M = \frac{1}{3}\begin{matrix}1 & 1\\ 2 & 1\end{matrix}$ by considering the matrices given by

$$\begin{matrix}1 & 1\\ 2 & 1\end{matrix}\quad M,\qquad \begin{matrix}1 & 1\\ 2 & 1\end{matrix}\quad M,\qquad \begin{matrix}\frac{1}{3} & 1 & 1\\ 2 & 1\end{matrix}\quad M$$

that is

$$\begin{matrix}\frac{1}{3} & 2 & 2\\ 4 & 4\end{matrix},\quad \begin{matrix}\frac{1}{3} & 4 & 2\\ 4 & 2\end{matrix},\quad \begin{matrix}\frac{1}{3} & 2 & 0\\ 0 & 2\end{matrix}\ .$$

These matrices are independent so that $\mathrm{Jac}^c\,\Psi(0)$ is an affine set whose dimension is 3.

Matrix inversion is a semialgebraic diffeomorphism (when restricted to invertible matrices) so it preserves dimension. For this reason the set $[\mathrm{Jac}^c\,\Psi(0)]^{-1} = \{M^{-1}, M \in \mathrm{Jac}^c\,\Psi(0)\}$ is a semialgebraic set of dimension 3, and we have

$$[\mathrm{Jac}^c\,\Psi(0)]^{-1} \subset [\mathrm{Jac}^c\,\Phi(0)]. \tag{17}$$

However, we have shown that $z \in [\mathrm{Jac}^c\,\Psi(\Phi(z))]^{-1}$ is a conservative Jacobian. This example excludes the possibility of a simple inverse (implicit) function theorem with a "Clarke Jacobian calculus" and illustrates the requirement for a more flexible notion (conservativity) when using calculus rules in an implicit function (or inverse function) context.

The Lipschitz definable implicit and inverse function theorems. In the definable (e.g. semialgebraic case) our results have a remarkably simple expression that we give below.

Theorem 4 (Lipschitz definable inverse function theorem) *Let U and V be two open neighborhoods of 0 in R^n and $\Phi : U \to V$ a locally Lipschitz definable mapping with $\Phi(0) = 0$. Assume that Φ has a conservative Jacobian J_Φ such that $J_\Phi(0)$ contains only invertible matrices. Then, locally, Φ has locally Lipschitz definable inverse Ψ with a conservative Jacobian given by*

$$J_\Psi(y) = \{A^{-1} : A \in J_\Phi(\Psi(y))\}\ .$$

Proof: It suffices to use the fact that definable mappings are path differentiable, see [14], and that the the graph of Ψ is given by a first-order formula.

The same type of arguments gives:

Theorem 5 (Lipschitz definable implicit function theorem) *Let $F : \mathrm{R}^n \times \mathrm{R}^m \to \mathrm{R}^m$ be locally Lipchitz and definable with conservative Jacobian J_F. Let $(\hat{x}, \hat{y}) \in \mathrm{R}^n \times \mathrm{R}^m$ be such that $F(\hat{x}, \hat{y}) = 0$. Assume that $J_F(\hat{x}, \hat{y})$ is convex and that, for each $[A\ B] \in J_F(\hat{x}, \hat{y})$, the matrix B is invertible. Then, there exists an open neighborhood $U \times V$ of $(\hat{x}, \hat{y})$ and a locally Lipschitz definable function $G : U \to V$ such that, for all $x \in U$,*

$$F(x, G(x)) = 0.$$

Moreover, for each $x \in U$, the mapping $J_G : x \to -B^{-1}A : [A\ B] \in J_F(x, G(x))$ is conservative for G.

C Results from Section 3

C.1 Monotone operator deep equilibrium networks

Proposition 3 (Path differentiation through monotone layers) *Assume that J_σ is convex-valued and that, for all $J \in J_\sigma(Wz(W, b) + b)$, the matrix $(\mathrm{Id}_m \quad JW)$ is invertible. Consider a loss-like*

function $\ell\colon \mathrm{R}^m \to \mathrm{R}$ *with conservative gradient* $D_\ell\colon \mathrm{R}^m \to \mathrm{R}^m$, *then* $g : (W, z) \mapsto \ell(z(W, b))$ *is path differentiable and has a conservative gradient* D_g *defined through*

$$D_g\colon (W,b) \to J^T(\mathrm{Id}_m - JW)^{-T} v z^T, J^T(\mathrm{Id}_m - JW)^{-T} v) : J \in J_\sigma(Wz+b),\ v \in D_\ell(z) \quad .$$

Proof: The quantity $z(W, b)$ is defined implicitly by the relation

$$z(W,b) - \sigma(Wz(W,b)+b) = 0. \tag{18}$$

We set $M = m + m + m - m$ and represent the pair $(W, b) \in \mathrm{R}^{m\times m} \times \mathrm{R}^m$ as $(w_1, \ldots, w_m, b) \in \mathrm{R}^{M-m}$ where $w_i \in \mathrm{R}^m$ is the i-th row of W for $i \in \{1, \ldots, m\}$. We denote by $B\colon \mathrm{R}^M \to \mathrm{R}^m$ the bilinear map defined as

$$B(w_1, \ldots, w_m, b, z) := Wz + b$$

so that B is infinitely differentiable. Equation (18) is then equivalent to

$$z - (\sigma - B)(w_1, \ldots, w_m, b, z) = 0.$$

We denote by F the mapping

$$F\colon (w_1, \ldots, w_m, b, z) \mapsto z - (\sigma - B)(w_1, \ldots, w_m, b, z).$$

For $i \in \{1 \ldots m\}$, denote by $Z_i \in \mathrm{R}^{m\times m}$ the matrix whose i-th row is z, and remaining rows are null. The Jacobian of B, $\mathrm{Jac}\, B\colon \mathrm{R}^M \to \mathrm{R}^{m\times M}$ is as follows:

$$\mathrm{Jac}\, B(w_1, \ldots, w_m, b, z) = [Z_1 \quad \ldots \quad Z_m \quad \mathrm{Id}_m \quad W]$$

where $[A\, B]$ is used to denote the columnwise concatenation of matrices A and B. By hypothesis, we have a conservative Jacobian for σ, J_σ. Conservative Jacobians may be composed as usual Jacobians [14, Lemma 5]. As B is continuously differentiable, $\mathrm{Jac}\, B$ is also a conservative Jacobian for B. Therefore, we have the following conservative Jacobian for F,

$$J_F(w_1, \ldots, w_m, b, z) \quad \{[\, JZ_1 \quad \ldots \quad JZ_m \quad J - \mathrm{Id}_m - JW]\,,\ J \in J_\sigma(Wz+b)\}\,.$$

Finally, by hypothesis, for any W, b, and z such that $F(W, b, z) = 0$ and any $J \in J_\sigma(Wz + b)$, the matrix $\mathrm{Id}_m - JW$ is invertible. Therefore, Theorem 2 applies and, setting $\tilde{M} = m - m + m = M - m$, the set-valued mapping

$$\begin{aligned} J_z\colon \mathrm{R}^{\tilde{M}} &\to \mathrm{R}^{m\times\tilde{M}} \\ (w_1, \ldots, w_m, b) &\to (\mathrm{Id}_m - JW)^{-1} J\, [Z_1 \quad \ldots \quad Z_m \quad \mathrm{Id}_m]\,,\ J \in J_\sigma(Wz+b) \end{aligned}$$

is conservative for $(W, b) \mapsto z(W, b)$ as defined in (18). We denote by $Z \in \mathrm{R}^{m\times\tilde{M}}$ the matrix $[Z_1 \ldots Z_m\, \mathrm{Id}_m]$ appearing in the definition of J_z. Given the loss function ℓ, the mapping $J_\ell\colon z \mapsto \{v^T,\ v \in D_\ell(z)\}$ is a conservative Jacobian for ℓ [14, Lemma 3] and therefore, the set-valued mapping

$$\begin{aligned} J_g\colon \mathrm{R}^{\tilde{M}} &\to \mathrm{R}^{1\times\tilde{M}} \\ (w_1, \ldots, w_m, b) &\to v^T(\mathrm{Id}_m - JW)^{-1} JZ,\ J \in J_\sigma(Wz+b),\ v \in D_\ell(z(W,b)) \end{aligned}$$

is a conservative Jacobian for $g\colon (W, b) \mapsto \ell(z(W, b))$. Using [14, Lemma 4], we obtain a conservative gradient field for g by a simple transposition as follows

$$D_g\colon (w_1, \ldots, w_m, b) \to Z^T J^T(\mathrm{Id}_m - JW)^{-T} v,\ J \in J_\sigma(Wz+b),\ v \in D_\ell(z(W,b)) \quad .$$

We now identify the terms by block computation; recall that $Z = [Z_1 \ldots Z_m\, \mathrm{Id}_m]$ and that $Z_i \in \mathrm{R}^{m\times m}$ is the matrix whose i-th row is z with remaining rows null for each $i \in \{1, \ldots, m\}$. The term associated to b corresponds to the last $m - m$ block in Z, it is indeed of the form $J^T(\mathrm{Id}_m - JW)^{-T} v$. Similarly, for each $i \in \{1, \ldots, m\}$, the term associated to w_i is of the form $Z_i^T J^T(\mathrm{Id}_m - JW)^{-T} v$. For any $a \in \mathrm{R}^m$ and $i \in \{1, \ldots, m\}$, we have $Z_i^T a = a_i z$ where a_i is the i-th coordinate of a and z corresponds to the i-th row of Z_i^T. So the component associated to w_i in D_g is of the form $[J^T(\mathrm{Id}_m - JW)^{-T} v]_i z$, where $[\,]_i$ denotes the i-th coordinate. Since w_i denotes the i-th row of W, rearranging this expression in matrix format provides a term of the form $J^T(\mathrm{Id}_m - JW)^{-T} v z^T$ for the W component. This concludes the proof.

C.2 Optimization layers: the conic program case

Let us first expand on the link between zeros of the residual map and KKT solutions. We provide a simplified view of [17, 3], ignoring cases of infeasibility and unboundedness. Note that this corresponds to enforcing $w = 1$ as done in [2, 3].

The following is due to Moreau [38]. Recall that the polar of a closed convex cone K R^m is given by $K^\circ =$ x 2 R^m, $y^T x$ 0, 8y 2 K , in which case $(K^\circ)^\circ = K$ and the dual cone satisfies $K^* =$ K°.

Proposition 6 *Let s, y, v 2 R^m; the following are equivalent*

- $v = s + y$, s 2 K, y 2 K°, $s^T y = 0$.
- $s = P_{\mathcal{K}}(v)$, $y = P_{\mathcal{K}}$ (v).

We may reformulate this equivalence as follows, using changes of signs on y and v, noticing that $P_{\mathcal{K}}$ ( ) $= P_{\mathcal{K}}$ ( ) since $K^* =$ K°,

(i) $v = y$ s, s 2 K, y 2 K^*, $s^T y = 0$.
(ii) $s = P_{\mathcal{K}}$ (v) v, $y = P_{\mathcal{K}}$ (v).

Now the KKT system in (x, y, s) for problem (P) and (D) can be written as follows (see, for example, [17]),

$$\begin{aligned} A^T y + c = 0, \quad & y \text{ 2 } K^* \\ Ax + b = s, \quad & s \text{ 2 } K \\ s^T y = 0 & \end{aligned}$$

which is equivalent, by setting $v = y$ s and $u = x$, to

$$\begin{aligned} A^T P_{\mathcal{K}} \text{ } (v) + c = 0 & \\ Au + b = P_{\mathcal{K}} \text{ } (v) \quad v & \end{aligned} \tag{19}$$

The system (19) is equivalent to $N(z, A, b, c) = 0$ with $z = (u, v)$. We have shown that (x, y, s) is a KKT solution to the system if and only if $(x, y, s) = (u, P_{\mathcal{K}}$ $(v), P_{\mathcal{K}}$ (v) $v) = \phi(z)$ for $z = (x, y$ $s)$ such that $N(z, A, b, c) = 0$.

Proposition 4 (Path differentiation through cone programming layers) *Assume that $P_{\mathcal{K}}$, N are path differentiable, denote respectively by $J_{P_{\mathcal{K}}}$, $J_{\mathcal{N}}$ corresponding convex-valued conservative Jacobians. Assume that for all A, b, c 2 $\mathrm{R}^{m\times n}$ R^m R^n, $z = \nu(A, b, c)$ 2 R^n R^m is the unique solution to $N(z, A, b, c) = 0$, and that all matrices formed from the N first columns of $J_{\mathcal{N}}(z, A, b, c)$ are invertible. Then, ϕ, ν, and* sol *are path differentiable functions with conservative Jacobians:*

$$\begin{aligned} J_\nu(A, b, c) &:= \quad U^{-1}V : [U \; V] \text{ 2 } J_{\mathcal{N}}(\nu(A, b, c), A, b, c) \quad , \\ J_\phi(z) &:= \begin{pmatrix} \mathrm{Id}_n & 0 \\ 0 & J_{P_{\mathcal{K}}} \text{ } (v) \\ 0 & (J_{P_{\mathcal{K}}} \text{ } (v) \quad \mathrm{Id}_m) \end{pmatrix} , \\ J_{\mathrm{sol}}(A, b, c) &:= J_\phi(\nu(A, b, c)) J_\nu(A, b, c). \end{aligned}$$

Proof: First, the assumptions clearly ensure that ν and sol are single-valued and can be interpreted as functions such that $\mathrm{sol} = \phi$ ν. By assumption, ϕ is differentiable. We will first use Corollary 1 to obtain a conservative Jacobian for ν and then justify the expression for ϕ. The composition obtained for J_{sol} results from Proposition 2.

Let A, b, c 2 $\mathrm{R}^{m\times n}$ R^m R^n, $z := (u, v)$ 2 R^n R^m such that $N(z, A, b, c) = 0$. By assumption, the submatrices formed from the first N columns of $J_{\mathcal{N}}(z, A, b, c)$ are invertible. Then applying Corollary 1, there exist open neighborhoods U $\mathrm{R}^{m\times n}$ R^m R^n and V R^N and a locally Lipschitz function $G : U$! V satisfying, for all s 2 U $N(G(s), s) = 0$ with G is path differentiable. Since, by assumption, the solution $\nu(A, b, c)$ to $N(\nu(A, b, c), A, b, c) = 0$ is unique,

ν coincides with G on U. Thus, ν is path differentiable and a conservative Jacobian for ν is given by:

$$J_\nu(A,b,c) = \quad U^{-1}V : [U\ V]\ 2\ J_{\mathcal{N}}(\nu(A,b,c),A,b,c)$$

Let us now turn to ϕ. Since $P_{\mathcal{K}}$ has for conservative Jacobian J_{P_K}, we may construct a conservative Jacobian for the function ϕ as follows using [14, Lemmas 3, 4, and 5]:

$$J_\phi(z) = \begin{bmatrix} \mathrm{Id}_n & 0 & \\ 0 & J_{P_K}(v) & \\ 0 & (J_{P_K}(v) & \mathrm{Id}_m) \end{bmatrix}.$$

It follows from Proposition 2 that the composition $\mathrm{sol} = \phi \quad \nu$ is also path differentiable with conservative Jacobian

$$J_{\mathrm{sol}}(A,b,c) = J_\phi(\nu(A,b,c))J_\nu(A,b,c).$$

C.3 Hyperparameter selection for nonsmooth Lasso-type model

Proposition 5 (Conservative Jacobian for the solution mapping) *For all $\lambda\ 2$ R, assume $X_{\mathcal{E}}^T X_{\mathcal{E}}$ is invertible where $X_{\mathcal{E}}$ is the submatrix of X formed by taking the columns indexed by E. Then $\hat{\beta}(\lambda)$ is single-valued, path differentiable with conservative Jacobian, $J_{\hat{\beta}}(\lambda)$, given for all λ as*

$$\left\{ e^\lambda\ \mathrm{Id}_p \quad \mathrm{diag}(q)\ \mathrm{Id}_p \quad X^TX \right)^{-1} \mathrm{diag}(q)\,\mathrm{sign}\ \hat{\beta} \quad X^T \quad X\hat{\beta} \quad y \Big) : q\ 2\ M(\lambda) \right\}$$

where $M(\lambda) \quad \mathrm{R}^p$ *is the set of vectors q such that* $q_i\ 2 \begin{cases} \{1\} & i\ 2\ \mathrm{supp}\,\hat{\beta} \\ [0,1] & i\ 2\ E\ n\,\mathrm{supp}\,\hat{\beta} \\ \{0\} & i\ \not{2}\ E \end{cases}$.

Proof: Our goal is to apply Corollary 1 to the path differentiable "optimality gap" function $F : \mathrm{R} \quad \mathrm{R}^p\ !\ \mathrm{R}^p$ defined in (3). For each $\lambda\ 2$ R, the invertibility of $X_{\mathcal{E}}^T X_{\mathcal{E}}$ guarantees the uniqueness of $\hat{\beta}(\lambda)$ (see [39], [36, Lemma 1]), i.e., $\hat{\beta} : \mathrm{R}\ !\ \mathrm{R}^p$ is a function. Because $k\,k_1$ is separable, the components of the prox can be written, for any $(\lambda,u)\ 2$ R $\quad \mathrm{R}^p$, for all $i\ 2\ \{1,\dots,p\}$, as

$$[\mathrm{prox}_{e\ \|\cdot\|_1}(u)]_i = \mathrm{prox}_{e\ |\cdot|}(u_i)$$

which have Clarke subdifferentials

$$\partial^c \mathrm{prox}_{e\ |\cdot|} : u_i \quad \mathbb{1}_{u_i,e} \qquad \begin{matrix} 1 \\ \mathrm{sign}(u_i) \end{matrix} \quad \text{where} \quad \mathbb{1}_e\ (u_i) := \begin{cases} \{0\} & |u_i| < e^\lambda \\ [0,1] & |u_i| = e^\lambda \\ \{1\} & |u_i| > e^\lambda \end{cases}.$$

Thus a conservative Jacobian for F at (λ,β) is given by

$$J_F : (\lambda,\beta) \quad \{[\underbrace{e^\lambda \mathrm{diag}(q)\mathrm{sign}(\beta}_{A} \quad \underbrace{X^T(X\beta \quad y)\Big)}_{B} \quad \underbrace{\mathrm{Id}_p \quad \mathrm{diag}(q)}_{B} \quad \underbrace{\mathrm{Id}_p \quad X^TX}_{B}] : q\ 2\ C\} \tag{20}$$

with $C := \{q : q_i\ 2\ \mathbb{1}_e \quad \beta_i \quad X_i^T(X\beta \quad y)\}$. Let us estimate the factors q_i above in terms of the equicorrelation set E. Recall the KKT conditions [46] for the Lasso problem; a solution $\hat{\beta}$ must satisfy

$$X^T \quad y \quad X\hat{\beta} \quad = e^\lambda \delta \qquad \text{where} \qquad \delta_i\ 2 \begin{cases} \{\mathrm{sign}\ \hat{\beta}_i\} & i\ 2\ \mathrm{supp}\,\hat{\beta} \\ [\ 1,1] & i\ \not{2}\ \mathrm{supp}\,\hat{\beta} \end{cases}. \tag{21}$$

For $i\ 2\ \mathrm{supp}\,\hat{\beta}$, (21) gives

$$\begin{aligned} X_i^T \quad y \quad X\hat{\beta} \quad = e^\lambda \mathrm{sign} \quad \hat{\beta}_i \quad &=) \quad \mathrm{sign} \quad X_i^T \quad y \quad X\hat{\beta} \quad = \mathrm{sign} \quad \hat{\beta}_i \\ &=) \quad \mathrm{sign} \quad \hat{\beta}_i \quad = \mathrm{sign} \quad \hat{\beta}_i \quad X_i^T \quad X\hat{\beta} \quad y \\ &\qquad\qquad = \mathrm{sign} \quad X_i^T \quad y \quad X\hat{\beta} \quad . \end{aligned}$$

Noting that $\hat{\beta}_i > 0$ and $X_i^T (y - X\hat{\beta}) = e^\lambda$ since $i \in \operatorname{supp}\hat{\beta} \subseteq E$,

$$\begin{aligned}
\hat{\beta}_i X_i^T X\hat{\beta} &= \operatorname{sign}(\hat{\beta}_i) X_i^T X\hat{\beta} - \hat{\beta}_i X_i^T X\hat{\beta} \\
&= \operatorname{sign}(\hat{\beta}_i)\hat{\beta}_i + \operatorname{sign}(X_i^T(y - X\hat{\beta}))X_i^T(y - X\hat{\beta}) \\
&= \underbrace{\hat{\beta}_i}_{>0} + \underbrace{X_i^T(y - X\hat{\beta})}_{=e} \\
&\Rightarrow q_i = 1.
\end{aligned}$$

For $i \notin E$, $\hat{\beta}_i = 0$ since $\operatorname{supp}\hat{\beta} \subseteq E$. By (21), we have $|X_i^T(y - X\hat{\beta})| \leq e^\lambda$. However, since $i \notin E$, the inequality is strict

$$|X_i^T(y - X\hat{\beta})| < e^\lambda$$

and can be used to solve for q_i

$$\hat{\beta}_i X_i^T X\hat{\beta} = X_i^T(y - X\hat{\beta}) < e^\lambda \Rightarrow q_i = 0.$$

Finally, for $i \in E \setminus \operatorname{supp}\hat{\beta}$, $\hat{\beta}_i = 0$ and $|X_i^T X\hat{\beta}| = |X_i^T y| = e^\lambda$ which gives

$$\hat{\beta}_i X_i^T X\hat{\beta} = X_i^T y = e^\lambda$$

and thus $q_i \in [0, 1]$. Putting everything together we get an expression for q_i in terms of E and $\operatorname{supp}\hat{\beta}$

$$q_i \in \begin{cases} \{1\} & i \in \operatorname{supp}\hat{\beta} \\ [0,1] & i \in E \setminus \operatorname{supp}\hat{\beta} \\ \{0\} & i \notin E \end{cases}, \tag{22}$$

i.e., $q \in M$. We proceed to show that B is invertible for all $\lambda \in \mathrm{R}$. Denote $Q := \operatorname{diag}(q)$ for brevity; using the same argument of [50, Theorem 2] involving similarity transformations and continuity, the matrix B is invertible if and only if

$$\tilde{B} := \mathrm{Id}_p - Q^{1/2} + \mathrm{Id}_p - X^T X + Q^{1/2} = \mathrm{Id}_p - Q + Q^{1/2} X^T X Q^{1/2}$$

is invertible. Since $\tilde{B} - \mathrm{Id}_p - Q$, it follows that $\ker \tilde{B} \subseteq \ker(\mathrm{Id}_p - Q)$, however $\ker(\mathrm{Id}_p - Q)$ is a subspace of $W_\mathcal{E} := \operatorname{span}\{e_j : j \in E\}$ corresponding to $q_j = 1$. Since $q_j = 1 \Rightarrow j \in E$ by (22), the restriction of $\tilde{B}$ to $\ker(\mathrm{Id}_p - Q)$ is a principal submatrix of (possibly equal to) $X_\mathcal{E}^T X_\mathcal{E}$ which is invertible by assumption. Thus B is invertible and applying Corollary 1 then yields the final result.

Remark 4 Taking $q_i = 1$ for all $i \in E$ gives a selection of the conservative Jacobian for $\hat{\beta}$ in Proposition 5, for all $j \in \{1, \ldots, p\}$,

$$[J_{\hat{\beta}}(\lambda)]_j = \left[e^\lambda \left(X_\mathcal{E}^T X_\mathcal{E} \right)^{-1} \operatorname{sign}(X_\mathcal{E}^T(y - X\hat{\beta})) \right]_j \text{ if } j \in E, \text{ and } [J_{\hat{\beta}}(\lambda)]_j = 0 \text{ otherwise.}$$

This corresponds to the directional derivative given by LARS algorithm [27], see also [36]. Alternatively, taking $q_i = 0$ for $i \notin \operatorname{supp}\hat{\beta}$ gives, for all $j \in \{1, \ldots, p\}$,

$$[J_{\hat{\beta}}(\lambda)]_j = \left[e^\lambda (X_{\operatorname{supp}\hat{\beta}}^T X_{\operatorname{supp}\hat{\beta}}^{-1}) \operatorname{sign}(X_{\operatorname{supp}\hat{\beta}}^T (y - X\hat{\beta})) \right]_j, \text{ if } j \in \operatorname{supp}\hat{\beta}$$

and $[J_{\hat{\beta}}(\lambda)]_j = 0$ otherwise. This is the weak derivative given by [9]. Both of these expressions are particular selections in $J_{\hat{\beta}}$, which is the underlying conservative field. They agree if $E = \operatorname{supp}\hat{\beta}$, which holds under qualification assumptions, see for example [10] and references therein.

D Results from Section 4

Theorem 3 (Convergence result) *Consider minimizing ℓ given in* (8) *using algorithm* (10) *under Assumption 1. Assume furthermore the following*

- ***Step size:*** $\sum_{k=1}^{+\infty} \alpha_k = +\infty$ *and* $\alpha_k = o(1/\log(k))$.
- ***Boundedness:*** *there exists $M > 0$, and $K \subset \mathrm{R}^p$ open and bounded, such that, for all $s \in (s_{\min}, s_{\max})$ and $w_0 \in \mathrm{cl}\, K$, $\|w_k\| \leq M$ almost surely.*

For almost all $w_0 \in K$ and $s \in (s_{\min}, s_{\max})$, the objective value $\ell(w_k)$ converges and all accumulation points $\bar{w}$ of w_k are Clarke-critical in the sense that $0 \in \partial^c \ell(\bar{w})$.

Proof: We first show that if w_0 is taken uniformly at random on K then, almost surely, all iterates $(w_k)_{k\in\mathbb{N}}$ are random variables which are absolutely continuous with respect to the Lebesgue measure. This is essentially a repeating of the arguments developed in [11] for constant step sizes. Assume from now on that w_0 is random, uniformly on K.

For $i \in \{1, \ldots, N\}$, denoting by $\phi(\,\cdot\,, i)\colon \mathrm{R}^p \to \mathrm{R}^p$ the output of backpropagation applied to $\ell_i = g_{i,L} \circ g_{i,L-1} \circ \cdots \circ g_{i,1}$, we have that $x \mapsto \phi(x, i)$ is a selection in the conservative Jacobian (actually conservative gradient) J_i. Therefore, using [11, Proposition 1] the sequence $(w_k)_{k\in\mathbb{N}}$ is an SGD sequence in the sense of [11, Definition 2].

Compositions of definable functions and functions implicitly defined based on definable functions are definable. Therefore by Assumption 1, for each $i \in \{1, \ldots, N\}$, ℓ_i is locally Lipschitz and definable and thus so is ℓ. Definable functions are twice differentiable almost everywhere so that [11, Proposition 3] applies. Following the recursion argument in [11, Proposition 2], there exists a set $\Gamma \subset (0, +\infty)$ of full Lebesgue measure such that, if $s\alpha_k \in \Gamma$ for all $k \in \mathbb{N}$, each iterate $(w_k)_{k\in\mathbb{N}}$ is a random variable which is absolutely continuous with respect to the Lebesgue measure. We have that

$$\{s \in (s_{\min}, s_{\max}) : \exists k \in \mathbb{N}, s\alpha_k \in (0, +\infty) \setminus \Gamma\} = \bigcup_{k=1}^{+\infty} \{s \in (s_{\min}, s_{\max}) : s\alpha_k \in (0, +\infty) \setminus \Gamma\}$$

is a countable union of null sets and thus a null set, i.e., for almost all $s \in (s_{\min}, s_{\max})$, for all $k \in \mathbb{N}$, $s\alpha_k \in \Gamma$. As a result, for almost all s, w_k has a density with respect to the Lebesgue measure for all $k \in \mathbb{N}$.

Conservative gradients are gradients almost everywhere and so there is a full measure set S such that, for all $w \in S$ and all $i \in \{1, \ldots, N\}$, $J_i(w) = \{\nabla \ell_i(w)\}$ [14, Theorem 1]. Combining this with the fact that each element of the sequence is absolutely continuous with respect to the Lebesgue measure, the same argument as in [11, Theorem 1] gives, for almost all $s \in (s_{\min}, s_{\max})$, for every $k \in \mathbb{N}$, almost surely

$$w_{k+1} = w_k - s\alpha_k \nabla \ell_{I_k}(w_k)$$

and

$$\mathrm{E}(w_{k+1} | w_0, \ldots, w_k) = w_k - s\alpha_k \nabla \ell(w_k) = w_k - s\alpha_k \partial^c \ell(w_k).$$

Therefore, the sequence is actually a Clarke stochastic subgradient sequence almost surely (see, for example, [24]) and thus can be analyzed using the method developed in [8]. Indeed, conservativity ensures that ℓ is a Lyapunov function for the differential inclusion $\dot{w} \in -\partial^c \ell(w)$, that is decreasing along solutions, strictly outside of $\mathrm{crit}_\ell := \{w \in \mathrm{R}^p, 0 \in \partial^c \ell(w)\}$. Since ℓ is definable, the set of its critical values, $\ell(\mathrm{crit}_\ell)$ is finite [13] and thus has empty interior. By [8, Theorem 3.6] and [8, Proposition 3.27], it is then guaranteed that $\ell(\bar{w})$ is constant for all accumulation points $\bar{w}$ of $(w_k)_{k\in\mathbb{N}}$ and that $0 \in \partial^c \ell(\bar{w})$. This occurs almost surely with respect to the randomness induced by w_0 and $(I_k)_{k\in\mathbb{N}}$ and therefore it is true with probability one for almost all w_0.

E Results from Section 5

E.1 Cyclic gradient descent

E.1.1 Fixed-point formulation

Consider the optimization problem

$$(s_1, s_2) \; 2 \quad \underset{(a,b)\in[0,3]\times[0,5]}{\arg\max} \quad (a+b)(\quad 3x+y+2). \tag{23}$$

The optimality condition for this problem can be expressed using the fixed-point equation of the projected gradient descent algorithm. Denote for $x, y \; 2 \; \mathrm{R}^2$, $q_{x,y} : (a,b) \; 7! \;\; (a+b)(\quad 3x+y+2)$; we can verify (s_1, s_2) is solution to (11) if and only if it satisfies the equality

$$\begin{matrix} s_1 \\ s_2 \end{matrix} \quad = P_{\mathcal{U}} \quad \begin{matrix} s_1 \\ s_2 \end{matrix} \quad + \; r \, q_{x,y}(s_1, s_2) \quad = P_{\mathcal{U}} \quad \begin{matrix} s_1 \\ s_2 \end{matrix} \quad + \quad \begin{matrix} 3x+y+2 \\ 3x+y+2 \end{matrix} \quad .$$

Where $P_{\mathcal{U}}$ is the projection on the set $U := [0,3] \quad [0,5]$ which can be implemented as a difference of relu functions

$$P_{\mathcal{U}}(x,y) = \mathrm{relu}(x,y) \quad \mathrm{relu}(x \quad 3, y \quad 5).$$

Let $h : \mathrm{R}^2 \quad \mathrm{R} \quad \mathrm{R} \;\; ! \;\; \mathrm{R}^2$ be the function

$$h : (s,x,y) \; 7! \;\; P_{\mathcal{U}} \quad \begin{matrix} s_1 \\ s_2 \end{matrix} \quad + \quad \begin{matrix} 3x+y+2 \\ 3x+y+2 \end{matrix} \quad .$$

Then the original problem (23) is equivalent to the fixed point equation $s = h(x,y,s)$. Indeed, we can easily verify the solutions $s : \mathrm{R}^2 \;\; ! \;\; \mathrm{R}^2$ to (23) are

$$s(x,y) = \begin{cases} f(0,0)g & \text{if} \quad 3x+y+2 < 0 \\ f(3,5)g & \text{if} \quad 3x+y+2 > 0 \\ [0,3] \quad [0,5] & \text{if} \quad 3x+y+2 = 0 \end{cases}$$

which creates a discontinuity for the function $\ell(\ , s(\ ))$, now expressed as

$$\ell(x,y,s(x,y)) = \begin{matrix} x^2+4y^2 & \text{if} & 3x+y+2<0 \\ (x \quad 3)^2+4(y \quad 5)^2 & \text{if} & 3x+y+2>0 \end{matrix} .$$

E.1.2 Perturbed experiments

Perturbed experiments are done on the following perturbed loss function

$$\ell_\varepsilon(x,y,s) = \quad \frac{1}{4} + \varepsilon_1 \quad (x \quad s_1)^2 + (1+\varepsilon_2)(y \quad s_2)^2$$

$$s \; 2 \; s_\varepsilon(x,y) := \arg\max f(a+b)(\quad (3+\varepsilon_3)x + y + 2 + \varepsilon_4) : \; a \; 2 \; [0,3 \quad \varepsilon_5], b \; 2 \; [0,5 \quad \varepsilon_6]g$$

with $\varepsilon_1, \ldots, \varepsilon_6$ the perturbations. In Figure **??**, we consider several realizations of independent Gaussian variables $\varepsilon_1, \ldots, \varepsilon_6 \quad N(0, \sigma^2)$ with $\sigma^2 = 0.05$; despite this added noise, the unwanted dynamics persist.

For different initial points the gradient flow converges to the same limit cycle (Figure 5a). The cycle persists even if we increase the noise variance σ^2.

(a) (b)

Figure 5: (a) Gradient flows for several initializations. (b) Gradient flows for 20 perturbed experiments with $\sigma^2 = 0.4$.

E.1.3 Conic canonicalization

Let $c \ 2 \ \mathrm{R}^2$ be a parameter vector and consider the problem

$$\max_{x \in [0,3] \times [0,5]} \quad c^T x.$$

It can be formulated as a cone program (P) and its dual (D):

$$\text{(P)} \quad \begin{array}{ll} \inf & c^T x \\ \text{subject to} & Ax + s = b \\ & s \ 2 \ K \end{array} \qquad\qquad \text{(D)} \quad \begin{array}{ll} \inf & b^T y \\ \text{subject to} & A^T y + c = 0 \\ & y \ 2 \ K^*, \end{array} \tag{24}$$

where

$$A = \begin{matrix} \mathrm{Id}_2 \\ \mathrm{Id}_2 \end{matrix} \quad \text{and} \quad b = \begin{matrix} 2 \\ 3 \\ 6 \\ 5 \\ 0 \\ 0 \end{matrix}.$$

Let (x, y, s) be a solution to the cone program (24) where x is the primal variable, y is the dual variable, and s the primal slack variable. Then it follows from (6) that a solution z to $N(z, c) = 0$ is obtained by $z = (x, y \quad s)$. For $c = (0, 0)$, the solutions are $x \ 2 \ [0, 3] \quad [0, 5]$, $s = b \quad Ax$, and $y = (0, 0, 0, 0)$, hence the uniqueness assumption for Proposition 4 is not satisfied.

E.1.4 A chaotic dynamics in R^4

We combine two cycles of the previous example into a gradient dynamics in R^4. To perform this, we consider a block-separable sum of the same function where we add a scaling parameter $\eta > 0$:

$$g : (x, y, z, w) \ 7! \ f(x, y) + \eta f(z, w).$$

This will combine the two cycles but the parameter η will make one cycle "faster" than the other. Projecting the path of the gradient descent on the variables (y, z) we obtain a chaotic dynamics filling the space as the number of iterations increases.