

A descent algorithm for the optimal control of ReLU neural network informed PDEs based on approximate directional derivatives

Guozhi Dong¹, Michael Hintermüller^{2,3}, Kostas Papafitsoros⁴

submitted: October 25, 2022

¹ School of Mathematics and Statistics
HNP-LAMA, Central South University
Lushan South Road 932
410083 Changsha, China
E-Mail: guozhi.dong@csu.edu.cn

² Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: michael.hintermueller@wias-berlin.de

³ Institute for Mathematics
Humboldt-Universität zu Berlin
Unter den Linden 6, 10099 Berlin, Germany
E-Mail: hint@math.hu-berlin.de

⁴ School of Mathematical Sciences
Queen Mary University of London
Mile End Road, E1 4NS, UK
E-Mail: k.papafitsoros@qmul.ac.uk

No. 2964
Berlin 2022



2020 *Mathematics Subject Classification.* 35R30, 49J52, 49K20, 49M41, 68T07, 90C46.

Key words and phrases. Optimal control of nonsmooth partial differential equations, data-driven models, neural networks, bundle-free methods, descent algorithms.

This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689). The work of GD is supported by an NSFC grant, No. 12001194. The work of MH is partially supported by the DFG SPP 1962, project-145r.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: <http://www.wias-berlin.de/>

A descent algorithm for the optimal control of ReLU neural network informed PDEs based on approximate directional derivatives

Guozhi Dong, Michael Hintermüller, Kostas Papafitsoros

Abstract

We propose and analyze a numerical algorithm for solving a class of optimal control problems for learning-informed semilinear partial differential equations. The latter is a class of PDEs with constituents that are in principle unknown and are approximated by nonsmooth ReLU neural networks. We first show that a direct smoothing of the ReLU network with the aim to make use of classical numerical solvers can have certain disadvantages, namely potentially introducing multiple solutions for the corresponding state equation. This motivates us to devise a numerical algorithm that treats directly the nonsmooth optimal control problem, by employing a descent algorithm inspired by a bundle-free method. Several numerical examples are provided and the efficiency of the algorithm is shown.

1 Introduction

1.1 Context and motivation

In this paper we study a numerical algorithm for the following artificial neural network based optimal control problem:

$$\begin{aligned} & \text{minimize} && J(y, u) := \frac{1}{2} \|y - g\|_{L^2(\Omega)}^2 + \frac{\alpha}{2} \|u\|_{L^2(\Omega)}^2, && \text{over } (y, u) \in H_0^1(\Omega) \times L^2(\Omega), \\ & \text{subject to} && \begin{cases} -\Delta y + \mathcal{N}(\cdot, y) = u, & \text{in } \Omega, \\ y = 0, & \text{on } \partial\Omega, \end{cases} && \text{and } u \in \mathcal{C}_{ad}. \end{aligned} \tag{P_{\mathcal{N}}}$$

Here Ω denotes an open, bounded, Lipschitz domain in \mathbb{R}^d with boundary $\partial\Omega$, $d \geq 2$, $g \in L^2(\Omega)$ is a given desired state, $\alpha > 0$ is fixed, and \mathcal{C}_{ad} is an admissible set for the control u , which is assumed to be a nonempty, closed and convex subset of $L^p(\Omega)$ for some $p \geq 2$. The state (variable) is y which, given a control u , solves a semilinear elliptic partial differential equation (PDE), the state equation. The term that renders the above problem nonstandard is the function $\mathcal{N} : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}$, a constituent of the PDE acting as a constraint for the minimization problem. In fact, throughout we assume that \mathcal{N} represents a ReLU (Rectified Linear Unit) artificial neural network, that is, a neural network that has the ReLU $\sigma(t) := \max(t, 0)$ as its activation function. We note that the ReLU is one of the most common and advantageous activation functions in deep learning [6, 15], see Section 2 for more details and definitions. As a result, \mathcal{N} is in general a nonlinear and nonsmooth function. We mention that here we consider \mathcal{N} to be monotonically increasing in the variable y which guarantees the uniqueness of a solution to the state equation, resulting in a well-defined control-to-state map.

The semilinear PDE in $(P_{\mathcal{N}})$ is thus an instance of a *learning-informed* PDE, a concept that was introduced in [11] and further explored recently in other works [1, 20]. We assume that it forms an approximating model to an unknown ground truth physical law expressed by

$$\begin{cases} -\Delta y + f(\cdot, y) = u, & \text{in } \Omega, \\ y = 0, & \text{on } \partial\Omega, \end{cases} \quad (1.1)$$

with the function f being some unknown nonlinearity, which is approximated by the network \mathcal{N} . This could be achieved for instance in a setting where we have at our disposal a dataset

$$D := \{(y_i, u_i) : y_i \text{ (approximately) solves (1.1) for } u_i, i = 1 \dots, n_D\},$$

which corresponds to some pre-specified controls and associated state responses, collected for example through measurements or computations. This dataset can be used towards evaluation instances of f via $f(x_j, y_i(x_j)) \simeq u_i(x_j) + \Delta y_i(x_j)$ where $\{x_j\}_{j=1}^{\ell}$ is an appropriate discrete collection of points in the domain. Using these instances as a training set, a neural network \mathcal{N} can be trained in the context of supervised learning in an *offline phase*, and take the role of an approximating map for the unknown f . Applications of the above framework were considered in [11] in order to learn the physical law that governs the separation of a fluid into two immiscible phases as well as to learn the physical law behind magnetic resonance imaging (MRI), where, instead of a PDE, a system of ordinary differential equations (ODEs) acts as a constraint [10].

Several theoretical aspects of the optimal control problem $(P_{\mathcal{N}})$ were studied in detail in [12]. There, existence and uniqueness of solutions to the state equation were shown, as well as continuity and directional differentiability properties of the control-to-state map. The main challenge here is the aforementioned nonsmoothness of \mathcal{N} due to the ReLU. In fact, it can be shown that the set of functions represented by ReLU neural networks coincides with the family of piecewise affine maps. In general one does not expect the associated control-to-state map $S_{\mathcal{N}}$ to be Gâteaux differentiable which poses difficulties in the derivation of first-order optimality conditions for the optimal control problem. Addressing this latter aspect, stationary conditions were derived in the companion work [12] based on generalized differentiability concepts. In this paper, we focus on establishing algorithms for the numerical solution of $(P_{\mathcal{N}})$ towards the approximation of so-called B-stationary points.

With the desire of making use of classical numerical solvers, an immediate approach to solving $(P_{\mathcal{N}})$ would be to regularize the problem by smoothing the nonsmooth component \mathcal{N} . As a consequence, the classical Karush-Kuhn-Tucker theory for stationarity (see, e.g., [27]) becomes available and solvers from (smooth) nonlinear programming, such as sequential quadratic programming [22, 19], may be employed. Indeed such an approach has also been for long used in order to derive limiting optimality conditions (under vanishing regularization) which unfortunately typically leads to stationarity systems containing less information when compared to the strong stationarity conditions as in [5, 9, 21], obtained by using nonsmooth analysis techniques. In this work, we show that in the case of ReLU learning-informed PDEs, additional issues can arise from a smoothing approach. In particular, due to a potentially large architecture of a network \mathcal{N} (large number of layers and neurons), a natural and efficient way to smoothen \mathcal{N} (after its training has been completed) would be via simply smoothing the ReLU function in \mathcal{N} , denoted now by σ_{ϵ} , resulting in a smooth network \mathcal{N}_{ϵ} approximating \mathcal{N} . We refer to this technique as *canonical smoothing* of \mathcal{N} . However we show with simple examples that this type of smoothing does not necessarily preserve monotonicity for deep enough networks, and in fact it does not even preserve it in a way that monotonicity of the PDE operator could still be shown. This possibly renders the resulting control-to-state map $S_{\mathcal{N}_{\epsilon}}$ multi-valued, posing difficulties when resorting to classical algorithms for the solution of the regularized problem. This is yet another motivation for

us to devise numerical methods which are capable of directly solving $(P_{\mathcal{N}})$. In this vein, we propose to adapt the bundle-free method from [18], originally developed for a class of mathematical programs with equilibrium constraints (MPECs). The proposed algorithm makes use of an auxiliary optimization problem as in [18], and we show that by approximating the derivatives of the ReLU network (but not the ReLU itself!) via a smoothed \max -function, then a descent direction for a reduced version of $(P_{\mathcal{N}})$ at a given control iterate is identified or (ideally) B -stationarity of that iterate can be diagnosed. We also mention that in [9], an algorithm for solving a very specific nonsmooth semilinear PDE (in a first-discretize-then-optimize flavor) in the absence of control constraints has been proposed, where $\mathcal{N}(\cdot, y) = \max(0, y)$. However, as it was also noted by the authors of [9] their algorithm cannot be applied to general nonsmooth semilinear PDEs, and an efficient algorithm for the general case calls for new ideas. The current paper aims to cover this gap.

1.2 Structure of the paper

In Section 2 we focus on the structure of the functional form of ReLU networks. We are in particular interested in understanding how this structure changes after smoothing the ReLU network function via regularization of the associated activation function (canonical smoothing). Here our main focus is on how this kind of smoothing can break the monotonicity of the network. The implication of the latter concerning the emergence of nonuniqueness of solutions of the learning-informed state equation is discussed in Section 3. We also collect basic results concerning the general optimal control problem $(P_{\mathcal{N}})$ and in particular we recall the stationarity conditions derived in [12]. In Section 4, we introduce and analyze a descent algorithm that directly treats the nonsmooth optimal control problem. It is applied in Section 5 to several instances of an optimal control problem with a ReLU network-informed semilinear second-order elliptic PDE. In particular, also a non-monotone setting is considered in order to challenge the solver.

2 Smoothings of ReLU neural networks

2.1 Definition and basic properties

We first fix some notation. For a set A , the characteristic and the indicator functions χ_A and \mathcal{X}_A , respectively, are defined as $\chi_A(x) = 1$ if $x \in A$ and $\chi_A(x) = 0$ otherwise, and $\mathcal{X}_A(x) = 0$ if $x \in A$ and $\mathcal{X}_A(x) = +\infty$ otherwise. Unless otherwise stated $\langle \cdot, \cdot \rangle$ denotes the standard L^2 inner product.

Definition 2.1 (Standard feedforward multilayer neural network). *Let $L \in \mathbb{N}$, network parameters $\theta = ((W_1, b_1), \dots, (W_L, b_L))$ with $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$, $b_i \in \mathbb{R}^{n_i}$, for $i = 1, \dots, L$ and $n_i \in \mathbb{N}$ for $i = 0, \dots, L$. Furthermore let $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ be an arbitrary function. We call a function $\mathcal{N} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_L}$ a neural network with weight matrices $(W_i)_{i=1}^L$, bias vectors $(b_i)_{i=1}^L$ (the network parameters) and activation function σ if $\mathcal{N}(x)$ can be defined through the following recursive relation for any $x \in \mathbb{R}^{n_0}$:*

$$z_0 = x, \tag{2.1}$$

$$z_\ell = \sigma(W_\ell z_{\ell-1} + b_\ell), \quad \ell = 1, \dots, L-1, \tag{2.2}$$

$$\mathcal{N}(x) = W_L z_{L-1} + b_L. \tag{2.3}$$

The action of the activation function σ in (2.2) is considered component-wise i.e. for a vector $y = (y^1, \dots, y^n) \in \mathbb{R}^n$ we set $\sigma(y) := (\sigma(y^1), \dots, \sigma(y^n))$. More compactly, \mathcal{N} can also be defined as

$$\mathcal{N}(x) = T_L \circ \sigma(T_{L-1}) \circ \dots \circ \sigma(T_2) \circ \sigma(T_1(x)), \quad x \in \mathbb{R}^{n_0}, \tag{2.4}$$

where for every $\ell = 1, \dots, L$, T_ℓ denotes the affine transformation $z \mapsto W_\ell z + b_\ell$.

We call \mathcal{N} a ReLU neural network if σ is the ReLU (Rectified Linear Unit) activation function:

$$\sigma(t) = \max(t, 0), \quad t \in \mathbb{R}. \quad (2.5)$$

Following the standard neural network terminology, we say that a neural network defined as in (2.1)–(2.3), has L layers and $L - 1$ hidden layers, with the latter denoting the operations in (2.2). The final operation (2.3) is called the *output layer*. Furthermore, n_i is the number of *neurons* in the i -th layer, $i = 1, \dots, L$, that is, it is the number of rows of the weight matrix W_i . The number of neurons of a given layer is also called the *width* of that layer, while the number of layers is called the *depth* of the network.

We should note that a neural network as a function, does not necessarily admit a unique representation with respect to the weight matrices, the bias vectors and the activation functions. Furthermore in the Definition 2.1, the input of the ℓ -th layer consists only of the output $z_{\ell-1}$ of the previous layer. A more general neural network definition would allow the input for each layer to depend on the output of all the previous layers. In that case every W_ℓ would be a weight matrix of size $\mathbb{R}^{n_i \times (\sum_{k=0}^{\ell-1} n_k)}$. However, since every network of the latter type can be realized by a network as in (2.1), see [16], we will stick to the more classical definition given above.

We are interested in the regularity of the functions that are realized by ReLU neural networks. It turns out that the latter class coincides with the class of *continuous piecewise affine functions*.

Definition 2.2 (Continuous piecewise affine functions). *Let $n_0 \in \mathbb{N}$. We say that a function $\mathcal{F} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}$ is continuous piecewise affine (CPWA) if the following condition holds:*

- \mathcal{F} is continuous and there exist finitely many affine maps $f_1, \dots, f_p : \mathbb{R}^{n_0} \rightarrow \mathbb{R}$ for some $p \in \mathbb{N}$ such that for every $x \in \mathbb{R}^{n_0}$, there exists an $i \in \{1, \dots, p\}$ such that $\mathcal{F}(x) = f_i(x)$.

We refer to [2, 3, 26] for further equivalent characterizations of CPWA functions.

Theorem 2.3. (Characterization of ReLU neural networks, [3]) *A function $\mathcal{N} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}$ is a ReLU neural network if and only if it is a CPWA function.*

From the definition (2.1)–(2.3) it is clear that $\mathcal{N} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_L}$, $n_L \geq 1$, is an \mathbb{R}^{n_L} -valued ReLU neural network if and only if $\mathcal{N} = (\mathcal{N}_1, \dots, \mathcal{N}_L)$ with each $\mathcal{N}_i : \mathbb{R}^{n_0} \rightarrow \mathbb{R}$, $i = 1, \dots, L$ being a scalar-valued ReLU neural network. Thus \mathcal{N} is an \mathbb{R}^{n_L} -valued ReLU neural network if and only if it is an \mathbb{R}^{n_L} -valued CPWA function, with the latter defined exactly as in Definition 2.2 with the only difference being that the affine maps f_i are \mathbb{R}^{n_L} -valued.

To give an example, for $p \geq 2$ and $t_1 \leq \dots \leq t_{p-1}$, we consider the following one dimensional continuous piecewise affine function \mathcal{F} with

$$\mathcal{F}(t) = \begin{cases} a_1 t + \gamma_1 & \text{if } t \leq t_1, \\ a_i t + \gamma_i & \text{if } t_{i-1} \leq t \leq t_i, \quad i = 2, \dots, p-1, \\ a_p t + \gamma_p & \text{if } t \geq t_{p-1}. \end{cases} \quad (2.6)$$

Note that we assume that $(a_i, \gamma_i)_{i=1}^p$ satisfy the appropriate conditions such that \mathcal{F} is continuous.

Then it can be checked, see for instance [2, Corollary 3.5], that \mathcal{F} can be written as

$$\begin{aligned}\mathcal{F}(t) &= a_1 t + \gamma_1 + \sum_{i=1}^{p-1} (a_{i+1} - a_i) \max(t - t_i, 0) \\ &= a_1 (\max(t, 0) - \max(-t, 0)) + \sum_{i=1}^{p-1} (a_{i+1} - a_i) \max(t - t_i, 0) + \gamma_1.\end{aligned}\quad (2.7)$$

This means that \mathcal{F} can be realized as a ReLU neural network with one hidden layer having $p + 1$ neurons. In particular, $\mathcal{F} = T_2 \circ \sigma(T_1)$, where $T_1(t) = W_1 t + b_1$, $T_2(z) = W_2 z + b_2$ with $W_1 = (1, -1, 1, 1, \dots, 1)^T \in \mathbb{R}^{(p+1) \times 1}$, $b_1 = (0, 0, -t_1, \dots, -t_{p-1})^T \in \mathbb{R}^{(p+1) \times 1}$, and $W_2 = (a_1, -a_1, a_2 - a_1, \dots, a_p - a_{p-1}) \in \mathbb{R}^{1 \times (p+1)}$, $b_2 = \gamma_1 \in \mathbb{R}$.

Another characteristic of ReLU neural networks are their approximation capabilities. In fact it can be easily checked that given a bounded domain $U \subset \mathbb{R}^{n_0}$ with Lipschitz boundary we have that for every $\epsilon > 0$ and $f \in W^{1,\infty}(U)$ there exists a ReLU neural network $\mathcal{N}_\epsilon : \mathbb{R}^{n_0} \rightarrow \mathbb{R}$ such that $\|\mathcal{N}_\epsilon - f\|_{W^{1,\infty}(U)} < \epsilon$, see also [12, Section 2.2].

2.2 Smoothings of ReLU neural networks

We are also interested in smoothing versions of ReLU networks. One canonical way to achieve smoothing is via appropriately smoothing the ReLU function σ which is the constituent of the network that determines its regularity. In optimal control, typically specific approximating sequences are used [9, 21, 23] which we will also employ here.

Definition 2.4 (Canonical smoothing of ReLU). *We say that the family $\sigma_\epsilon : \mathbb{R} \rightarrow \mathbb{R}$ (or ReLU_ϵ), $\epsilon > 0$, is a canonical smoothing of the ReLU function if:*

- (i) σ_ϵ is a positive, convex, monotonically increasing $C^1(\mathbb{R})$ function for all $\epsilon > 0$.
- (ii) $\sigma_\epsilon \rightarrow \sigma$ uniformly and monotonically as $\epsilon \rightarrow 0$, i.e.,

$$|\sigma_{\epsilon_1}(x) - \sigma(x)| \leq |\sigma_{\epsilon_2}(x) - \sigma(x)| \quad \text{for } 0 < \epsilon_1 \leq \epsilon_2 \quad \text{and for every } x \in \mathbb{R}.$$

We say that a family of networks $\mathcal{N}_\epsilon : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_L}$, $\epsilon > 0$, is a canonical smoothing of the ReLU network $\mathcal{N} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_L}$ if it results from \mathcal{N} by simply substituting the activation function σ by σ_ϵ .

Lemma 2.5. *Let $(\sigma_\epsilon)_{\epsilon>0}$, be a canonical smoothing of the ReLU function. Then the following two additional properties hold for $\epsilon > 0$ small enough:*

- (i) $0 \leq \sigma'_\epsilon(t) \leq 1$, for all $t \in \mathbb{R}$.
- (ii) For every $\delta > 0$, σ'_ϵ converges uniformly to 1 on $[\delta, \infty)$ and uniformly to 0 on $(-\infty, \delta]$ as $\epsilon \rightarrow 0$.

Proof. Suppose that (i) does not hold. Then because every σ_ϵ is convex and hence σ'_ϵ is increasing, there exists $\epsilon_n \rightarrow 0$ and $t_n \in \mathbb{R}$ such that $\sigma'_{\epsilon_n}(t) > 1$ for every $t \in [t_n, \infty)$. But that means that for every $n \in \mathbb{N}$ there exists $t \in [t_n, \infty)$ such that $\sigma_{\epsilon_n}(t)$ is arbitrary far away from $\sigma(t)$ contradicting the uniform convergence.

For (ii), we fix $\delta > 0$. Using a similar argument as before we deduce that for every $\epsilon > 0$ small enough it holds that $\lim_{t \rightarrow \infty} \sigma'_\epsilon(t) = 1$. Hence, given the monotonicity of σ'_ϵ and (i), it suffices to show that $\lim_{\epsilon \rightarrow 0} \sigma'_\epsilon(\delta) = 1$. But if this is not the case it can easily be checked that there exists a subsequence σ_{ϵ_n} and $\eta > 0$ such that $|\sigma_{\epsilon_n}(0)| > \eta$ contradicting the convergence $\lim_{\epsilon \rightarrow 0} \sigma_\epsilon(0) = 0$. The uniform convergence of σ'_ϵ to 0 on $(-\infty, \delta)$ is proved similarly. \square

There are numerous options for a canonical smoothing of the ReLU function, see for instance Figure 1. It is also clear that $\mathcal{N}_\epsilon \rightarrow \mathcal{N}$ uniformly but as we will show later with a counterexample the convergence does not have to be necessarily monotonic.

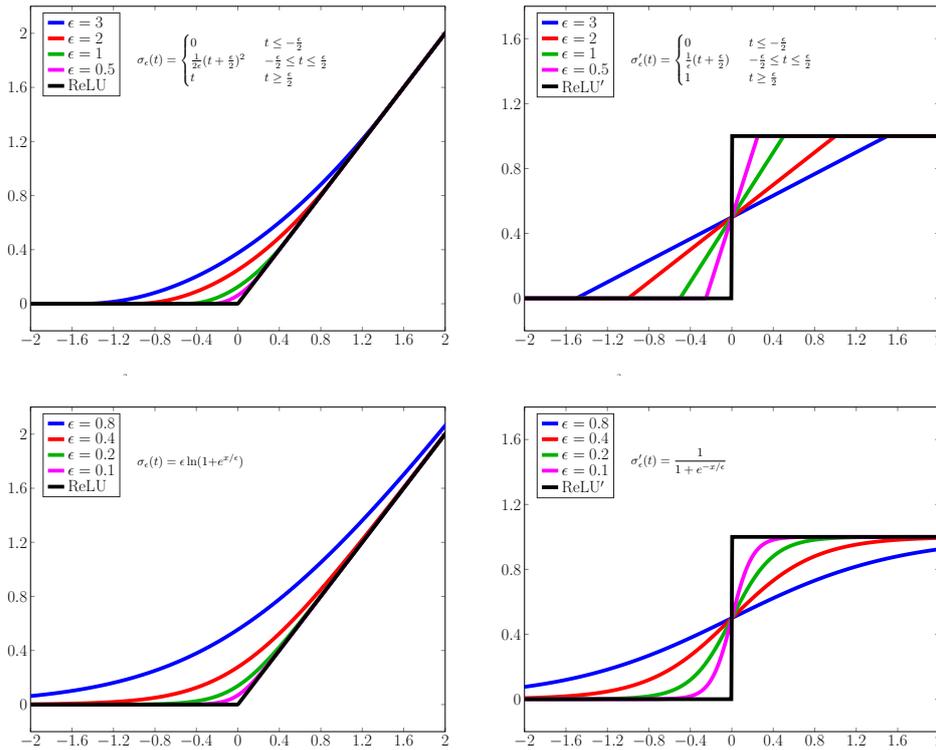


Figure 1: Examples of canonical smoothings of the ReLU functions together with depiction of the corresponding derivative approximations. In particular, the second one is the so-called *Softplus* function, whose derivative is the logistic function - both extensively used in machine learning.

Nevertheless the following holds:

Proposition 2.6. *Let $\mathcal{N}, (\mathcal{N}_\epsilon)_{\epsilon > 0} : \mathbb{R}^{n_0} \rightarrow \mathbb{R}^{n_L}$ be a ReLU network and a canonical smoothing of it. Then it holds:*

$$\|\mathcal{N}_\epsilon - \mathcal{N}\|_\infty \leq M \|\sigma_\epsilon - \sigma\|_\infty, \tag{2.8}$$

where the constant M does not depend on ϵ but only on the parameters of \mathcal{N} . In particular, $\mathcal{N}_\epsilon \rightarrow \mathcal{N}$ uniformly as $\epsilon \rightarrow 0$.

Furthermore, for every $1 \leq p < \infty$ and for every open bounded $U \subset \mathbb{R}^{n_0}$ we have that

$$\|\nabla \mathcal{N}_\epsilon - \nabla \mathcal{N}\|_{L^p(U)} \rightarrow 0, \quad \text{as } \epsilon \rightarrow 0. \tag{2.9}$$

Proof. In order to show (2.8) we will show the result for networks with two hidden layers and then one can proceed via induction. Let $\mathcal{N} = T_3(\sigma(T_2(\sigma(T_1))))$ and $\mathcal{N}_\epsilon = T_3(\sigma_\epsilon(T_2(\sigma_\epsilon(T_1))))$ be

a two hidden layer ReLU network and its corresponding canonical smoothing, in accordance to the formulation (2.4) (without loss of generality let T_3 be linear). Then, setting $N^{(2)}(x) := T_2(\sigma(T_1(x)))$ and $N_\epsilon^{(2)}(x) := T_2(\sigma_\epsilon(T_1(x)))$, we estimate successively for $x \in \mathbb{R}^{n_0}$

$$\begin{aligned} |N_\epsilon^{(2)}(x) - N^{(2)}(x)| &= |T_2(\sigma_\epsilon(T_1(x))) - T_2(\sigma(T_1(x)))| \\ &\leq \|T_2\| \|\sigma_\epsilon(T_1(x)) - \sigma(T_1(x))\| \leq \|T_2\| \|\sigma_\epsilon - \sigma\|_\infty. \end{aligned} \quad (2.10)$$

We then further estimate

$$\begin{aligned} |\mathcal{N}_\epsilon(x) - \mathcal{N}(x)| &= \|T_3\| |(\sigma_\epsilon(N_\epsilon^{(2)}(x)) - \sigma(N^{(2)}(x)))| \\ &\leq \|T_3\| (|\sigma_\epsilon(N_\epsilon^{(2)}(x)) - \sigma_\epsilon(N^{(2)}(x))| + |\sigma_\epsilon(N^{(2)}(x)) - \sigma(N^{(2)}(x))|) \\ &\leq \|T_3\| (\|T_2\| \|\sigma_\epsilon - \sigma\|_\infty + \|\sigma_\epsilon - \sigma\|_\infty) \\ &\leq M \|\sigma_\epsilon - \sigma\|_\infty, \end{aligned}$$

where we employed the mean value theorem for σ_ϵ , using the fact that $0 \leq \sigma'_\epsilon \leq 1$. The induction step follows similarly.

For (2.9), notice first that \mathcal{N} restricted to U (a function that we still denote by \mathcal{N}) belongs to $W^{1,\infty}(U)$, being Lipschitz. In particular $\nabla \mathcal{N} : U \rightarrow \mathbb{R}^{n_L \times n_0}$ is a function in $L^\infty(U)$ and – see [7, Theorem III.1] – for almost every x it is equal to

$$\nabla \mathcal{N}(x) = W_L \cdot \sigma'(N^{(L-1)}(x)) \cdot W_{L-1} \cdot \dots \cdot \sigma'(N^{(1)}(x)) \cdot W_1, \quad (2.11)$$

with $N^{(K)}$ defined as above and $\sigma' := \chi_{(0,\infty)}$ being applied pointwise. Note that, while $\sigma'(N^i(x))$ is an \mathbb{R}^{n_i} -vector, in (2.11) using the same notation we denote the $n_i \times n_i$ diagonal matrix with the same vector in the diagonal. Analogously $\nabla \mathcal{N}_\epsilon \in C(\bar{U})$ where for every x

$$\nabla \mathcal{N}_\epsilon(x) = W_L \cdot \sigma'_\epsilon(N_\epsilon^{(L-1)}(x)) \cdot W_{L-1} \cdot \dots \cdot \sigma'_\epsilon(N_\epsilon^{(1)}(x)) \cdot W_1. \quad (2.12)$$

We check that $\nabla \mathcal{N}_\epsilon \rightarrow \nabla \mathcal{N}$ almost everywhere, and then (2.9) follows by employing the dominated convergence theorem using the fact that $\nabla \mathcal{N}_\epsilon$ is uniformly bounded in $L^\infty(U)$ since $0 \leq \sigma'_\epsilon \leq 1$ for every $\epsilon > 0$. In order to show the almost everywhere pointwise convergence of the gradients, in view of the recursive formulas (2.11) and (2.12), and considering an inductive argument it suffices to show that if $N := (N_1, \dots, N_n) : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is a ReLU network, N_ϵ is a canonical smoothing such that $\nabla N_\epsilon \rightarrow \nabla N$ almost everywhere as $\epsilon \rightarrow 0$ then also

$$\sigma'_\epsilon(N_\epsilon) \nabla N_\epsilon \rightarrow \sigma'(N) \nabla N, \quad \text{a.e. as } \epsilon \rightarrow 0. \quad (2.13)$$

Let \tilde{U} be the set of full measure where $\nabla N_\epsilon \rightarrow \nabla N$ converges pointwise. Fixing an $1 \leq i \leq n$, as a first case, let $x \in \tilde{U}$ be such that $N_i(x) \neq 0$. Then using $N_{i,\epsilon}(x) \rightarrow N_i(x)$ and Lemma 2.5 (ii), we get $\sigma'_\epsilon(N_{i,\epsilon}(x)) \rightarrow \sigma'(N_i(x))$ and hence (2.13) holds for that x and the i -th row. Let now $x \in \tilde{U}$ such that $N_i(x) = 0$. Since N_i is Lipschitz then, see e.g. [14, Theorem 3.3(i)], the set of such x such that $\nabla_i N(x) \neq 0$ has a zero Lebesgue measure, so we can assume that $\nabla_i N(x) = 0$. Then (2.13) for the i -th row follows from the fact that $\nabla_i N_\epsilon(x) \rightarrow \nabla_i N(x) = 0$ and the fact that $0 \leq \sigma'_\epsilon \leq 1$. \square

Remark 2.7. Looking at the examples of Figure 1 one can see that σ_ϵ can be actually chosen such that

$$\|\sigma_\epsilon - \sigma\|_\infty \leq c\epsilon, \quad \text{for every } \epsilon > 0, \quad (2.14)$$

for some constant $c > 0$. Then (2.8) could be written in a stronger form as

$$\|\mathcal{N}_\epsilon - \mathcal{N}\|_\infty \leq M\epsilon, \quad \text{for every } \epsilon > 0. \quad (2.15)$$

Since our focus here is on ReLU learning-informed PDEs, that is, PDEs that contain a ReLU neural network, we are particularly interested in monotonically increasing networks. As we will see later, they will guarantee uniqueness for the corresponding PDE. In particular we are interested in whether the monotonicity of the ReLU networks can be preserved under canonical smoothing, or not. If the latter is the case, then we study to which extent the resulting nonmonotone part can be controlled by the smoothing parameter $\epsilon > 0$. In what follows we will always make a distinction between monotonically increasing and strictly monotonically increasing functions. The following proposition sheds some light on this context.

Proposition 2.8. *The following are true:*

- (i) *There exists a canonical smoothing $\tilde{\sigma}_\epsilon$ of the ReLU function such that for every (strictly) monotonically increasing one-hidden layer ReLU network $\mathcal{N} : \mathbb{R} \rightarrow \mathbb{R}$, its corresponding canonical smoothing \mathcal{N}_ϵ under $\tilde{\sigma}_\epsilon$ is also (strictly) monotone for every $\epsilon > 0$. However preservation of monotonicity of one-hidden layer ReLU networks does not necessarily hold for an arbitrary canonical smoothing.*
- (ii) *The property of the above canonical smoothing $\tilde{\sigma}_\epsilon$ does not hold for ReLU networks with more than one hidden layers. That is, there exists a monotone increasing two-hidden layer ReLU network such that its canonical smoothing \mathcal{N}_ϵ under $\tilde{\sigma}_\epsilon$ is not monotonically increasing for every $\epsilon > 0$.*

Proof. For (i) it suffices to define $\tilde{\sigma}_\epsilon = \rho_\epsilon * \sigma$, where $\rho_\epsilon(t) = \epsilon^{-1} \rho(t/\epsilon)$ and ρ being the standard mollifier,

$$\rho(t) = ce^{-\frac{1}{1-t^2}}.$$

Here $c > 0$ is a constant such that $\int_{\mathbb{R}} \rho dx = 1$. It is easy to check that σ_ϵ is a canonical smoothing. Let \mathcal{N} be an one hidden layer ReLU network, that is

$$\begin{aligned} \mathcal{N}(t) &= b_2 + W_2 \sigma(W_1 t + b_1) \\ &= b_2 + \sum_{i=1}^{n_1} w_2^i \sigma(w_1^i t + b_1^i), \end{aligned}$$

where $W_1 = (w_1^1, \dots, w_1^{n_1})^T$, $W_2 = (w_2^1, \dots, w_2^{n_1})$, $b_1 = (b_1, \dots, b_{n_1})^T$ and $b_2 \in \mathbb{R}$. Then from the linearity of convolution we have

$$\mathcal{N}_\epsilon(t) := b_2 + \sum_{i=1}^{n_1} w_2^i \rho_\epsilon * \sigma(w_1^i t + b_1^i) = \rho_\epsilon * \left(b_2 + \sum_{i=1}^{n_1} w_2^i \sigma(w_1^i t + b_1^i) \right) = \rho_\epsilon * \mathcal{N}(t).$$

Hence if \mathcal{N} is (strictly) monotone then \mathcal{N}_ϵ is (strictly) monotone as well, since it is immediate to check that this convolution preserves (strict) monotonicity.

In order to see that the above property does not hold for an arbitrary canonical smoothing, consider for instance the canonical smoothing of the first example of Figure 1. Let \mathcal{N} be the ReLU neural network defined as

$$\mathcal{N}(t) = \max(\lambda_1 t, 0) + \max(\lambda_2 t, 0) - \max(\lambda_3 t, 0) - \max(\lambda_4 t, 0),$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4 > 0$ and $\lambda_1 + \lambda_2 = \lambda_3 + \lambda_4$. This network has one hidden layer with 4 neurons and obviously, $\mathcal{N} \equiv 0$ and is hence monotone. Given $\epsilon > 0$, we have that for every $t \in [-\epsilon/2\lambda_{\max}, \epsilon/2\lambda_{\max}]$, with $\lambda_{\max} := \max_i \lambda_i$ that

$$\mathcal{N}'_\epsilon(t) = (\lambda_1^2 + \lambda_2^2 - \lambda_3^2 - \lambda_4^2) \frac{t}{\epsilon}. \quad (2.16)$$

Then by simply choosing λ_i such that the specific linear combination of their squares in (2.16) is not zero, we get that the derivative of \mathcal{N}_ϵ changes sign in a small neighbourhood of the origin and thus implies nonmonotonicity.

In order to produce a counterexample for (ii) consider

$$\mathcal{M}(t) = \max(-\max(t, 0), 0) = \sigma(-\sigma(t)),$$

which is a two-hidden layer ReLU neural network realizing again the zero function, and let $\mathcal{M}_\epsilon(t) = \tilde{\sigma}_\epsilon(-\tilde{\sigma}_\epsilon(t))$ denote its canonical smoothing under $\tilde{\sigma}_\epsilon = \rho_\epsilon * \sigma$. Note that

$$(\rho_\epsilon * \sigma)(t) = \begin{cases} 0 & \text{if } t \leq -\epsilon, \\ \frac{1}{\epsilon} \int_{B(t,\epsilon)} \rho\left(\frac{t-s}{\epsilon}\right) \sigma(s) ds & \text{if } -\epsilon < t < \epsilon, \\ t & \text{if } t \geq \epsilon. \end{cases}$$

For the derivative of \mathcal{M}_ϵ , it obviously holds that $\mathcal{M}'_\epsilon(t) = -\tilde{\sigma}'_\epsilon(-\tilde{\sigma}_\epsilon(t))\tilde{\sigma}'_\epsilon(t) \leq 0$. Furthermore, for $t \geq \epsilon$, we have $\mathcal{M}_\epsilon = 0$, while for $t \leq -\epsilon$ we have $\mathcal{M}_\epsilon(t) = \sigma_\epsilon(0) > 0$. Hence \mathcal{M}_ϵ is monotonically decreasing. \square

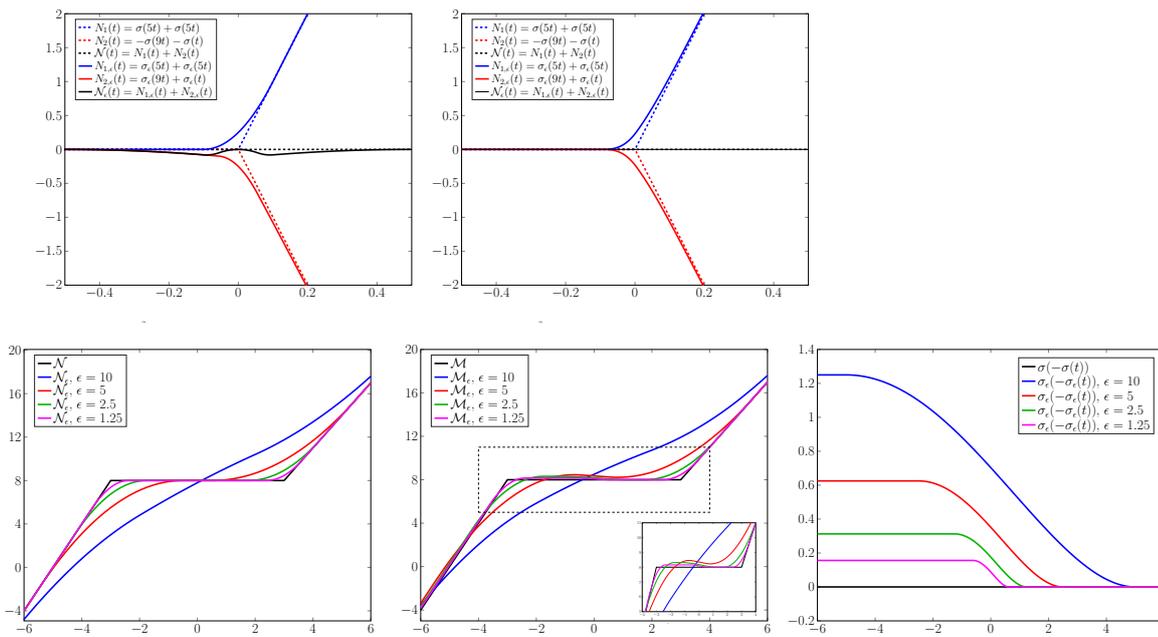


Figure 2: Examples of non preservation of monotonicity of ReLU networks after canonical smoothing as these are dictated by Proposition 2.8.

In Figure 2 we see a visualization of the examples given in Proposition 2.8. At the top part of the figure, we provide an example corresponding to (i) of Proposition 2.8. There, the zero function is written as a one hidden layer ReLU network, $\mathcal{N}(t) = \max(5t) + \max(5t) - \max(9t, 0) - \max(t, 0)$ (dashed black line). Using the canonical smoothing of the first example of Figure 2.8 the monotonicity is not preserved (top left, black solid line). On the other hand the linear smoothing $\tilde{\sigma}_\epsilon = \rho_\epsilon * \sigma$ preserves the monotonicity, see top right plot. At the bottom part of Figure 2 we have expressed the same monotone increasing CPWA function as a ReLU network of both one and two hidden layers, \mathcal{N} and

\mathcal{M} respectively,

$$\begin{aligned}\mathcal{N}(t) &= 4\max(t, 0) - 4\max(-t, 0) + 20 - 4\max(t + 3, 0) + 3\max(t - 3, 0), \\ \mathcal{M}(t) &= \mathcal{N}(t) + \max(-\max(t, 0), 0),\end{aligned}$$

where \mathcal{M} trivially results by adding the zero function $\max(-\max(t, 0), 0)$ to \mathcal{N} . Nevertheless their canonical smoothings \mathcal{N}_ϵ and \mathcal{M}_ϵ under the same σ_ϵ look rather different. We note that here we depict this for the canonical smoothing of the first example of Figure 2.8 but the differences are similar for $\tilde{\sigma}_\epsilon$. The canonical smoothing \mathcal{N}_ϵ of the one hidden layer network remains monotonically increasing (bottom left), which is not the case for the two hidden layer network \mathcal{M}_ϵ (bottom middle). This is due to the term $\max(-\max(t, 0), 0)$ whose canonical smoothing introduces a decreasing part near the origin, see bottom right plot of Figure 2.

Remark 2.9. *Note that even though canonical smoothings do not necessarily preserve monotonicity – in particular as we saw, if \mathcal{N} is increasing, \mathcal{N}_ϵ does not have to be increasing as well – nevertheless the negative part of the derivative $(\mathcal{N}'_\epsilon)^- := \max(-\mathcal{N}'_\epsilon, 0)$ can be controlled. Specifically, according to Proposition 2.6, if $\mathcal{N} : \mathbb{R} \rightarrow \mathbb{R}$ is a monotonically increasing ReLU network – in particular $(\mathcal{N}')^- = 0$, then given an open bounded $U \subset \mathbb{R}$, and $1 \leq p < \infty$, we have for every canonical smoothing \mathcal{N}_ϵ that*

$$\|(\mathcal{N}'_\epsilon)^-\|_{L^p(U)} \rightarrow 0, \quad \text{as } \epsilon \rightarrow 0. \quad (2.17)$$

3 Basic facts of the optimal control problem and implications of smoothing

Recall the main learning-informed optimal control problem:

$$\begin{aligned} & \text{minimize} \quad J(y, u) := \frac{1}{2}\|y - g\|_{L^2(\Omega)}^2 + \frac{\alpha}{2}\|u\|_{L^2(\Omega)}^2, \quad \text{over } (y, u) \in H_0^1(\Omega) \times L^2(\Omega), \\ & \text{subject to} \quad \begin{cases} -\Delta y + \mathcal{N}(\cdot, y) = u, & \text{in } \Omega, \\ y = 0, & \text{on } \partial\Omega, \end{cases} \quad \text{and } u \in \mathcal{C}_{ad}, \end{aligned} \quad (P_{\mathcal{N}})$$

where the different constituents are defined in the introduction. We mention that in [12] a more general setting was adopted by considering a function f instead of \mathcal{N} , belonging to a slightly larger family than the one defined by ReLU neural networks, with the main characteristic that $y \mapsto f(x, y)$ is directionally differentiable. Here, we note that \mathcal{N} is additionally Hadamard directionally differentiable with respect to the second variable. Using the chain rule for Hadamard directionally differentiable functions we can state a recursive formula for $\mathcal{N}'_x(y; h)$, where for every $x \in \mathbb{R}^d$

$$\mathcal{N}'_x(y; h) := \lim_{t_n \rightarrow 0^+} \frac{\mathcal{N}(x, y + t_n h) - \mathcal{N}(x, y)}{t_n}.$$

Indeed, for $z := (x, y)$, and for $N^{(2)}(z) = W_2 \cdot \sigma(W_1 z + b_1) + b_2$, $W_2 \in \mathbb{R}^{1 \times n_1}$, $W_1 \in \mathbb{R}^{n_1 \times (d+1)}$, $b_1 \in \mathbb{R}^{n_1}$, $b_2 \in \mathbb{R}$, we have that for any $y, h \in \mathbb{R}$

$$(N^{(2)})'_x(y; h) = W_2 \cdot \left(\chi_{(0, \infty)}(W_1 z + b_1) W_1(:, n_0) h + \chi_{\{0\}}(W_1 z + b_1) \max(0, W_1(:, n_0) h) \right). \quad (3.1)$$

Here $W_1(:, n_0)$ denotes the last column of W_1 , and $\chi_{(0, \infty)}(W_1 z + b_1)$ is a diagonal matrix, whose diagonal consists of the vector resulting from the componentwise action of the function $\chi_{(0, \infty)}(\cdot)$ on the

vector $W_1 z + b_1$ – similarly for the second summand in (3.1). Recursively for $N^{(\ell)} = W_\ell \sigma(N^{(\ell-1)}(z)) + b_\ell$ we have

$$(N^{(\ell)})'_x(y; h) = W_\ell \cdot (\chi_{(0, \infty)}(N^{(\ell-1)}(z))(N^{(\ell-1)})'_x(y; h) + \chi_{\{0\}}(N^{(\ell-1)}(z)) \max(0, (N^{(\ell-1)})'_x(y; h))). \quad (3.2)$$

Comparing the formulas (3.1)–(3.2) with the formula (2.11) for the weak gradient of \mathcal{N} , one notes that while (2.11) holds almost everywhere, the formulas for the directional derivatives hold at every point.

We will also make use of the function space

$$Y := \{y \in H_0^1(\Omega) : \Delta y \in L^2(\Omega)\},$$

which is a separable Hilbert space equipped with the inner product $(y, v)_Y := \int_\Omega \Delta y \Delta v + \nabla y \nabla v + yv \, dx$ and it is compactly embedded in $H_0^1(\Omega)$. Let N be the Nemytskii operator $y \mapsto N(y)$, with $N(y)(x) = \mathcal{N}(x, y(x))$ for y in some L^p space. Note that $N : L^p(\Omega) \rightarrow L^p(\Omega)$ is Lipschitz continuous for every fixed $1 \leq p \leq \infty$. We also remark, see [12, Section 3.2], that $N : L^p(\Omega) \rightarrow L^p(\Omega)$ for $1 \leq p < \infty$ is Hadamard directionally differentiable with the directional derivative $N'(y; h) \in L^p(\Omega)$ defined via $N'(y; h)(x) = \mathcal{N}'_x(y(x); h(x))$. In the next theorem we briefly summarize the basic results from [12] concerning the optimal control problem $(P_{\mathcal{N}})$.

Theorem 3.1 ([12]). *The following hold for the learning-informed optimal control problem $(P_{\mathcal{N}})$ where we also assume that $p > \frac{d}{2}$ and $p \geq 2$:*

- (i) *For every $u \in L^p(\Omega)$, there exists a unique solution $y \in Y \cap C^{0,a}(\overline{\Omega})$ for the state equation of $(P_{\mathcal{N}})$, where $a > 0$ depends only on p, d and Ω . In particular, for every $M > 0$ there exists a constant c_a (that depends on M) such that*

$$\|y\|_{C^{0,a}(\overline{\Omega})} \leq c_a \|u - \mathcal{N}(\cdot, 0)\|_{L^p(\Omega)}, \quad \text{for all } \|u\|_{L^p(\Omega)} \leq M. \quad (3.3)$$

- (ii) *The control-to-state map $S : L^p(\Omega) \rightarrow Y$ is Hadamard directionally differentiable, and given $u \in L^p(\Omega)$ and a direction $h \in L^p(\Omega)$, $S'(u; h) := z_h \in Y \cap C^{0,a}(\overline{\Omega})$ is the unique solution of*

$$\begin{cases} -\Delta z_h + N'(y; z_h) = h, & \text{in } \Omega, \\ z_h = 0, & \text{on } \partial\Omega, \end{cases} \quad (K)$$

where $y = S(u)$.

- (iii) *The optimal control problem $(P_{\mathcal{N}})$ has a solution.*

- (iv) *(B-stationarity) If $\bar{u} \in L^p(\Omega)$ is a local minimizer for $(P_{\mathcal{N}})$, $\bar{y} = S(\bar{u})$ is the associated state, and $\mathcal{J}(\cdot) = J(S(\cdot), \cdot)$ is the reduced objective for $(P_{\mathcal{N}})$, then the pair (\bar{u}, \bar{y}) satisfies the following variational inequality:*

$$\mathcal{J}'(\bar{u}; h) = \langle \bar{y} - g, S'(\bar{u}; h) \rangle + \alpha \langle \bar{u}, h \rangle \geq 0, \quad \text{for all } h \in T_{\mathcal{C}_{ad}}(\bar{u}). \quad (3.4)$$

Here $T_{\mathcal{C}_{ad}}(\bar{u})$ denotes the contingent cone of \mathcal{C}_{ad} at $\bar{u} \in \mathcal{C}_{ad}$.

- (v) *(C-stationarity) If $\bar{u} \in L^p(\Omega)$ is a local minimizer for $(P_{\mathcal{N}})$, and $\bar{y} = S(\bar{u})$ is the associated state, then the pair (\bar{u}, \bar{y}) satisfies the following optimality system:*

$$\begin{cases} -\Delta \bar{p} + \bar{\zeta} \bar{p} = \bar{y} - g & \text{in } \Omega, \quad \bar{p} = 0 \text{ on } \partial\Omega, \\ \bar{\zeta}(x) \in \partial \mathcal{N}(x, \bar{y}(x)) & \text{for almost every } x \in \Omega, \\ (\bar{p} + \alpha \bar{u}, u - \bar{u}) \geq 0 & \text{for all } u \in \mathcal{C}_{ad}, \end{cases} \quad (3.5)$$

for some nonnegative $\bar{\zeta} \in L^\infty(\Omega)$ and for some adjoint state $\bar{p} \in Y$. Here $\partial\mathcal{N}(x, \bar{y}(x))$ is the Clarke generalized gradient of $\mathcal{N}_x := \mathcal{N}(x, \cdot) : \mathbb{R} \rightarrow \mathbb{R}$ evaluated at $\bar{y}(x)$.

- (vi) (Weak stationarity) We say that $\bar{u} \in L^p(\Omega)$ and $\bar{y} = S(\bar{u})$ satisfy the weak stationarity condition if the first and the third conditions of (3.5) are satisfied for some nonnegative $\bar{\zeta} \in L^\infty(\Omega)$ and for some adjoint state $\bar{p} \in Y$. Obviously any pair (\bar{u}, \bar{y}) of local minimizers for $(P_{\mathcal{N}})$ is weak stationary.

We note that $T_{\mathcal{C}_{ad}}(\bar{u})$ is defined as

$$T_{\mathcal{C}_{ad}}(\bar{u}) := \{h \in L^p(\Omega) : \exists t_n \downarrow 0 \text{ and } h_n \rightarrow h \in L^p(\Omega), \text{ s.t. for all } n \in \mathbb{N}, \bar{u} + t_n h_n \in \mathcal{C}_{ad}\}.$$

Note that it can be shown [8, Lemma 6.34], that if \mathcal{C}_{ad} is of the form

$$\mathcal{C}_{ad} = \{u \in L^p(\Omega) : u_a(x) \leq u(x) \leq u_b(x), \text{ for almost every } x \in \Omega\} \quad (3.6)$$

with $u_a, u_b \in L^\infty(\Omega)$, $u_a < u_b$ almost everywhere, then $T_{\mathcal{C}_{ad}}(\bar{u})$ can be characterized by

$$T_{\mathcal{C}_{ad}}(\bar{u}) = \left\{ h \in L^p(\Omega) : \begin{array}{l} h(x) \geq 0, \text{ almost everywhere in } \{x \in \Omega : \bar{u}(x) = u_a(x)\} \\ h(x) \leq 0, \text{ almost everywhere in } \{x \in \Omega : \bar{u}(x) = u_b(x)\} \end{array} \right\}. \quad (3.7)$$

Apart from the primal notion of B -stationarity, and the primal-dual notions of weak and C -stationarity also one more primal-dual stationarity concept was discussed in [12], namely strong stationarity. There, the relationships between all these concepts were analyzed. Here we focus on B -stationarity, and in particular our developed algorithm studied in Section 4 builds on that notion. We only mention that the C -stationarity system (which is weaker than strong stationarity) is obtained as a limiting optimality system where \mathcal{N} is substituted by some smooth version \mathcal{N}_ϵ and the smoothing parameter ϵ vanishes. In that case the smoothing of the network \mathcal{N} does not need to be canonical as it is only used as a tool in order to get this stationarity system in the limit. Next, we discuss the limitations that arise when this regularization is used not in order to study the limiting case, but in order to solve the corresponding regularized optimal control problem with a classical numerical solver, via smoothing the problem for a fixed $\epsilon > 0$.

Implications of the ReLU smoothing on the uniqueness of the state equation

There are, in general, two levels of approximation involved in the optimal control of learning-informed PDEs. The first level of approximation arises from the approximation of f by a sequence of ReLU neural networks \mathcal{N}_n and can be thought as the capability of the ReLU-informed PDE to approximate some ground truth nonsmooth physical model. This is studied in [12, Proposition 3.3]. The second level of approximation – as we mentioned above – considers the approximating PDEs that arise after smoothing the ReLU network in order to treat the problem algorithmically with classical solvers. As we have mentioned in the introduction, due to the potentially large architecture of a network \mathcal{N} (large number of layers and neurons), a natural and efficient way to smoothen it (after its training has been completed) would be via simply smoothing the ReLU function, with the canonical smoothing procedure described in the previous section. This would result in the following smoothed version of the ReLU learning-informed PDE

$$\begin{cases} -\Delta y + \mathcal{N}_\epsilon(\cdot, y) = u, & \text{in } \Omega, \\ y = 0, & \text{on } \partial\Omega. \end{cases} \quad (E_{\mathcal{N}_\epsilon})$$

However, nonuniqueness issues for the solutions of $(E_{\mathcal{N}_\epsilon})$ can arise, as demonstrated above in Proposition 2.8, since the resulting canonically smoothed network \mathcal{N}_ϵ is not necessarily monotonically increasing. Uniqueness for the solutions of $(E_{\mathcal{N}_\epsilon})$ could be derived by showing that the operator $A_\epsilon : H_0^1(\Omega) \rightarrow H^{-1}(\Omega)$ with

$$\langle A_\epsilon(y), z \rangle_{H^{-1}(\Omega), H_0^1(\Omega)} := \int_{\Omega} \nabla y \nabla z \, dx + \int_{\Omega} \mathcal{N}_\epsilon(x, y) z \, dx,$$

is strongly monotone and then applying the Browder-Minty theorem. This is certainly the case if \mathcal{N}_ϵ was monotone in y , but it could also follow, at least for small $\epsilon > 0$, see [12, Proposition 3.3], if $\nabla \mathcal{N}_\epsilon \rightarrow \nabla \mathcal{N}$ uniformly. However in the case of a canonical smoothing \mathcal{N}_ϵ , the convergence of $\nabla \mathcal{N}_\epsilon$ to $\nabla \mathcal{N}$ as $\epsilon \rightarrow 0$ can only be guaranteed to hold with respect to the L^p norm, for every $1 \leq p < \infty$, see (2.9). The potential nonuniform convergence of $\nabla \mathcal{N}_\epsilon$ to $\nabla \mathcal{N}$ makes the application of the Browder-Minty theorem problematic. In order to be more precise, it would suffice as in the proof of [12, Proposition 3.3], to show that for every $\eta > 0$ there exists $\epsilon_0 > 0$ such that for every $0 < \epsilon < \epsilon_0$

$$\int_{\Omega} (\mathcal{N}_\epsilon(x, y_1) - \mathcal{N}_\epsilon(x, y_2))(y_1 - y_2) \, dx \geq -\eta \|y_1 - y_2\|_{L^2(\Omega)}^2, \quad (3.8)$$

for all $y_1, y_2 \in H_0^1(\Omega)$. Indeed in that case, denoting by c_Ω the Poincaré inequality constant, we would have for every $y_1, y_2 \in H_0^1(\Omega)$

$$\begin{aligned} \langle A_\epsilon(y_1) - A_\epsilon(y_2), y_1 - y_2 \rangle &\geq \frac{1}{(c_\Omega + 1)^2} \|y_1 - y_2\|_{H_0^1}^2 + \int_{\Omega} (\mathcal{N}_\epsilon(x, y_1) - \mathcal{N}_\epsilon(x, y_2))(y_1 - y_2) \, dx \\ &\geq \left(\frac{1}{(c_\Omega + 1)^2} - \eta \right) \|y_1 - y_2\|_{H_0^1}^2, \end{aligned}$$

and thus by choosing $0 < \eta < 1/(c_\Omega + 1)^2$ we would get strong monotonicity for the operator A_ϵ for small enough $\epsilon > 0$. Consider now the example of Figure 2, where for $\mathcal{N}_\epsilon : \mathbb{R} \rightarrow \mathbb{R}$ it holds that there exists a $c > 0$ such that for every $\epsilon > 0$, there exists a $\delta > 0$ such that

$$\nabla \mathcal{N}_\epsilon(t) = \mathcal{N}'_\epsilon(t) < -c, \quad \text{for every } t \in (-\delta, \delta).$$

This means that for every $y_1 < y_2 \in H_0^1(\Omega)$ with values in $(-\delta, \delta)$ a pointwise application of the mean value theorem gives for some θ , with $\theta(x) \in (y_1(x), y_2(x))$

$$\int_{\Omega} (\mathcal{N}_\epsilon(y_1) - \mathcal{N}_\epsilon(y_2))(y_1 - y_2) \, dx = \int_{\Omega} \mathcal{N}'_\epsilon(\theta)(y_1 - y_2)^2 \, dx < -c \|y_1 - y_2\|_{L^2(\Omega)}^2.$$

Hence if $c > 0$ turns out to be large, the absorption of the last term into $\frac{1}{(c_\Omega + 1)^2} \|y_1 - y_2\|_{H_0^1(\Omega)}^2$ is not possible. We note however that one can still prove existence of solutions for the PDEs with nonmonotone nonlinearity for instance by showing that the latter is equivalent to the Euler-Lagrange equation of an associated variational problem, see for instance [11] or by using the theory of type M operators as it is done in the next section. Nevertheless uniqueness can no longer be guaranteed.

Having a (canonical) smoothing \mathcal{N}_ϵ of \mathcal{N} , that satisfies the properties of Proposition 2.6 with the additional property that $\mathcal{N}_\epsilon(x, \cdot)$ is monotonically increasing for every $x \in \Omega$, could be theoretically achieved in two ways: The first way would be to take advantage of the fact that any ReLU network \mathcal{N} of arbitrary number of layers can be realized by a ReLU network of one hidden layer. Then one could use the canonical smoothing derived from convolution $\sigma_\epsilon := \rho_\epsilon * \sigma$ that preserves monotonicity, see (i) of Proposition 2.8. Of course such an approach would not necessarily work in practice in the case one wants to use a classical solver in order to solve a smooth version of $(P_{\mathcal{N}})$, since the one-hidden

layer version of \mathcal{N} cannot be easily derived. The second way, would be to consider abandoning the canonical smoothing approach and smooth directly the multilayer network as $\mathcal{N}_\epsilon := \rho_\epsilon * \mathcal{N}$. While such an approach preserves monotonicity, the computation of a convolution of the network could be computationally demanding and the resulting function cannot necessarily be represented by a neural network. Hence, both approaches appear impractical.

Our discussion here should serve as a warning that using feasible canonical smoothing approaches of \mathcal{N} with the target of solving a smooth approximating problem to $(P_{\mathcal{N}})$ using standard algorithms could be problematic since multiple solutions for the smoothed state equation might be introduced by this process. This provides a further motivation for designing algorithms that directly solve the nonsmooth problem as we do in the following Sections 4 and 5.

4 A descent algorithm for B-stationarity

In this section we introduce a descent algorithm for the ReLU network learning-informed optimal control problem $(P_{\mathcal{N}})$ and discuss its convergence. We recall that \mathcal{N} is assumed monotone in y which gives rise to a unique solution of the learning-informed state equation.

4.1 A descent algorithm

We aim to compute local minimizers for $(P_{\mathcal{N}})$ that satisfy certain stationarity conditions, as outlined in Theorem 3.1. Here we are particularly interested in B -stationarity, i.e., control-state pairs (u, y) that satisfy the following variational inequality:

$$\mathcal{J}'(u; h) = \langle y - g, S'(u; h) \rangle + \alpha \langle u, h \rangle \geq 0, \quad \text{for all } h \in T_{\mathcal{C}_{ad}}(u). \quad (4.1)$$

For the ease of exposition, from now on we focus on the case where \mathcal{C}_{ad} is of the form (3.6) and thus $T_{\mathcal{C}_{ad}}(u)$ can be written as in (3.7).

We proceed in terms of the reduced version of $(P_{\mathcal{N}})$, i.e., by considering the state as dependent on u , i.e., $y = S(u)$, which allows to eliminate the state as an independent variable. Then, given some $u \in \mathcal{C}_{ad}$, following [18] we consider the following auxiliary problem:

$$\text{minimize } \langle S(u) - g, S'(u; h) \rangle + \alpha \langle u, h \rangle \quad \text{over } h \in T_{\mathcal{C}_{ad}}(u). \quad (4.2)$$

Note that according to the definition of B -stationarity (4.1), it holds that $h = 0 \in T_{\mathcal{C}_{ad}}(u)$ is a solution of (4.2) if and only if $(u, S(u))$ is a B -stationary point. We point out that when $(u, S(u))$ is not B -stationary, then problem (4.2) is not necessarily well-posed. As a remedy, we introduce the regularized version

$$\text{minimize } \frac{1}{2}q(h, h) + \langle S(u) - g, S'(u; h) \rangle + \alpha \langle u, h \rangle \quad \text{over } h \in T_{\mathcal{C}_{ad}}(u), \quad (4.3)$$

where $q : L^2(\Omega) \times L^2(\Omega) \rightarrow \mathbb{R}$ is a symmetric functional with $v \mapsto q(v, v)$ convex, differentiable (typically quadratic, hence the notation) and for every $v, v' \in L^2(\Omega)$ satisfying

$$q(v, v) \geq C_1 \|v\|_{L^2(\Omega)}^2 \quad \text{and} \quad q(v, v') \leq C_2 \|v\|_{L^2(\Omega)} \|v'\|_{L^2(\Omega)}, \quad (4.4)$$

for some constants $C_1, C_2 > 0$. Note that according to [18, Lemma 2.1] $h = 0$ is a solution of (4.3) if and only $h = 0$ is a solution of (4.2). Furthermore, the following proposition holds.

Proposition 4.1. *Let $u \in \mathcal{C}_{ad}$ be a feasible point for the reduced version of $(P_{\mathcal{N}})$. Then the following properties are satisfied:*

- (1) *The problem (4.3) admits a solution $\bar{h} \in T_{\mathcal{C}_{ad}}(u)$.*
- (2) *If $\bar{h} \neq 0$, then \bar{h} is a descent direction for the reduced objective \mathcal{J} associated with $(P_{\mathcal{N}})$.*
- (3) *If the directional derivative $S'(u; \cdot) : L^p(\Omega) \rightarrow Y$ is bounded and linear, then \bar{h} is unique.*

Proof. The proof is essentially the same as the one of [18, Proposition 2.3], with the only difference that h is constrained to $T_{\mathcal{C}_{ad}}(u)$ instead of the whole $L^p(\Omega)$. For the first assertion we only need to notice that $T_{\mathcal{C}_{ad}}(u)$ is non-empty, convex and closed due to the assumption that \mathcal{C}_{ad} is non-empty, convex and closed. Then existence of solutions follows from the direct method of the calculus of variations. For the second one, notice that since u is feasible, it follows that $0 \in T_{\mathcal{C}_{ad}}(u)$. Therefore the same argument as [18, Proposition 2.3] can be applied here. The third assertion follows from the strong convexity of the resulting problem. \square

From this discussion it follows that for computing a descent direction for the reduced version of $(P_{\mathcal{N}})$ at a non B -stationary point u , it suffices to solve (4.3). Notice, however, that solving (4.3) is delicate whenever $S'(u; \cdot)$ is not bounded and linear. The latter is typically connected to active nonsmoothness of \mathcal{N} , that is when the set

$$\Omega_{\mathcal{N}}(u) := \{x \in \Omega : \mathcal{N}(x, \cdot) \text{ is nondifferentiable at } y(x) = S(u)(x)\},$$

has a strictly positive Lebesgue measure (which we denote by \mathfrak{m}). In such a situation we will consider a specific approximation of (4.3) as detailed below. Note that $\Omega_{\mathcal{N}}(u)$ is Lebesgue measurable since \mathcal{N} is jointly continuous on $\Omega \times \mathbb{R}$. We mention also that in the case where $\Omega_{\mathcal{N}}(u)$ has zero Lebesgue measure then (4.3) is a standard quadratic problem, presuming q quadratic.

The specific approximation of (4.3) which we utilize in the nonsmooth case consists of a substitution of the nonlinear (and nonsmooth) map $S'(u; \cdot)$ by a differentiable approximation $\Pi_{\epsilon}(u; \cdot)$. More precisely, fixing an $\epsilon > 0$, we define $d_{\epsilon} \in \Pi_{\epsilon}(u; h)$ where d_{ϵ} is a solution of the problem

$$\begin{cases} -\Delta d_{\epsilon} + D_{\epsilon}(y; d_{\epsilon}) = h, & \text{in } \Omega, \\ d_{\epsilon} = 0, & \text{on } \partial\Omega. \end{cases} \quad (4.5)$$

The crucial point here is that D_{ϵ} is the Nemytskii operator that corresponds to a function \mathcal{D}_{ϵ} which is smooth with respect to the second variable but it does not correspond to the derivative \mathcal{N}'_{ϵ} of some smoothing \mathcal{N}_{ϵ} of \mathcal{N} . In order to define \mathcal{D}_{ϵ} we fix a canonical smoothing $(\sigma_{\epsilon})_{\epsilon > 0}$ of the ReLU function such that also (2.14) holds. Then \mathcal{D}_{ϵ} is defined by simply substituting the ReLU (the max function) by σ_{ϵ} whenever this ReLU is applied to the direction d , but leaving the derivative of ReLU intact, wherever that appears in the recursive formulas (3.1)–(3.2) for the directional derivative $\mathcal{N}'_x(y; d)$ of the ReLU network \mathcal{N} . Specifically, for $z := (x, y)$,

$$\begin{aligned} (\mathcal{D}_{\epsilon}^{(2)})_x(y; d) &= W_2 \cdot (\chi_{(0, \infty)}(W_1 z + b_1) W_1(:, n_0) d + \chi_{\{0\}}(W_1 z + b_1) \sigma_{\epsilon}(W_1(:, n_0) d)), \\ (\mathcal{D}_{\epsilon}^{(\ell)})_x(y; d) &= W_{\ell} \cdot (\chi_{(0, \infty)}(N^{(\ell-1)}(z)) (\mathcal{D}_{\epsilon}^{(\ell-1)})_x(y; d) + \chi_{\{0\}}(N^{(\ell-1)}(z)) \sigma_{\epsilon}((\mathcal{D}_{\epsilon}^{(\ell-1)})_x(y; d))), \\ (\mathcal{D}_{\epsilon})_x(y; d) &= (\mathcal{D}_{\epsilon}^{(L)})_x(y; d), \end{aligned}$$

with $\ell = 3, \dots, L$, where L is the number of layers of \mathcal{N} . It is easy to check that the regularity of \mathcal{D}_{ϵ} with respect to d is dictated by the regularity of σ_{ϵ} . For the sake of clarity, we state the formulas

of \mathcal{D}_ϵ for the case of one and two-hidden layer ReLU networks, where also for simplicity, there is no explicit dependence on x , i.e., $\mathcal{N} : \mathbb{R} \rightarrow \mathbb{R}$. For the one-hidden layer case we have for $W_1 = (w_1^1, \dots, w_1^{n_1})^T$, $W_2 = (w_2^1, \dots, w_2^{n_1})$, $b_1 = (b_1, \dots, b_{n_1})^T$, $b_2 \in \mathbb{R}$,

$$\begin{aligned}\mathcal{N}(y) &= b_2 + \sum_{i=1}^{n_1} w_2^i \max(w_1^i y + b_1^i, 0), \\ \mathcal{N}'(y; d) &= \sum_{i=1}^{n_1} w_2^i \left(\chi_{(0, \infty)}(w_1^i y + b_1^i) w_1^i d + \chi_{\{0\}}(w_1^i y + b_1^i) \max(w_1^i d, 0) \right), \\ \mathcal{D}_\epsilon(y; d) &= \sum_{i=1}^{n_1} w_2^i \left(\chi_{(0, \infty)}(w_1^i y + b_1^i) w_1^i d + \chi_{\{0\}}(w_1^i y + b_1^i) \sigma_\epsilon(w_1^i d) \right).\end{aligned}$$

On the other hand for a two-hidden layer case we have, for $W_1 = (w_1^1, \dots, w_1^{n_1})^T$, $W_2 = (w_2^{j,i})_{j,i}$, $i = 1, \dots, n_1$, $j = 1, \dots, n_2$, $W_3 = (w_3^1, \dots, w_3^{n_2})$, $b_1 = (b_1, \dots, b_{n_1})^T$, $b_2 = (b_1, \dots, b_{n_2})^T$, $b_3 \in \mathbb{R}$,

$$\begin{aligned}\mathcal{N}(y) &= b_3 + \sum_{j=1}^{n_2} w_3^j \max \left(\sum_{i=1}^{n_1} w_2^{j,i} \max(w_1^i y + b_1^i, 0) + b_2^j, 0 \right), \\ \mathcal{N}'(y; d) &= \sum_{j=1}^{n_2} w_3^j \chi_{(0, \infty)}(v^j) \left(\sum_{i=1}^{n_1} w_2^{j,i} \left(\chi_{(0, \infty)}(w_1^i y + b_1^i) w_1^i d + \chi_{\{0\}}(w_1^i y + b_1^i) \max(w_1^i d, 0) \right) \right) \\ &\quad + \sum_{j=1}^{n_2} w_3^j \chi_{\{0\}}(v^j) \max \left(\sum_{i=1}^{n_1} w_2^{j,i} \left(\chi_{(0, \infty)}(w_1^i y + b_1^i) w_1^i d + \chi_{\{0\}}(w_1^i y + b_1^i) \max(w_1^i d, 0) \right), 0 \right), \\ \mathcal{D}_\epsilon(y; d) &= \sum_{j=1}^{n_2} w_3^j \chi_{(0, \infty)}(v^j) \left(\sum_{i=1}^{n_1} w_2^{j,i} \left(\chi_{(0, \infty)}(w_1^i y + b_1^i) w_1^i d + \chi_{\{0\}}(w_1^i y + b_1^i) \sigma_\epsilon(w_1^i d) \right) \right) \\ &\quad + \sum_{j=1}^{n_2} w_3^j \chi_{\{0\}}(v^j) \sigma_\epsilon \left(\sum_{i=1}^{n_1} w_2^{j,i} \left(\chi_{(0, \infty)}(w_1^i y + b_1^i) w_1^i d + \chi_{\{0\}}(w_1^i y + b_1^i) \sigma_\epsilon(w_1^i d) \right) \right),\end{aligned}$$

where $v^j = \sum_{i=1}^{n_1} w_2^{j,i} \max(w_1^i y + b_1^i, 0) + b_2^j$.

We have the following approximation result.

Lemma 4.2. *Let $1 \leq p < \infty$, $\mathcal{N} : \Omega \times \mathbb{R} \rightarrow \mathbb{R}$ be a ReLU neural network, $N : L^p(\Omega) \rightarrow L^p(\Omega)$ its corresponding Nemytskii operator, $u \in L^p(\Omega)$, $y = S(u)$ and $N'(y; \cdot) : L^p(\Omega) \rightarrow L^p(\Omega)$ be the directional derivative of N . Then for the operator $L^p(\Omega) \ni d \mapsto D_\epsilon(y; d)$ it holds that*

$$\|N'(y; d) - D_\epsilon(y; d)\|_{L^p(\Omega)} \leq C\epsilon \text{ for all } d \in L^p(\Omega), \quad (4.6)$$

where $C > 0$ is some constant independent of ϵ . In particular $D_\epsilon(y; \cdot) : L^p(\Omega) \rightarrow L^p(\Omega)$.

Proof. The proof is straightforward via induction over the number of layers of \mathcal{N} , using (2.14), and thus we omit the details. \square

Note that in particular for the functions $\mathcal{N}'(y(\cdot); \cdot), \mathcal{D}_\epsilon(y(\cdot); \cdot) : \Omega \times \mathbb{R} \rightarrow \mathbb{R}$ we also have that there exists a constant $C > 0$ such that for every $\epsilon > 0, d \in \mathbb{R}$ and for almost every $x \in \Omega$,

$$|\mathcal{N}'(y(x); d) - \mathcal{D}_\epsilon(y(x); d)| < C\epsilon. \quad (4.7)$$

Using the fact that $y \in L^\infty(\Omega)$, as well as $|\sigma'_\epsilon| \leq 1$, it can also be deduced that $\mathcal{D}_\epsilon(y(x); \cdot)$ is uniformly Lipschitz, i.e., there exists a $c > 0$ such that for every $d_1, d_2 \in \mathbb{R}$, for every $\epsilon > 0$ and almost every $x \in \Omega$

$$|\mathcal{D}_\epsilon(y(x); d_1) - \mathcal{D}_\epsilon(y(x); d_2)| \leq c|d_1 - d_2|. \quad (4.8)$$

In particular this also implies that there exist $a, b > 0$ such that for every $d \in \mathbb{R}$ and almost every $x \in \Omega$

$$|\mathcal{D}_\epsilon(y(x); d)| \leq a + b|d|. \quad (4.9)$$

Remark 4.3. We note that the constant $C > 0$ in (4.7), and hence also the one in (4.6), can be considered to be independent of the state y and as a result also independent of the corresponding control u . Indeed, observe that C is dependent on the L^∞ norm of y , but given the estimate (3.3) and the fact that every u considered here belongs to the box constraint set \mathcal{C}_{ad} of the form (3.6), we have that the L^∞ norm of y is uniformly bounded.

We note that one cannot necessarily expect the functions $\mathcal{D}_\epsilon(y(x); \cdot)$ to be monotone, see the discussion in Section 2.2. Hence the Browder-Minty theorem can no longer be applied, in order to get the existence of a unique solution for the regularized adjoint equation (4.5). Nevertheless existence of solutions can be shown via applying the theory of *type M* operators, see [24]. We recall that if V is a reflexive Banach space, and V^* is its dual, then an operator $\mathcal{A} : V \rightarrow V^*$ is called to be of *type M* whenever it holds that if $d_n \rightharpoonup d, \mathcal{A}d_n \rightharpoonup h$ and $\limsup_n \langle \mathcal{A}d_n, d_n \rangle \leq \langle h, d \rangle$ then it follows that $\mathcal{A}d = h$. The corresponding proposition follows next.

Proposition 4.4. For every $h \in L^2(\Omega)$, the equation (4.5) admits a solution $d_\epsilon \in H_0^1(\Omega)$.

Proof. According to [24, Corollary 2.2] it suffices to show that $\mathcal{A} : H_0^1(\Omega) \rightarrow H^{-1}(\Omega)$ is *type M*, bounded and coercive where for every $d, v \in H_0^1(\Omega)$

$$\mathcal{A}d(v) := \langle \nabla d, \nabla v \rangle + \langle D_\epsilon(y; d), v \rangle. \quad (4.10)$$

Note that the second term on the right-hand side of (4.10) is well-defined due to (4.9). The first term of (4.10) defines a hemicontinuous and monotone operator and hence it is of *type M*, [24, Lemma 2.1]. Thus in order to show that \mathcal{A} is of *type M*, according to [24, Example 2.B] it suffices to show that the operator $\mathcal{B} : H_0^1(\Omega) \rightarrow H^{-1}(\Omega)$

$$\mathcal{B}d(v) := \langle D_\epsilon(y; d), v \rangle$$

is completely continuous, i.e., whenever $d_n \rightharpoonup d$ in $H_0^1(\Omega)$ it holds that $\mathcal{B}d_n \rightarrow \mathcal{B}d$ strongly in $H^{-1}(\Omega)$. Indeed from the compact embedding of $H_0^1(\Omega)$ into $L^2(\Omega)$ we have that $d_n \rightarrow d$ in $L^2(\Omega)$. Using (4.8) we estimate

$$\|D_\epsilon(y; d_n) - D_\epsilon(y; d)\|_{L^2(\Omega)} \leq c\|d_n - d\|_{L^2(\Omega)} \quad (4.11)$$

and thus $D_\epsilon(y; d_n) \rightarrow D_\epsilon(y; d)$ in $L^2(\Omega)$ which implies that $\mathcal{B}d_n \rightarrow \mathcal{B}d$ strongly in $H^{-1}(\Omega)$. Finally, clearly $\mathcal{A} : H_0^1(\Omega) \rightarrow H^{-1}(\Omega)$ is a bounded operator, and also coercive. Indeed, for the latter

property, we have for $d \in H_0^1(\Omega)$ that

$$\begin{aligned} \frac{\mathcal{A}d(d)}{\|d\|_{H_0^1(\Omega)}} &\geq \frac{1}{(c_\Omega + 1)^2} \|d\|_{H_0^1(\Omega)} + \frac{1}{\|d\|_{H_0^1(\Omega)}} \langle N'(y; d), d \rangle + \frac{1}{\|d\|_{H_0^1(\Omega)}} \langle D_\epsilon(y; d) - N'(y; d), d \rangle \\ &\geq \frac{1}{(c_\Omega + 1)^2} \|d\|_{H_0^1(\Omega)} + \frac{1}{\|d\|_{H_0^1(\Omega)}} \underbrace{\langle N'(y; d) - N'(y; 0), d - 0 \rangle}_{\geq 0} - \frac{\tilde{C}\epsilon}{\|d\|_{H_0^1(\Omega)}} \|d\|_{L^2(\Omega)}, \end{aligned}$$

for some constant $\tilde{C} > 0$. Here, c_Ω is the Poincaré constant and we have used the fact that $\mathcal{N}'(y; d)$ is monotonically increasing with respect to d and also (4.7). □

Upon fixing an $\epsilon > 0$, we use a solution of (4.5), denoted by $d_\epsilon = d_\epsilon(h) \in \Pi_\epsilon(u; h)$, to replace $S'(u; h)$ when $\Omega_{\mathcal{N}}(u)$ has positive Lebesgue measure. In particular, (4.3) is approximated by the following problem:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}q(h, h) + \langle S(u) - g, d_\epsilon \rangle + \alpha \langle u, h \rangle \quad \text{over } h \in L^2(\Omega), d_\epsilon \in H_0^1(\Omega) \\ &\text{subject to} \quad \begin{cases} -\Delta d_\epsilon + D_\epsilon(y; d_\epsilon) = h, & \text{in } \Omega, \\ d_\epsilon = 0, & \text{on } \partial\Omega, \end{cases} \quad \text{and } h \in T_{\mathcal{C}_{ad}}(u). \end{aligned} \quad (4.12)$$

Proposition 4.5. *The minimization problem (4.12) has a solution.*

Proof. The first claim is that there exist constants $c_1, c_2 > 0$ independent of h and small $\epsilon > 0$ such that the following estimate holds true

$$\|d_\epsilon\|_{H_0^1(\Omega)} \leq c_1 + c_2 \|h\|_{L^2(\Omega)}, \quad (4.13)$$

from which it straightforwardly follows that the objective in (4.12) is bounded from below and coercive in $L^2(\Omega)$. In order to show (4.13) we add and subtract $N'(y; d_\epsilon)$ in (4.5) and test with d_ϵ getting

$$\begin{aligned} &\|\nabla d_\epsilon\|_{L^2(\Omega)}^2 + \langle D_\epsilon(y; d_\epsilon) - N'(y; d_\epsilon), d_\epsilon \rangle + \underbrace{\langle N'(y; d_\epsilon), d_\epsilon \rangle}_{\geq 0} = \langle h, d_\epsilon \rangle \\ &\Rightarrow \|\nabla d_\epsilon\|_{L^2(\Omega)}^2 - \tilde{C}\epsilon \|d_\epsilon\|_{L^2(\Omega)} \leq \|h\|_{L^2(\Omega)} \|d_\epsilon\|_{L^2(\Omega)}. \end{aligned}$$

By estimating the H_0^1 norm by the L^2 norm using the Poincaré inequality and by dividing by $\|d_\epsilon\|_{H_0^1(\Omega)}$ we have the result. Consider now two minimizing sequences $(h_n)_{n \in \mathbb{N}}$ and $(d_\epsilon^n)_{n \in \mathbb{N}}$. From the coercivity of the objective and from the estimate (4.13) it follows that these are bounded in $L^2(\Omega)$ and $H_0^1(\Omega)$ respectively and hence there exist $h^* \in L^2(\Omega)$ and d_ϵ^* in $H_0^1(\Omega)$ such that $h_n \rightharpoonup h^*$ in $L^2(\Omega)$ and $d_\epsilon^n \rightharpoonup d_\epsilon^*$ in $H_0^1(\Omega)$. Since $T_{\mathcal{C}_{ad}}(u)$ is convex and L^2 -strongly closed it follows that $h^* \in T_{\mathcal{C}_{ad}}(u)$. It remains to show that (h^*, d_ϵ^*) is a feasible pair, i.e., it satisfies (4.5). For this it suffices to show that $D_\epsilon(y; d_\epsilon^n) \rightharpoonup D_\epsilon(y; d_\epsilon^*)$ weakly in $L^2(\Omega)$, which follows similarly as in the proof of Proposition 4.4. The proof is complete in view of the lower semicontinuity of the objective in (4.12) with respect to the corresponding weak convergences. □

In the remainder of this section, we show that for sufficiently small $\epsilon > 0$, we are still able to find a descent direction by solving (4.12) instead of (4.3). We start with the following lemma.

Lemma 4.6. *Let \mathcal{N} be a ReLU neural network, $u, h \in L^p(\Omega)$, $y = S(u)$, $\epsilon > 0$, and let $d = S'(u; h)$, $d_\epsilon \in \Pi_\epsilon(u; h)$ be defined as before. Then the following estimate holds:*

$$\|d_\epsilon - S'(u; h)\|_{H^1(\Omega)} \leq C \|N'(y; d_\epsilon) - D_\epsilon(y; d_\epsilon)\|_{L^2(\Omega)}. \quad (4.14)$$

with a constant $C > 0$ independent of h and ϵ . In particular in view of (4.6) the inequality

$$\|d_\epsilon - S'(u; h)\|_{H^1(\Omega)} \leq C\epsilon \quad (4.15)$$

holds for a generic constant $C > 0$ still independent of h and ϵ .

Proof. We have that d_ϵ, d satisfy

$$-\Delta d_\epsilon + D_\epsilon(y; d_\epsilon) = h, \quad -\Delta d + N'(y; d) = h, \quad \text{in } \Omega, \quad \text{and} \quad d_\epsilon = d = 0 \quad \text{on } \partial\Omega.$$

It follows that $e_\epsilon := d_\epsilon - d$, satisfies

$$-\Delta e_\epsilon + N'(y; d_\epsilon) - N'(y; d) = N'(y; d_\epsilon) - D_\epsilon(y; d_\epsilon) \quad \text{in } \Omega, \quad \text{and} \quad e_\epsilon^h = 0 \quad \text{on } \partial\Omega.$$

Identifying $N'(y; d_\epsilon) - N'(y; d) = \xi(d_\epsilon - d)$ for some $\xi \in L^\infty(\Omega)$, with $\xi \geq 0$ a mean value representation (cf. [12, Proposition 3.1]), and using standard estimates for elliptic PDEs (e.g., [13, Chapter 6, Theorem 2]) we have the conclusion. \square

Remark 4.7. *The estimate in Lemma 4.6 is uniform for every element of the set $\Pi_\epsilon(u; h)$ which is potentially a non-singleton. We also note again that the constants $C > 0$ in (4.14) and (4.15) can also be considered to be independent of y and u . This follows from Remark 4.3 and the fact that the L^∞ norm of ξ above can be upper bounded independently of y (and ϵ), making the constant $C > 0$ in the first estimate (4.14) independent on y (and ϵ).*

Lemma 4.6 indicates that $\Pi_\epsilon(u; h) \rightarrow S'(u; h)$ in $H^1(\Omega)$ as $\epsilon \rightarrow 0$. We note that in order to rigorously state this convergence we would need to define a selection function that chooses a solution of (4.12) for every $\epsilon > 0$. While this can be done using the axiom of choice, or at least the axiom of countable choice, for a sequence $\epsilon_n \rightarrow 0$, we will refrain from using it whenever possible and constrain ourselves to estimates of the type (4.15).

The next proposition shows that for sufficiently small $\epsilon > 0$, we can indeed compute a descent direction by solving (4.12) instead of (4.3).

Proposition 4.8. *Let $u \in \mathcal{C}_{ad}$ be a feasible point for the reduced problem of $(P_{\mathcal{N}})$ which is not B -stationary. Then there exists $\epsilon^* > 0$, such that for $0 < \epsilon < \epsilon^*$ a solution h_ϵ of problem (4.12) is a descent direction for the reduced objective \mathcal{J} of $(P_{\mathcal{N}})$ at u (in particular $h_\epsilon \neq 0$).*

Proof. Our goal is to show that there exist $\epsilon^* > 0$ such that, for all $\epsilon < \epsilon^*$, if h_ϵ solves (4.12), then

$$\langle S(u) - g, S'(u; h_\epsilon) \rangle + \alpha \langle u, h_\epsilon \rangle < 0.$$

Observe first that from the fact that $S'(u; 0) = 0$ and from (4.15), we have that there exists a constant $C > 0$ independent of ϵ such that for every d_ϵ solving (4.5) for $h = 0$, we have $\|d_\epsilon\|_{H^1(\Omega)} \leq C\epsilon$. It follows that if (h_ϵ, d_ϵ) is a solution of (4.12), then we have

$$\frac{1}{2}q(h_\epsilon, h_\epsilon) + \langle S(u) - g, d_\epsilon \rangle + \alpha \langle u, h_\epsilon \rangle \leq C\epsilon. \quad (4.16)$$

again for a constant independent of $\epsilon > 0$. Based on (4.16), we have

$$\langle S(u) - g, S'(u; h_\epsilon) \rangle + \alpha \langle u, h_\epsilon \rangle \leq C\epsilon - \frac{1}{2}q(h_\epsilon, h_\epsilon) + \langle S(u) - g, S'(u; h_\epsilon) - d_\epsilon \rangle. \quad (4.17)$$

Now in view of the estimate (4.15), we have for a generic constant $C > 0$ still independent of $\epsilon > 0$ and u

$$\langle S(u) - g, S'(u; h_\epsilon) \rangle + \alpha \langle u, h_\epsilon \rangle \leq C\epsilon - \frac{1}{2}q(h_\epsilon, h_\epsilon). \quad (4.18)$$

In order to finish the proof it suffices to show that there exists $\epsilon^* > 0$ and $M > 0$ such that for every $\epsilon < \epsilon^*$

$$M \leq q(h_\epsilon, h_\epsilon), \quad (4.19)$$

or in view of the coercivity estimate in (4.4), it suffices to show

$$M \leq \|h_\epsilon\|_{L^2(\Omega)}^2. \quad (4.20)$$

Then by potentially reducing ϵ^* further, the results follows. Suppose towards contradiction that (4.20) does not hold. Then there exists a sequence $\epsilon_n \rightarrow 0$ such that $\|h_{\epsilon_n}\|_{L^2(\Omega)} \rightarrow 0$, which implies that $h_{\epsilon_n} \rightarrow 0$ in $L^2(\Omega)$. Then from Lemma 4.9 below we deduce that $\bar{h} = 0$ is a minimizer of (4.3) which is a contradiction since we have assumed that u is not B -stationary. \square

Lemma 4.9. *Let $u \in \mathcal{C}_{ad}$, $\epsilon_n \rightarrow 0$ and let $h_{\epsilon_n}^*$ be a minimizer for the problem (4.12) for every $n \in \mathbb{N}$. Then there exists a subsequence $(h_{\epsilon_{n_k}}^*)_{k \in \mathbb{N}}$ and a minimizer h^* of (4.3) such that $h_{\epsilon_{n_k}}^* \rightharpoonup h^*$ in $L^2(\Omega)$ as $k \rightarrow \infty$.*

Proof. We first claim that the sequence $(h_{\epsilon_n}^*)_{n \in \mathbb{N}}$ is bounded in $L^2(\Omega)$. This can be seen for instance from (4.15) and the fact that $\frac{1}{2}q(\cdot, \cdot) + \langle S(u) - g, S'(u; \cdot) \rangle$ is coercive. It follows that there exists a subsequence $(h_{\epsilon_{n_k}}^*)_{k \in \mathbb{N}}$ and $h^* \in L^2(\Omega)$ such that $h_{\epsilon_{n_k}}^* \rightharpoonup h^*$ in $L^2(\Omega)$ as $k \rightarrow \infty$. From the estimate (4.13) we can assume that $d_{\epsilon_{n_k}}^* \rightharpoonup d^*$ in $H_0^1(\Omega)$ for some $d^* \in H_0^1(\Omega)$ where $d_{\epsilon_{n_k}}^*$ satisfies (4.5) for $h_{\epsilon_{n_k}}^*$ as right-hand side, also assuming that it has been selected using the axiom of countable choice. Note that we can easily check that $d^* = S'(u; h^*)$, i.e., the pair (h^*, d^*) satisfies the unregularized adjoint equation (K) . Indeed, this follows from the fact that $-\Delta d_{\epsilon_{n_k}}^* \rightharpoonup -\Delta d^*$ and $h_{\epsilon_{n_k}}^* \rightharpoonup h^*$ in $H^{-1}(\Omega)$ and from the fact that $D_{\epsilon_{n_k}}(y; d_{\epsilon_{n_k}}^*) \rightarrow N'(y; d^*)$ in $L^2(\Omega)$. The last convergence can be inferred from the estimate

$$\|D_{\epsilon_{n_k}}(y; d_{\epsilon_{n_k}}^*) - N'(y; d^*)\|_{L^2(\Omega)} \leq \|D_{\epsilon_{n_k}}(y; d_{\epsilon_{n_k}}^*) - N'(y; d_{\epsilon_{n_k}}^*)\|_{L^2(\Omega)} + \|N'(y; d^*) - N'(y; d_{\epsilon_{n_k}}^*)\|_{L^2(\Omega)}$$

in combination with (4.6), the Lipschitz continuity of $N'(y; \cdot)$ and the fact that $d_{\epsilon_{n_k}}^* \rightarrow d^*$ in $L^2(\Omega)$.

Using the minimizing property of $h_{\epsilon_{n_k}}^*$ and letting $G : L^2(\Omega) \times H_0^1(\Omega) \rightarrow \mathbb{R}$ with $G(h, d) = \frac{1}{2}q(h, h) + \langle S(u) - g, d \rangle + \alpha \langle u, h \rangle + \mathcal{X}_{\mathcal{C}_{ad}(u)}(h)$ we have that

$$G(h_{\epsilon_{n_k}}^*, d_{\epsilon_{n_k}}^*) \leq G(h_{\epsilon_{n_k}}, d_{\epsilon_{n_k}}), \quad (4.21)$$

for all pairs $(h_{\epsilon_{n_k}}, d_{\epsilon_{n_k}})$ that satisfy (4.5) for $\epsilon := \epsilon_{n_k}$. We now claim that for every pair (h, d) satisfying (K) there exists a pair sequence $(\bar{h}_{\epsilon_{n_k}}, \bar{d}_{\epsilon_{n_k}})$ that satisfies (4.5) for $\epsilon := \epsilon_{n_k}$ for each index k such that $\bar{h}_{\epsilon_{n_k}} \rightarrow h$ in $L^2(\Omega)$ and $\bar{d}_{\epsilon_{n_k}} \rightarrow d$ in $H_0^1(\Omega)$. Indeed we can set $\bar{h}_{\epsilon_{n_k}} := h$ for all $k \in \mathbb{N}$, and choose $\bar{d}_{\epsilon_{n_k}} \in H_0^1(\Omega)$ a solution of

$$-\Delta \bar{d}_{\epsilon_{n_k}} + D_{\epsilon_{n_k}}(y; \bar{d}_{\epsilon_{n_k}}) = h.$$

Similarly as before we can check that $\bar{d}_{\epsilon_{n_k}} \rightharpoonup d$ in $H_0^1(\Omega)$ where $d = S'(u; h)$. By employing the inequality (4.21) and taking limits on both sides we have

$$G(h^*, d^*) = \liminf_{k \rightarrow \infty} G(h_{\epsilon_{n_k}}^*, d_{\epsilon_{n_k}}^*) \leq \lim_{k \rightarrow \infty} G(\bar{h}_{\epsilon_{n_k}}, \bar{d}_{\epsilon_{n_k}}) = G(h, d).$$

Since (h, d) was a arbitrary pair satisfying (K) , the result follows. \square

Remark 4.10. We note that since the value of the constant $M > 0$ in (4.20) potentially depends on $u \in \mathcal{C}_{ad}$, it cannot be guaranteed that $\epsilon^* > 0$ can be chosen to have a common fixed value for all $u \in \mathcal{C}_{ad}$.

Details on how we solve (4.12) in practice are provided below in Section 4.2. Once (4.12) is solved, and a descent direction h is identified, we perform an Armijo line search in order to compute a step length that sufficiently decreases the reduced objective \mathcal{J} . For the sake of completeness we outline this in Algorithm 1, which assumes that we have already computed $u_k, u_k + h \in \mathcal{C}_{ad}$ at the k -th iteration of the main algorithm. Here $\eta > 0$ is some parameter that prevents the step size from becoming too

Algorithm 1 Armijo line search

Input: $h \in T_{\mathcal{C}_{ad}}(u_k), \tau_0 = \tau > 0, c \in (0, 1), 0 < \eta \ll 1, \nu \in (0, 1), i = 0.$

While

$$\mathcal{J}(u_k + \tau_i h) > \mathcal{J}(u_k) + \nu \tau_i \mathcal{J}'(u_k; h) \quad \text{and} \quad \tau_i > \eta \quad (4.22)$$

Set: $\tau_{i+1} = c\tau_i, i = i + 1$

end while

small.

Note that the directional derivative of the reduced objective $\mathcal{J}'(u_k; h)$ in (4.1), can be evaluated using standard adjoint calculus. The corresponding involved PDEs (the state equation in $(P_{\mathcal{N}})$ and the adjoint equation (K)) are solved numerically via a (semismooth) Newton algorithm.

In practice, the decrease of the step length τ in Algorithm 1 may be faster than the decrease of the magnitude of the descent direction h . This may result in an insufficient decrease of the cost functional \mathcal{J} , particularly when the iterates approach some nonstationary point where the (reduced) objective is nonsmooth. In such a case we perform a *robustification step* similar to [18, Algorithm 4]. That is, we resort to a smoothed optimal control problem in order to compute a new control u_k , and then compute a new descent direction based on this u_k . In particular, we solve the following problem

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|y - g\|_{L^2(\Omega)}^2 + \frac{\alpha}{2} \|u\|_{L^2(\Omega)}^2, \quad \text{over } (y, u) \in H_0^1(\Omega) \times L^2(\Omega), \\ & \text{subject to} \quad \begin{cases} -\Delta y + \mathcal{N}_\delta(\cdot, y) = u, & \text{in } \Omega \\ y = 0, & \text{on } \partial\Omega \end{cases}, \quad \text{and} \quad u \in \mathcal{C}_{ad}. \end{aligned} \quad (4.23)$$

where \mathcal{N}_δ is a (canonically) smoothed version of the network \mathcal{N} . Note that since problem (4.23) is merely a helpful tool in the overall algorithm (in practice the robustification step is rarely activated - see next section), and not the final problem to be solved, the potential nonuniqueness of its solutions is not a point of concern. The numerical solver for this smooth problem can be found for instance in [11]. After every robustification step, we decrease the parameter δ by a factor $\tilde{c} \in (0, 1)$.

We state now in Algorithm 2 the overall descent algorithm which is based on the strategy of sequentially minimizing the cost function in (4.12) in order to obtain descent directions. A few initial remarks on Algorithm 2 are in order. Note that if u_k is not a B -stationary point, then the internal loop which is triggered in Step 3, in the case where h_k is not a descent direction, is finite. This is indeed guaranteed in view of Proposition 4.8. The extra update $\epsilon \rightarrow c_1\epsilon$ in Step 4 after every successful Armijo line search, ensures that the parameter ϵ goes to zero along the iterations.

Algorithm 2 Proposed algorithm for optimal control of ReLU-network-informed PDEs

Input: $u_0 \in \mathcal{C}_{ad}$, $\eta > 0$, $\epsilon = \epsilon_0 > 0$, $\delta = \delta_0 > 0$, $1 \geq \tau > \tau_{\min} > 0$, and $c, \tilde{c}, c_1, c_2, \nu \in (0, 1)$.

Obtain $y_0 = S(u_0)$ by solving the state equation in $(P_{\mathcal{N}})$ using a semismooth Newton method.

Perform the following iteration for $k = 0, 1, 2, \dots$:

Step 1: Solve problem (4.24) at $u = u_k$ in order to get initial values for h_k (details in Section 4.5).

If $\mathfrak{m}(\Omega_{\mathcal{N}}(u_k)) = 0$, go directly to Step 4.

Else go to Step 2.

Step 2: Solve the subproblem (4.12) (details in Sections 4.2 and 4.5) and update h_k accordingly.

Step 3: Check if h_k is a descent direction, i.e., whether

$$\langle S(u_k) - g, S'(u_k; h_k) \rangle + \alpha \langle u, h_k \rangle < 0.$$

If this is not satisfied, update $\epsilon \rightarrow c_1\epsilon$, and return to Step 2.

Step 4: Perform the Armijo line search in Algorithm 1 with parameters c, ν and

$$\eta := \min \left\{ \tau_{\min}, \tilde{c} \|h_k\|_{L^2(\Omega)} \right\}$$

to obtain a step length $\tau \in (0, 1]$, and then update $\epsilon \rightarrow c_1\epsilon$. If $\tau < \eta$, stop the line search, perform the robustification step by solving (4.23) to obtain a new u_k , update $\delta \rightarrow c_2\delta$, let $k = k + 1$ and return to Step 1.

Step 5: Set $u_{k+1} = u_k + \tau h_k$, and compute $S(u_{k+1})$ using again a semismooth Newton method by solving the state equation in $(P_{\mathcal{N}})$. Let $k = k + 1$.

We mention already here that in order to get an initial value for h_k in Step 1, which is used as initialization for Step 2, we solve the following problem

$$\text{minimize } \frac{1}{2}q(h, h) + \langle S(u) - g, \Pi_0(u; h) \rangle + \alpha \langle u, h \rangle \quad \text{over } h \in T_{\mathcal{C}_{ad}}. \quad (4.24)$$

Here, for $h \in L^2(\Omega)$, $\Pi_0(u; h) \in H_0^1(\Omega)$ denotes a solution of the following linear equation:

$$\begin{cases} -\Delta d + D_0(y)d = h, & \text{in } \Omega, \\ d = 0, & \text{on } \partial\Omega, \end{cases} \quad (4.25)$$

where D_0 is the function that results by formally setting the derivatives of the ReLU functions at zero to be zero, recall formula (2.11). Note that in the case $\mathfrak{m}(\Omega_{\mathcal{N}}(u)) = 0$, (4.24) is equivalent to (4.3).

We point the reader to Remark 4.11 below regarding potential (but rare) complications which might be caused by D_0 in (4.25). We also note that if $q(h, h) := \|h\|_{L^2(\Omega)}^2$, then Algorithm 2 will perform exactly like a (sub-) gradient descent method, which can be slow in terms of convergence rates. In order to accelerate the algorithm, in the numerical examples we use the quadratic functional

$$q(h, h) = \langle \Pi_0(u; h), \Pi_0(u; h) \rangle + \alpha \langle h, h \rangle, \quad (4.26)$$

and we denote its derivative at h by Qh . We also note that if the network function $\mathcal{N}(x, y)$ is smooth with respect to y , then the proposed algorithm with the above quadratic functional is an SQP (Sequential Quadratic Programming) type method.

Remark 4.11. *As it was pointed out in [7], even though D_0 is almost equal to the gradient of \mathcal{N} and in particular it is an almost everywhere positive function, its values at the nondifferentiability points of \mathcal{N} could lie strictly below the Clark subdifferential of \mathcal{N} at these points. For example, if $\mathcal{N} : \mathbb{R} \rightarrow \mathbb{R}$, then for every $y \in \mathbb{R}$, it holds that $\partial\mathcal{N}(y) = [\underline{\partial}\mathcal{N}(y), \bar{\partial}\mathcal{N}(y)]$, where $\underline{\partial}\mathcal{N} := \min\{\mathcal{N}'_-, \mathcal{N}'_+\}$, $\bar{\partial}\mathcal{N} := \max\{\mathcal{N}'_-, \mathcal{N}'_+\}$ with \mathcal{N}'_{\pm} denoting the left- and right-sided derivatives. While due to \mathcal{N} being increasing we have $\underline{\partial}\mathcal{N}(y) > 0$ for every $y \in \mathbb{R}$, it could be the case that for some $y_0 \in \mathbb{R}$ it holds $D_0(y_0) < 0 < \underline{\partial}\mathcal{N}(y_0)$ and as a result if the function y in (4.25) attains the value y_0 at a set of positive measure the existence of that equation could be at stake. Since however (4.25) is only used to get some initial values for h_k , in practice, we can restrict ourselves to a nonnegative approximation by setting the negative values of D_0 to zero.*

4.2 Solving problem (4.12)

We continue by providing some details on solving (4.12) in Step 2 of Algorithm 2. From now on we assume that the state equation in (4.12) admits a unique solution $d_\epsilon = d_\epsilon(h_\epsilon)$. This is indeed the case when, e.g., for sufficiently small $\rho > 0$, $\langle D_\epsilon(y; d), d \rangle \geq -\rho \|d\|_{L^2(\Omega)}^2$ for all $d \in H_0^1(\Omega)$. Then the first-order optimality condition for an optimal h_ϵ reads

$$\langle Qh_\epsilon + p_\epsilon(u; h_\epsilon) + \alpha u, h \rangle \geq 0 \quad \text{for all } h \in T_{\mathcal{C}_{ad}}(u). \quad (4.27)$$

Note that in (4.27), $p_\epsilon(u; h_\epsilon)$ represents the directional derivative of the second term in the objective of (4.12).

Since we have assumed box constraints on the control variable (compare (3.6)), and in view of [25, Theorem 2.29], formally the above variational inequality can be equivalently characterized by a system of equations as follows:

$$\begin{cases} -\Delta d_\epsilon + D_\epsilon(y; d_\epsilon) - h_\epsilon = 0 & \text{in } \Omega, \quad \text{and } d_\epsilon = 0 \text{ on } \partial\Omega, \\ -\Delta p_\epsilon + \partial_d(D_\epsilon(y; d_\epsilon))p_\epsilon = y - g & \text{in } \Omega, \quad \text{and } p_\epsilon = 0 \text{ on } \partial\Omega, \\ Qh_\epsilon + p_\epsilon + \mu = -\alpha u, \\ \mu - \max(0, \mu + \lambda(u + h_\epsilon - u_b)) - \min(0, \mu + \lambda(u + h_\epsilon - u_a)) = 0, \end{cases} \quad (4.28)$$

where $\lambda > 0$ is a constant which is typically set equal to the cost of the control, i.e., $\lambda = \alpha$. The first equation in (4.28) is simply (4.5), while the second one is the adjoint equation that provides a way to calculate the directional derivative in (4.27). The third equation represents the first-order stationarity condition of (4.12) with μ being a slack variable, while the fourth one is used to enforce the box constraint $u + h_\epsilon \in \mathcal{C}_{ad}$, complementarity $\mu(u + h_\epsilon - u_a)(u + h_\epsilon - u_b) = 0$ a.e. in Ω , as well as $\mu \geq 0$ a.e. on $\{u + h_\epsilon = u_b\}$ and $\mu \leq 0$ a.e. on $\{u + h_\epsilon = u_a\}$. Under suitable assumptions, the

nonlinear and nonsmooth system (4.28) can be solved efficiently via a primal-dual active-set algorithm (PDAS) for which we provide the details in Section 4.5 below.

The subtle point of the system (4.28) is that additional conditions are required for the existence of solutions for the second equation since the L^∞ -function $\partial_d(D_\epsilon(y; d_\epsilon))$ might be negative on a large set and hence the corresponding PDE operator would not be coercive. This is due to the potential nonmonotonicity of $D_\epsilon(y; \cdot)$. Below we provide a sufficient condition which guarantees existence of solutions and, as a consequence the constraint qualification of [27] is satisfied. Thus, (4.28) indeed represents the Karush-Kuhn-Tucker (KKT) system for (4.12). Note that as we show in Proposition 4.13, essentially unless an early stopping occurs, it holds that $\|h_{\epsilon_k}\|_{L^2(\Omega)} \rightarrow 0$ along the iterations k of Algorithm 2. In view of (4.5), this implies that $\|d_{\epsilon_k}\|_{H_0^1(\Omega)} \rightarrow 0$ as well. The condition below leverages this fact.

Lemma 4.12. *Let $y \in Y \cap C^{0,a}(\overline{\Omega})$ be a solution of the state equation in (P_N) and let $d_\epsilon \in H_0^1(\Omega)$ be a solution of the first equation in (4.28) such that the estimate $\|d_\epsilon\|_{H_0^1(\Omega)} \rightarrow 0$ as $\epsilon \rightarrow 0$. Suppose that there exists $\delta > 0$, possibly dependent on y , such that for small enough $\epsilon > 0$, the set*

$$U = \{x \in \Omega : \mathcal{D}_\epsilon(y(x); \cdot) \text{ is monotone increasing in } (-\delta, \delta)\} \quad (4.29)$$

has a full Lebesgue measure. Then for small enough $\epsilon > 0$ the second equation in (4.28) has a solution $p_\epsilon \in H_0^1(\Omega)$.

Proof. It suffices to show that

$$\mathfrak{m}(\{x \in \Omega : \partial_d D_\epsilon(y(x); d_\epsilon(x)) < 0\}) \rightarrow 0 \quad \text{as } \epsilon \rightarrow 0. \quad (4.30)$$

Indeed, if this holds then the operator $\mathcal{A}_\epsilon : H_0^1(\Omega) \rightarrow H^{-1}(\Omega)$, where for $d, v \in H_0^1(\Omega)$, $\mathcal{A}_\epsilon d(v) := \langle \nabla d, \nabla v \rangle + \langle \partial_d D_\epsilon(y; d_\epsilon) d, v \rangle$ is coercive for small enough $\epsilon > 0$ and we can proceed as in Proposition 4.4.

Since we have $\|d_\epsilon\|_{H_0^1(\Omega)} \rightarrow 0$ as $\epsilon \rightarrow 0$, using the Chebyshev inequality, it follows that

$$\mathfrak{m}(\{x \in \Omega : |d_\epsilon(x)| \geq \delta\}) \rightarrow 0 \quad \text{as } \epsilon \rightarrow 0. \quad (4.31)$$

We then have the partition

$$\begin{aligned} (\{x \in \Omega : \partial_d D_\epsilon(y(x); d_\epsilon(x)) < 0\}) &= (\{x \in \Omega : \partial_d D_\epsilon(y(x); d_\epsilon(x)) < 0\} \cap \{x \in \Omega : |d_\epsilon(x)| < \delta\}) \\ &\quad \cup (\{x \in \Omega : \partial_d D_\epsilon(y(x); d_\epsilon(x)) < 0\} \cap \{x \in \Omega : |d_\epsilon(x)| \geq \delta\}). \end{aligned}$$

The first set in the partition above is a subset of $\Omega \setminus U$ so it has zero Lebesgue measure, while the measure of the second set goes to zero as $\epsilon \rightarrow 0$ in view of (4.31). That shows (4.30). \square

Lemma 4.12 indicates that, in order to have existence of solutions for the second equation in (4.28), it suffices to impose some condition that guarantees that the smoothed function $\mathcal{D}_\epsilon(y(x); \cdot)$ will not be decreasing in an area around zero; in a large set or in set of full measure as it is done here. This is the main region of interest since $\|d_{\epsilon_k}\|_{L^2(\Omega)} \rightarrow 0$ and thus all its values will be essentially concentrated around that area.

4.3 Convergence analysis

In this section, we provide information about the quality of the limits of the sequence of controls $(u_k)_{k \in \mathbb{N}}$ and pertinent states $(y_k)_{k \in \mathbb{N}}$ generated by Algorithm 2. We start with a result regarding the convergence of the sequence of descent directions $(h_k)_{k \in \mathbb{N}}$.

Proposition 4.13. *Let $(u_k)_{k \in \mathbb{N}}$ be a sequence of controls generated by Algorithm 2. If for every $k \in \mathbb{N}$, u_k is not a B -stationary point and the robustification step is activated only finitely many times, then $\|h_k\|_{L^2(\Omega)} \rightarrow 0$ as $k \rightarrow \infty$.*

Proof. Note that since the robustification step is activated finitely many times only, we have that the sequence $(\mathcal{J}(u_k))_{k \in \mathbb{N}}$ is eventually strictly decreasing. Since all its elements are positive, it follows that there exists $\mathcal{J}^* \geq 0$ such that $\mathcal{J}(u_k) \rightarrow \mathcal{J}^*$. Assume without loss of generality, that for all but finitely many iterates we have $\mathfrak{m}(\Omega_{\mathcal{N}}(u_k)) > 0$. Note that from the Armijo line search we have for large enough k , and a constant $C > 0$ independent of ϵ and u_k (see also (4.18))

$$\begin{aligned} \mathcal{J}(u_{k+1}) - \mathcal{J}(u_k) &\leq \nu\tau \mathcal{J}'(u_k; h_k) \leq \nu\tau C\epsilon - \frac{\nu\tau}{2} q(h_k, h_k) \\ &\leq \nu C\epsilon - \frac{\nu C_2}{2} \min(\tau_{min} \|h_k\|_{L^2(\Omega)}^2, \tilde{c} \|h_k\|_{L^2(\Omega)}^3), \end{aligned}$$

where we used the fact that $1 > \tau > \min(\tau_{min}, \tilde{c} \|h_k\|_{L^2(\Omega)})$ and the estimate (4.4). Since $0 > \mathcal{J}(u_{k+1}) - \mathcal{J}(u_k) \rightarrow 0$ as $k \rightarrow \infty$ and the fact that ϵ is also going to zero along the iterates, see the remarks after Algorithm 2, it follows that $\|h_k\|_{L^2(\Omega)} \rightarrow 0$. Lastly if $\mathfrak{m}(\Omega_{\mathcal{N}}(u_k)) = 0$ for infinitely many k 's then, along that subsequence, still denoted by $(u_k)_{k \in \mathbb{N}}$, we have

$$\mathcal{J}(u_{k+1}) - \mathcal{J}(u_k) \leq \nu\tau \mathcal{J}'(u_k; h_k) \leq -\frac{\nu\tau}{2} q(h_k, h_k) \leq -\frac{\nu C_2}{2} \min(\tau_{min} \|h_k\|_{L^2(\Omega)}^2, \tilde{c} \|h_k\|_{L^2(\Omega)}^3),$$

see (4.24) and (4.25). This concludes the proof. \square

The next theorem provides more details about the iterates of Algorithm 2. In fact, depending on properties with respect to robustification and the nonsmooth behavior of \mathcal{N} , along specific subsequences limit points satisfying different types of stationarity are obtained, respectively.

Theorem 4.14. *Let $\lambda = \alpha$ in the KKT system (4.28). Let $(u_k)_{k \in \mathbb{N}}$ be a sequence of controls generated by Algorithm 2, with $(y_k)_{k \in \mathbb{N}}$ the corresponding states. Then the following hold true:*

- (1) *Suppose the algorithm returns $h_{k_0} = 0$ after finitely many iterations, and u_{k_0} and y_{k_0} are the corresponding control and state, respectively. If $\mathfrak{m}(\Omega_{\mathcal{N}}(u_{k_0})) = 0$, then the algorithm returns a B -stationary point; otherwise the following conditions are satisfied:*

$$\begin{aligned} -\Delta y_{k_0} + \mathcal{N}(\cdot, y_{k_0}) - u_{k_0} &= 0 \text{ in } \Omega, \quad y_{k_0} = 0 \text{ on } \partial\Omega, \\ -\Delta p_{k_0} + \chi_\epsilon p_{k_0} - y_{k_0} &= -g \text{ in } \Omega, \quad p_{k_0} = 0 \text{ on } \partial\Omega, \\ (p_{k_0} + \alpha u_{k_0}, h) &\geq 0 \text{ for all } h \in T_{C_{ad}}(u_{k_0}). \end{aligned} \tag{4.32}$$

where $\chi_\epsilon = \partial_d D_\epsilon(y_{k_0}; d_\epsilon)$, and d_ϵ solves the PDE

$$-\Delta d + D_\epsilon(y_{k_0}; d) = 0 \text{ in } \Omega, \quad d = 0 \text{ on } \partial\Omega.$$

- (2) *When the robustification step is activated only finitely many times, the following two cases need to be distinguished:*

- (i) Along a subsequence where $\mathfrak{m}(\Omega_{\mathcal{N}}(u_{k_l})) = 0$ for all $l \in \mathbb{N}$, there exists a further subsequence still denoted by (u_{k_l}, y_{k_l}) , so that $u_{k_l} \rightarrow u^*$ in $L^2(\Omega)$, and $u^* \in \mathcal{C}_{ad}$ satisfies

$$\mathcal{J}^\circ(u^*; h) \geq 0 \text{ for all } h \in T_{\mathcal{C}_{ad}}(u^*), \quad (4.33)$$

where $\mathcal{J}^\circ(u^*; h)$ is the Clarke directional derivative of $\mathcal{J}(\cdot)$ at u^* in the direction h , i.e. $\mathcal{J}^\circ(u^*; h) = \sup_{\xi \in \partial \mathcal{J}(u^*)} \langle \xi, h \rangle$.

- (ii) Along a subsequence where $\mathfrak{m}(\Omega_{\mathcal{N}}(u_{k_l})) > 0$ for all $l \in \mathbb{N}$, there exists a further subsequence still denoted by (u_{k_l}, y_{k_l}) , so that $u_{k_l} \rightarrow u^*$ in $L^2(\Omega)$, and $u^* \in \mathcal{C}_{ad}$ satisfies the weak stationarity condition.

- (3) When the robustification step is activated for infinitely many times, then there exists a subsequence so that the algorithm converges to a \mathcal{C} -stationary point along that subsequence.

Proof. We prove each of the statement here.

- (1) The first statement on the smooth case is due to the setting of the algorithm. In fact, when $\mathfrak{m}(\Omega_{\mathcal{N}}(u_{k_0})) = 0$, then we have $\Pi_0(u_{k_0}; h) = S'(u_{k_0}; h)$. When 0 is a minimizer of (4.2), then for every $h \in T_{\mathcal{C}_{ad}}(u_{k_0})$, $\mathcal{J}'(u_{k_0}; h) \geq \mathcal{J}'(u_{k_0}; 0) = 0$ due to the property of the minimizer. When the nonsmooth part has positive measure, $S'(u_{k_0}; h)$ is replaced by the smooth approximation $\Pi_\epsilon(u_{k_0}; h)$ for some fixed $\epsilon > 0$. The conclusion is drawn by rewriting the KKT system in (4.28) where the equivalence between the third variational inequality in (4.32) and the third and the fourth equations in (4.28) is considered, a proof of which can be found in [25, Theorem 2.29].
- (2) We turn to the first assertion in the second statement. Notice that for a bounded sequence $(u_k)_{k \in \mathbb{N}} \subset \mathcal{C}_{ad} \subset L^2(\Omega)$, we can extract a weakly convergent subsequence denoted by (u_{k_l}) , and $u_{k_l} \rightharpoonup u^* \in \mathcal{C}_{ad}$. Let y_{k_l}, p_{k_l} be the solutions of the state equation and the adjoint equation corresponding to u_{k_l} , respectively, and y^*, p^* be the solutions corresponding to u^* . Using standard regularity results on solutions of elliptical PDEs, we have $y_{k_l} \in H_0^1(\Omega)$ and $p_{k_l} \in H_0^1(\Omega)$ for all $l \in \mathbb{N}$, and (y_{k_l}) and (p_{k_l}) are uniformly bounded in $H_0^1(\Omega)$, respectively. Using the compact embedding of $H_0^1(\Omega)$ into $L^2(\Omega)$, we conclude that $y_{k_l} \rightarrow y^*$ and $p_{k_l} \rightarrow p^*$ both in the $L^2(\Omega)$ norm topology. Referring to the fourth equation in the KKT system (4.28) for each u_{k_l} in the subsequence, we derive also that $u_{k_l} \rightarrow u^*$ strongly in $L^2(\Omega)$ if we choose $\lambda = \alpha > 0$. This is because of $Qh_{k_l} \rightarrow 0$ (as well as $h_{k_l} \rightarrow 0$) and the relation $Qh_{k_l} + p_{k_l} = -\mu_{k_l} - \alpha u_{k_l} \rightarrow p^*$ in $L^2(\Omega)$, and the connection given by the fourth equation in (4.28) when $\lambda = \alpha$, i.e.,

$$\mu_{k_l} = \max(0, \mu_{k_l} + \alpha(u_{k_l} + h_{k_l} - u_b) + \min(0, \mu_{k_l} + \alpha(u_{k_l} + h_{k_l} - u_a)))$$

which ensures that $\mu_{k_l} \rightarrow \mu^*$ in $L^2(\Omega)$, and subsequently $u_{k_l} \rightarrow u^*$ in $L^2(\Omega)$. Since $y_{k_l} \in L^\infty(\Omega)$ and $N(\cdot)$ is Lipschitz, $N'(y_{k_l}) \in L^\infty(\Omega)$ are uniformly bounded. Using the Banach-Alaoglu theorem, we have $N'(y_{k_l}) \overset{*}{\rightharpoonup} \zeta$ for some $\zeta \in L^\infty(\Omega)$. Using the definition of the Clarke subgradient, we have $\zeta \in \partial N(y^*)$ by upper semicontinuity of $\partial N(\cdot)$; see, e.g., [4].

Using the above convergence properties, we arrive at the system

$$\begin{aligned} -\Delta y^* + \mathcal{N}(\cdot, y^*) - u^* &= 0 \text{ in } \Omega, & y^* &= 0 \text{ on } \partial\Omega, \\ -\Delta p^* + \zeta p^* - y^* &= -g \text{ in } \Omega, & p^* &= 0 \text{ on } \partial\Omega, \\ \langle p^* + \alpha u^*, h \rangle &\geq 0 \text{ for all } h \in T_{\mathcal{C}_{ad}}(u^*), \end{aligned} \quad (4.34)$$

where the same argument for the third variational inequality holds as in Case (1). By the definition of Clarke's generalized directional derivative, we then conclude that

$$\mathcal{J}^\circ(u^*; h) \geq \langle p^* + \alpha u^*, h \rangle \geq 0 \text{ for all } h \in T_{\mathcal{C}_{ad}}(u^*).$$

For assertion (ii), we use the same argument as in (i) to have $y_{k_l} \rightarrow y^*$, $p_{k_l} \rightarrow p^*$, and $u_{k_l} \rightarrow u^*$ in $L^2(\Omega)$. Recall that $h_k \rightarrow 0$ in $L^2(\Omega)$ in the KKT system in (4.28). Now we show that $(\partial_d D_{\epsilon_k}(y_k; d_{\epsilon_k}))_{k \in \mathbb{N}}$ is a bounded sequence in $L^\infty(\Omega)$. Note that $N(\cdot)$ is Lipschitz continuous and $y_k \in H^1(\Omega) \cap L^\infty(\Omega)$, and $D_{\epsilon_k}(y_k; \cdot)$ is C^1 smooth and therefore Lipschitz with respect to the second variable, from which we have that $\partial_d D_{\epsilon_k}(\cdot, \cdot)$ is uniformly bounded with respect to both variables, i.e., for all $k \in \mathbb{N}$ we have $|\partial_d D_{\epsilon_k}(y_k, d_{\epsilon_k})| \leq M$. Thus, we have $\partial_d D_{\epsilon_k}(y_k, d_{\epsilon_k}) \in L^\infty(\Omega)$ for all $k \in \mathbb{N}$. Now using the Banach-Alaoglu theorem, we conclude that there exists a weakly star convergent sub-sequence of $\partial_d D_{\epsilon_{k_l}}(y_{k_l}, d_{\epsilon_{k_l}})$, i.e., there exists $\zeta \in L^\infty(\Omega)$ such that $\partial_d D_{\epsilon_{k_l}}(y_{k_l}, d_{\epsilon_{k_l}}) \overset{*}{\rightharpoonup} \zeta$ (still denoted using the same indices). Passing to the limit in the system (4.28) with respect to this subsequence, yields the conclusion.

- (3) For the third statement, we take the subsequence whose elements correspond to the control and state variables for activated robustification. This results in a sequence of optimal control problems with respect to the regularized PDEs in (4.23). Since in the l^{th} robustification step, $\delta_{l+1} = \tilde{c}\delta_l$ for some $\tilde{c} \in (0, 1)$ we infer $\delta_l \rightarrow 0$ as $l \rightarrow \infty$. This yields a C -stationary point in the limit as $l \rightarrow \infty$. For the associated analytical details on the convergence of the smoothed optimal control problems as $\delta_l \rightarrow 0$, we refer to the paper [12].

□

We note that case (1) of Theorem 4.14 rarely occurs in practice and yields a desirable B -stationary point if \mathcal{N} is differentiable at u_{k_0} ; otherwise an approximate version of a C -stationary point is reached. Case (2) either yields a form of C -stationary point in (i), or a an element satisfying weaker conditions in (ii). The latter case produces the least favorable limit point in terms of stationarity. Finally, case (3) provides a point satisfying C -stationarity conditions, which are weaker than B -stationarity conditions.

4.4 Practical aspects concerning Algorithm 2

We recall that for the sake of presentation we confine ourselves to the case where \mathcal{C}_{ad} is given by box constraints; see (3.6). We point out that such box constraints are relevant in numerous applications in PDE constrained optimization.

In order to account for possible violations of the control constraints in the practical numerical realization (e.g. due to inexact solves), we use the following merit function for the line search algorithm

$$E_k(\tau) := \mathcal{J}(u_k + \tau h) + \kappa \Psi(u_k + \tau h), \quad (4.35)$$

where

$$\Psi(u) := \|\max(0, u - u_b)\|_{L^2(\Omega)} + \|\min(0, u - u_a)\|_{L^2(\Omega)},$$

evaluates the violation of the box constraint. Here $\kappa > 0$ is the parameter from Algorithm 2. The above merit function replaces the objective function \mathcal{J} in (4.22) in Algorithm 1. Thus, we need to guarantee a descent direction for (4.35). In our setting the latter is connected to a practical stopping rule for terminating the utilized solver for (4.28). Notice that if $u_k \in \mathcal{C}_{ad}$, and the subproblem (4.12) in

particular the constraint has been settled with satisfactory accuracy, we shall have $u_k + h \in \mathcal{C}_{ad}$ as well. Then $(\Psi(u_k + h) - \Psi(u_k)) = 0$ and the standard Armijo line search is applied. This is often the case when a primal-dual active-set (PDAS) method (see, e.g., [17]) is applied to box constraints as we will explain in detail in the next section. In case one aims at only approximately satisfying the constraint along the iterates, i.e. $(\Psi(u_k + h) - \Psi(u_k)) > 0$, a similar termination condition for the solver of the sub-problem (4.12) as in [11, Algorithm 1, (4.62)] can be applied. It consists of the following inequalities:

$$\begin{aligned} \mathcal{J}'(u_k; h) + \kappa(\Psi(u_k + h) - \Psi(u_k)) &\leq -\xi q(h, h), \\ \text{and } \Psi(u_k + h) &\leq (1 - \xi)\Psi(u_k), \end{aligned} \quad \text{for some } \xi \in (0, 1). \quad (4.36)$$

The first inequality above guarantees a descent direction for the merit functional in every iteration. Whereas the second condition enforces uniform decay of the constraint violation along the iterations. Observe that if $\Psi(u_k) = 0$, then $\Psi(u_l) = 0$ for all $l > k$. The underlying assumption here is that the solver for (4.28) is able to achieve sufficiently accurate solutions.

Notice also that in Algorithm 2, we require $\tau_0 < \tau \leq 1$, and observe further that by solving the system (4.28) exactly we obtain a direction h with $u_k + h \in \mathcal{C}_{ad}$. Hence if $u_k \in \mathcal{C}_{ad}$ and the solution for (4.28) is accurate, then this implies that $u_k + \tau h \in \mathcal{C}_{ad}$ for all $\tau \in (0, 1]$ by convexity of \mathcal{C}_{ad} . Consequently all the iterates are feasible, and the merit functional (4.35) is equivalent to the reduced functional provided that all the systems are solved exactly. Indeed, in our experiments, we use the PDAS algorithm which can compute highly accurate solutions for (4.28).

4.5 Details on the PDAS Algorithm

In the following, we provide some details on the implementation of PDAS in Algorithm 2, as it is employed in two different steps. First we utilize PDAS to solve the KKT system of (4.24) for initialization, which is:

$$\begin{aligned} (K^{-1} + \alpha \text{Id})h + p_0 + \alpha u + \mu &= 0, \\ \mu - \max(0, \mu + \lambda(u + h - u_b)) - \min(0, \mu + \lambda(u + h - u_a)) &= 0, \end{aligned} \quad (4.37)$$

with $\lambda > 0$ fixed. In practice we typically set $\lambda := \alpha > 0$. Here $Qh := (K^{-1} + \alpha \text{Id})h$, with $Q : L^2(\Omega) \rightarrow L^2(\Omega)$ linear and continuous, i.e., $Q \in \mathcal{L}(L^2(\Omega))$, is the derivative of $\frac{1}{2}q(h, h)$ and $K^{-1} \in \mathcal{L}(L^2(\Omega), B)$ for some $B \subset L^2(\Omega)$ with

$$K \in \mathcal{L}(B, L^2(\Omega)), \quad Kt = (-\Delta + (D_0(y)))^*(-\Delta + (D_0(y)))t \text{ for } t \in B.$$

In fact, for given $h \in L^2(\Omega)$ and sufficiently smooth Ω , $Kt = h$ is realized via finding $(s, t) \in (H_0^1(\Omega) \cap H^2(\Omega))^2$ such that

$$-\Delta s + D_0(y)s = h \quad \text{and} \quad -\Delta t + D_0(y)t = s.$$

Thus, $B = H_0^1(\Omega) \cap H^2(\Omega)$.

Computationally, (4.37) is realized as follows: we first introduce an auxiliary variable $t \in B$ and $K^{-1}h = t$. Then, in every iteration of PDAS, we solve the linear system (4.38) below. For this purpose let \mathcal{A}_+ denote an estimate for the upper active set $\{x \in \Omega : u(x) + h^*(x) = u_b(x)\}$, or short $\{u + h^* = u_b\}$, at the solution $h^* \in L^2(\Omega)$ of (4.24) and analogously for \mathcal{A}_- and the lower active set $\{u + h^* = u_a\}$, with $\mathcal{A}_+ \cap \mathcal{A}_- = \emptyset$. Further, $\mathcal{I} = \Omega \setminus \mathcal{A}$, with $\mathcal{A} := \mathcal{A}_+ \cup \mathcal{A}_-$, is an estimate of the inactive set $\{u_a < u + h^* < u_b\}$.

The resulting linear system reads

$$\begin{aligned} Kt - h &= 0, \\ t + \alpha h + \mathcal{J}'_0(u) + \mu &= 0, \\ h|_{\mathcal{A}_+} &= u_b|_{\mathcal{A}_+} - u|_{\mathcal{A}_+}, \quad h|_{\mathcal{A}_-} = u_a|_{\mathcal{A}_-} - u|_{\mathcal{A}_-}, \quad \mu|_{\mathcal{I}} = 0, \end{aligned} \quad (4.38)$$

where $\mathcal{J}'_0(u) = p_0 + \alpha u$, and $v|_{\mathcal{S}}$ denotes the restriction of a function $v : \Omega \rightarrow \mathbb{R}$ to a set $\mathcal{S} \subset \Omega$. Notice that only $h|_{\mathcal{I}}$ and $\mu|_{\mathcal{A}}$ are our desired unknown variables now.

Next, let $E_{\mathcal{I}}$ denote the extension-by-zero operator from \mathcal{I} to Ω . Then, $E_{\mathcal{I}}^*$ is the restriction operator from Ω to \mathcal{I} . The operators $E_{\mathcal{A}_+}$, $E_{\mathcal{A}_+}^*$, $E_{\mathcal{A}_-}$, $E_{\mathcal{A}_-}^*$ and $E_{\mathcal{A}}$, $E_{\mathcal{A}}^*$ are defined analogously. Note that, for instance, $\mu|_{\mathcal{I}} = E_{\mathcal{I}}^* \mu$. For convenience, below we will use both notations for restriction operators. With these definitions and noting from the second and third equation in (4.38) that

$$h|_{\mathcal{I}} = -\alpha^{-1}(t|_{\mathcal{I}} + \mathcal{J}'_0(u)|_{\mathcal{I}}),$$

and

$$\mu|_{\mathcal{A}} = -t|_{\mathcal{A}} - \alpha E_{\mathcal{A}}^*(E_{\mathcal{A}_+}(u_b - u)|_{\mathcal{A}_+} + E_{\mathcal{A}_-}(u_a - u)|_{\mathcal{A}_-}) - \mathcal{J}'_0(u)|_{\mathcal{A}},$$

we can reduce the system in (4.38) to solving

$$(K + \alpha^{-1} E_{\mathcal{I}} E_{\mathcal{I}}^*)t = E_{\mathcal{A}_+}(u_b - u)|_{\mathcal{A}_+} + E_{\mathcal{A}_-}(u_a - u)|_{\mathcal{A}_-} - \alpha^{-1} E_{\mathcal{I}} \mathcal{J}'_0(u)|_{\mathcal{I}}. \quad (4.39)$$

for $t \in B$. Backward substitution then yields $h|_{\mathcal{I}}$ and $\mu|_{\mathcal{A}}$.

Utilizing the above considerations, PDAS solves (4.37) iteratively by estimating the active and inactive sets and solving the associated linear system of the type (4.39) in every iteration. In this context, the active set estimation works as follows: Assume that a current iterate $(h^l, \mu^l) \in L^2(\Omega)^2$, $l \in \mathbb{N}$, is available. Then the next active and inactive set estimates are determined by

$$\begin{aligned} \mathcal{A}_+^{l+1} &:= \{\mu^l + \lambda(u + h^l - u_b) > 0\}, & \mathcal{A}_-^{l+1} &:= \{\mu^l + \lambda(u + h^l - u_a) < 0\}, \\ \mathcal{A}^{l+1} &:= \mathcal{A}_+^{l+1} \cup \mathcal{A}_-^{l+1}, & \mathcal{I}^{l+1} &:= \Omega \setminus \mathcal{A}^{l+1}. \end{aligned}$$

These sets are then used in (4.39), respectively (4.38), to obtain (h^{l+1}, μ^{l+1}) . Unless some stopping rule is satisfied, PDAS returns to the next set estimation. We refer to [17] for more details on PDAS including convergence considerations. The choice of numerical solvers for (4.39) may depend on the size of the system after discretization. In our situation, the standard Matlab backslash is sufficient already. In our tests below, the PDAS iterations are terminated if the $L^2(\Omega)$ -norm residual of the second equation in (4.37) drops below 10^{-16} or a maximum of 50 iterations is reached.

The second application of PDAS is connected to numerically solving the nonlinear system in (4.28). Our strategy here is to decouple the system (4.28) into the following two subsystems:

$$\begin{cases} -\Delta d_\epsilon + D_\epsilon(y; d_\epsilon) - h = 0 & \text{in } \Omega, \quad \text{and } d_\epsilon = 0 \text{ on } \partial\Omega, \\ -\Delta p_\epsilon + \partial_{d_\epsilon}(D_\epsilon(y; d_\epsilon))p_\epsilon = y - g & \text{in } \Omega, \quad \text{and } p_\epsilon = 0 \text{ on } \partial\Omega, \end{cases} \quad (4.40)$$

and

$$\begin{cases} Qh + p_\epsilon + \mu = -\alpha u, \\ \mu - \max(0, \mu + \lambda(u + h - u_b)) - \min(0, \mu + \lambda(u + h - u_a)) = 0. \end{cases} \quad (4.41)$$

Then, in our implementation of Step 2 of Algorithm 2, while the $L^2(\Omega)$ residual norm of the system (4.28) is larger than 10^{-16} , or the iteration count is smaller than 50, we use a consecutive and iterative way to implement the following:

- (i) First we run Newton algorithm for (4.40) to get an update to d_ϵ and p_ϵ for a fixed h . The algorithm can be initialized using the solution from its last round, and 0s for the first round;
- (ii) Using the newly computed d_ϵ and p_ϵ from (i), we apply PDAS to (4.41) to obtain updates of μ and h . The PDAS step is similar to the one we have described above. Only the terms pertinent to the new quantities in (4.40) and (4.41) are adapted. Especially, now Q is associated with the functional $q_\epsilon(h, h) = \langle \Pi_\epsilon(u, \cdot)h, \Pi_\epsilon(u, \cdot)h \rangle + \alpha \langle h, h \rangle$. However, we note here that in our experiments, for each PDAS iteration for (4.41), we found that using simply the quadratic functional q from the initialization step above gives almost the same convergence behavior than using q_ϵ connected to Π_ϵ .

5 Numerical results

In this section, we demonstrate the practical performance of our proposed algorithm for solving optimal control problems with nonsmooth partial differential equations which contain ReLU network components.

Parameter setting of Algorithm 2 In our algorithm, we set $\eta = 10^{-16}$, $\tau_{min} = 10^{-16}$ and $\epsilon_0 = \delta_0 = 10^{-1}$, $c = 0.6$, $c_1 = c_2 = 0.1$, $\beta = 1.1$, $\tilde{c} = 0.5$. The parameter ν will be set depending on the value of α and the respective example. In all the tested examples, we use finite differences for the PDE discretization, and in particular the standard five-point stencil for the discrete Laplacian. The algorithm is terminated if $\|h\|_{L^2(\Omega)} \leq 10^{-16}$. For solving both the state equation and the adjoint equation, we use a (semismooth) Newton method [17], with the stopping rule on checking the $H^{-1}(\Omega)$ -norm of the residual. Specifically, if the residual norm is smaller than 10^{-16} or the number of iterations is bigger than 50, then we stop the Newton solver. Numerical calculations were performed on a laptop with Intel Core i7-10850H CPU and 64GB memory using Matlab R2020b.

5.1 Application to PDE with single max-function

Here we first show the result of our algorithm when applied to an example presented in [9]. We choose $\Omega = (0, 1) \times (0, 1)$ to be the unit square, and design the exact solution and its adjoint state of the optimal control problem to be

$$y = p = \begin{cases} \left((x_1 - \frac{1}{2})^4 + \frac{1}{2}(x_1 - \frac{1}{2})^3 \right) \sin(\pi x_2) & x_1 < \frac{1}{2}, \\ 0 & x_1 \geq \frac{1}{2}. \end{cases} \quad (5.1)$$

No active control constraint is considered in this example for simplicity. The state equation is given by the following second-order semilinear elliptic PDE:

$$-\Delta y + \max(0, y) = u + f \text{ in } \Omega, \quad \text{and } y = 0 \text{ on } \partial\Omega,$$

where, given y and p , the optimal control u and the given function f can also be explicitly calculated using the KKT condition of the optimal control problem. Note that introducing a given function f into the PDE does neither change the analysis nor the algorithm. Both y and p are twice continuously differentiable and have value 0 on the right half of Ω . Therefore, the nonsmoothness of the \max -function in the state equation at the solution appears on a set of positive measure in this example.

This renders the control-to-state map nonsmooth at the solution (u, y) . We test our algorithm by using different discretization sizes dx (uniform in both dimensions) and with respect to variants of the control cost α . Particularly, in all the numerical tests provided in this paper, we consider the following C^2 -smooth approximation of the max-function in \mathcal{D}_ϵ :

$$\sigma_\epsilon(t) = \begin{cases} t - \frac{\epsilon}{2}, & \text{if } t \geq \epsilon, \\ \frac{t^3}{\epsilon^2} - \frac{t^4}{2\epsilon^3}, & \text{if } t \in (0, \epsilon), \\ 0, & \text{if } t \leq 0, \end{cases} \quad (5.2)$$

for given $\epsilon > 0$. The numerical results are reported in Table 5.1. We observe here that our algo-

		$\alpha = 10^{-1}, \nu = 0.9$			$\alpha = 10^{-2}, \nu = 0.9$		
Mesh size	Cost	$\ u - u_h\ / \ u\ $	$\ y - y_h\ / \ y\ $	Cost	$\ u - u_h\ / \ u\ $	$\ y - y_h\ / \ y\ $	
dx=1/16	0.0261	0.0506	0.0234	0.0263	0.044	0.2021	
dx=1/32	0.0329	0.013	0.0058	0.0331	0.0116	0.052	
dx=1/64	0.0368	0.0029	0.0012	0.037	0.003	0.0131	
dx=1/128	0.0389	6.07×10^{-4}	1.03×10^{-4}	0.039	7.1×10^{-4}	0.0031	
		$\alpha = 10^{-3}, \nu = 0.9$			$\alpha = 10^{-4}, \nu = 0.9$		
Mesh size	Cost	$\ u - u_h\ / \ u\ $	$\ y - y_h\ / \ y\ $	Cost	$\ u - u_h\ / \ u\ $	$\ y - y_h\ / \ y\ $	
dx=1/16	0.0281	0.0216	0.7755	0.0455	0.0057	1.3518	
dx=1/32	0.0349	0.0057	0.2003	0.0523	0.0015	0.3501	
dx=1/64	0.0387	0.0015	0.0511	0.0562	3.94×10^{-4}	0.0896	
dx=1/128	0.0408	3.73×10^{-4}	0.0128	0.0582	1.0×10^{-4}	0.0226	
		$\alpha = 10^{-5}, \nu = 0.9$			$\alpha = 10^{-6}, \nu = 0.9$		
Mesh size	Cost	$\ u - u_h\ / \ u\ $	$\ y - y_h\ / \ y\ $	Cost	$\ u - u_h\ / \ u\ $	$\ y - y_h\ / \ y\ $	
dx=1/16	0.2194	0.0011	1.592	1.9592	2.35×10^{-4}	1.6832	
dx=1/32	0.2266	3.08×10^{-4}	0.4136	1.9702	8.32×10^{-5}	0.4421	
dx=1/64	0.2305	8.06×10^{-5}	0.1061	1.9744	2.31×10^{-5}	0.1159	
dx=1/128	0.2326	2.06×10^{-5}	0.027	1.9765	6.06×10^{-6}	0.0319	
		$\alpha = 10^{-7}, \nu = 0.9$			$\alpha = 10^{-8}, \nu = 0.9$		
Mesh size	Cost	$\ u - u_h\ / \ u\ $	$\ y - y_h\ / \ y\ $	Cost	$\ u - u_h\ / \ u\ $	$\ y - y_h\ / \ y\ $	
dx=1/16	19.3566	3.27×10^{-5}	1.7302	193.3311	3.4782×10^{-6}	1.8797	
dx=1/32	19.4059	2.03×10^{-5}	0.4713	193.7626	3.06×10^{-6}	0.6281	
dx=1/64	19.4128	7.43×10^{-6}	0.1330	193.7970	2.09×10^{-6}	0.3067	
dx=1/128	19.4151	2.09×10^{-6}	0.0468	193.8011	7.65×10^{-7}	0.2340	

Table 5.1: Convergence performance of the proposed algorithm (for the single-max function problem) in terms of mesh size dx and the regularization parameter α . The exact solution y is given in (5.1), and $u = p/\alpha$ can be informed through the KKT system as we assumed that the constraint is not active.

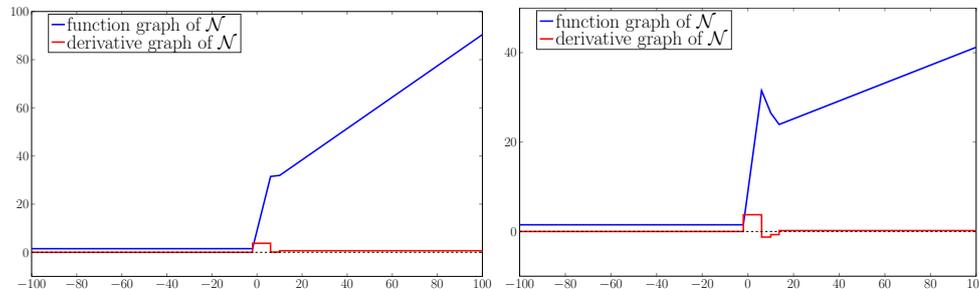
rithm can achieve quadratic convergence rates with respect to the mesh size dx as in [9]. When α becomes smaller, the convergence of the state variable becomes harder. It was reported in [9] that the semismooth Newton type method used there, achieved no convergence when $\alpha = 10^{-6}$ or smaller. However, our method is capable of preserving the quadratic convergence when $\alpha = 10^{-6}$. For the case $\alpha < 10^{-6}$, as provided in the last two groups in Table 5.1, quadratic convergence rate can be observed in the case of $\alpha = 10^{-7}$, and a suboptimal convergence rate appears when $\alpha = 10^{-8}$. This shows that the proposed method is more robust for ill-conditioned problems.

In the following test examples, we also show that our proposed algorithm copes well with semilinear PDEs whose nonlinearities are general ReLU network functions. In this sense, our proposed method can be considered a genuine nonsmooth solver for such type of problems.

5.2 Application to general multilayer ReLU network PDEs

We consider a ReLU neural network function $\mathcal{N} : \mathbb{R} \rightarrow \mathbb{R}$, with two-hidden-layers:

$$\mathcal{N}(y) = \sum_{l=1}^L w_2^l \sigma \left(\sum_{k=1}^K w_1^{l,k} \sigma(w_0^k y + b_0^k) + b_1^l \right) + b_2.$$



Weight parameters						Bias		
w_0^1	w_0^2	w_0^3				b_0^1	b_0^2	b_0^3
5	0.1	10	–	–	–	10	–1	–60
$w_1^{1,1}$	$w_1^{1,2}$	$w_1^{1,3}$	$w_1^{2,1}$	$w_1^{2,2}$	$w_1^{2,3}$	b_1^1	b_1^2	
0.3	2	–0.16	0.1	1	–0.03 (–0.12)	0	1	–
2	–	–	1.5	–	–	0	–	–

Figure 3: Graphs and corresponding weights for the monotone and nonmonotone ReLU networks. The only difference is their $w^{2,3}$ -value (–0.03 vs –0.12).

Our results here address optimal control problems for nonsmooth semilinear elliptic PDEs with both, monotone and nonmonotone network functions, respectively, as shown in Figure 3. For the sake of providing quantitative observations, we generate synthetic data by fixing the solution of the PDE. The data is generated from the function $g_0 = 200 \sin(\pi x) \sin(\pi y)$ and the control $u_0 = \min(u_b, \max(u_a, -\Delta g_0 + \mathcal{N}(g_0)))$ for $u_a = -1000$, $u_b = 1000$, giving rise to a state y_0 . In this example, we choose $\Omega = (0, 2) \times (0, 2)$. Then the function g in the objective of $(P_{\mathcal{N}})$ is computed numerically via the KKT-system for (u_0, y_0) . Both test examples, respectively containing monotone and nonmonotone network functions, are generated in this way. Optimal control of PDEs involving ReLU neural network components, as proposed and studied in this paper, is a new feature in the literature, and our proposed algorithm is specific for the optimization with these type of PDE constraints. For this reason we refrain from comparing our algorithm with other (less tailored) methods for this set of examples.

Our numerical results are summarized in Table 5.2. Here we collect three cases of discretization sizes with respect to varying cost parameter α . In all cases, we observe that $m(\Omega_{\mathcal{N}})$ is not zero at the solution rendering the control-to-state map genuinely nonsmooth. As a consequence, Step 2 in Algorithm 2 is always active. In Table 5.2, 'Cost' denotes the value of the objective functional of the optimal control problem at the final iterate, and 'Iterates' shows the number of outer iterations in Algorithm 2. From the results reported in Table 5.2 we find that in both cases, monotone and nonmonotone, the algorithm exhibits a robust behavior across the scales of $\text{d}x$ and α . Specifically, the almost constant iteration count for varying $\text{d}x$ can be associated with mesh-independent convergence of the algorithm. Moreover, In all cases highly accurate solutions could be obtained. In our computations, we also tested the algorithm in the extreme case of $\alpha = 10^{-16}$, which however exhibits still a similar performance as for the last set of examples for both monotone and nonmonotone functions. The only difference for this case is that we used $\lambda = 10^{-6}$ in (4.37) rather than $\lambda = \alpha$ as in the other cases.

		Monotone case				Nonmonotone case			
		$\alpha = 10^{-2} \nu = 0.7$				$\alpha = 10^{-2} \nu = 0.7$			
Mesh size	Cost	$\ h\ $	Iterates	CPU time	Cost	$\ h\ $	Iterates	CPU time	
dx=1/16	2453.4	2.0×10^{-27}	34	0.5s	2505.8	2.0×10^{-27}	33	0.4s	
dx=1/32	2444.1	3.4×10^{-27}	31	1.5s	2496.1	4.9×10^{-27}	31	1.5s	
dx=1/64	2441.6	1.4×10^{-28}	34	23.9s	2493.7	3.3×10^{-27}	35	39.0s	
		$\alpha = 10^{-10} \nu = 0.7$				$\alpha = 10^{-10} \nu = 0.7$			
Mesh size	Cost	$\ h\ $	Iterates	CPU time	Cost	$\ h\ $	Iterates	CPU time	
dx=1/16	1.4531×10^{-4}	7.9×10^{-24}	34	1.3s	1.4477×10^{-4}	4.2×10^{-24}	34	1.3s	
dx=1/32	1.4535×10^{-4}	5.4×10^{-23}	34	4.7s	1.4474×10^{-4}	1.0×10^{-22}	34	22.2s	
dx=1/64	1.4535×10^{-4}	3.9×10^{-21}	34	23.4s	1.4476×10^{-4}	4.7×10^{-21}	34	23.8s	
		$\alpha = 10^{-16} \nu = 0.7$				$\alpha = 10^{-16} \nu = 0.7$			
Mesh size	Cost	$\ h\ $	Iterates	CPU time	Cost	$\ h\ $	Iterates	CPU time	
dx=1/16	1.4531×10^{-10}	4.9×10^{-17}	55	14.1s	1.4477×10^{-10}	8.4×10^{-17}	54	13.5s	
dx=1/32	1.4535×10^{-10}	8.5×10^{-17}	55	52.2s	1.4474×10^{-10}	8.1×10^{-17}	55	51.6s	
dx=1/64	1.4535×10^{-10}	5.9×10^{-17}	55	273.6s	1.4476×10^{-10}	5.9×10^{-17}	55	283.3s	

Table 5.2: Convergence performance of the proposed algorithm for monotone and nonmonotone ReLU neural network functions.

6 Conclusion

In this paper, we have studied numerical aspects of optimal control problems with ReLU-network-informed PDEs. It was firstly shown that a canonical smoothing of a ReLU network, though practically very plausible, cannot always preserve its monotonicity, something that could imply lack of uniqueness of solutions for the corresponding ReLU-network-informed PDEs. Therefore traditional numerical approaches relying on such smooth approximations may encounter difficulties in the solution process. This motivates us to propose a genuine nonsmooth algorithm which respects the specific structure of ReLU networks in the PDEs. The proposed approach does not smoothen the state equation itself, but it rather approximates the derivative of the control-to-state map via smoothing of the max-function appearing at the directional derivatives. Such approximations were proven to converge strongly to the original directional derivative of the nonsmooth operator in a vanishing smoothing regime. Moreover, this approximation process allows to identify descent directions of the reduced optimal control problem with respect to the nonsmooth PDEs at a given control iterate. In our numerical tests, the proposed algorithm performs more robust in a benchmark optimal control problem when compared to recent nonsmooth algorithms designed specifically for the optimal control of PDEs with a single max-function. In addition, our algorithm also works well for optimal control of semilinear elliptic PDEs with deeper ReLU network functions, which have a more general nonsmooth structure when compared to a single max-function.

Acknowledgments

KP would like to thank Amal Alphonse for useful discussions.

References

- [1] C. Aarset, M. Holler, and T.T.N. Nguyen. Learning-informed parameter identification in nonlinear time-dependent PDEs. *arXiv:2202.10915*, 2022. <https://arxiv.org/abs/2202.10915>.

10915.

- [2] C.D. Aliprantis, D. Harris, and R. Tourky. Continuous piecewise linear functions. *Macroeconomic Dynamics*, 10(1):77–99, 2006. <https://doi.org/10.1017/S1365100506050103>.
- [3] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018. https://openreview.net/pdf?id=B1J_rgWRW.
- [4] J.-P. Aubin and H. Frankowska. *Set-valued analysis*. Springer Science & Business Media, 2009.
- [5] V. Barbu. Optimal control of variational inequalities. *Research Notes in Mathematics*, 100, 1984.
- [6] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- [7] J. Berner, D. Elbrächter, P. Grohs, and A. Jentzen. Towards a regularity theory for ReLU networks – chain rule and global error estimates. In *2019 13th International conference on Sampling Theory and Applications (SampTA)*, pages 1–5, 2019. <https://doi.org/10.1109/SampTA45681.2019.9031005>.
- [8] J.F. Bonnans and A. Shapiro. *Perturbation analysis of optimization problems*. Springer Science & Business Media, 2013.
- [9] C. Christof, C. Meyer, S. Walther, and C. Clason. Optimal control of a non-smooth semilinear elliptic equation. *Mathematical Control & Related Fields*, 8:247, 2018. <https://doi.org/10.3934/mcrf.2018011>.
- [10] G. Dong, M. Hintermüller, and K. Papafitsoros. Quantitative magnetic resonance imaging: From fingerprinting to integrated physics-based models. *SIAM Journal on Imaging Sciences*, 12(2), 2019. <https://doi.org/10.1137/18M1222211>.
- [11] G. Dong, M. Hintermüller, and K. Papafitsoros. Optimization with learning-informed differential equation constraints and its applications. *ESAIM: COCV*, 28:1–44, 2022. <https://doi.org/10.1051/cocv/2021100>.
- [12] G. Dong, M. Hintermüller, K. Papafitsoros, and K. Völkner. First-order conditions for the optimal control of learning-informed nonsmooth PDEs. *arXiv:2206.00297*, 2022. <https://arxiv.org/abs/2206.00297>.
- [13] L.C. Evans. *Partial Differential Equations, volume 19 of Graduate Studies in Mathematics, Second Edition*. American Mathematical Society, 2010.
- [14] L.C. Evans and R.F. Gariepy. *Measure theory and fine properties of functions*. CRC Press, Boca Raton, FL, 1992.
- [15] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.
- [16] I. Gühring, G. Kutyniok, and P. Petersen. Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms. *Analysis and Applications*, 18(05):803–859, 2020. <https://doi.org/10.1142/S0219530519410021>.

- [17] M. Hintermüller, K. Ito, and K. Kunisch. The primal-dual active set strategy as a semismooth Newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2003. <https://doi.org/10.1137/S1052623401383558>.
- [18] M. Hintermüller and T.M. Surowiec. A bundle-free implicit programming approach for a class of elliptic mpecs in function space. *Mathematical Programming*, 160:271–305, 2016. <https://doi.org/10.1007/s10107-016-0983-9>.
- [19] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*, volume 23. Springer Science & Business Media, 2009. <https://doi.org/10.1007/978-1-4020-8839-1>.
- [20] B. Kaltenbacher and T.T.N. Nguyen. Discretization of parameter identification in PDEs using neural networks. *arXiv preprint arXiv:2108.10618*, 2021. <https://arxiv.org/abs/2108.10618>.
- [21] F. Mignot and J.P. Puel. Optimal control in some variational inequalities. *SIAM Journal on Control and Optimization*, 22(3):466–476, 1984. <https://doi.org/10.1137/0322028>.
- [22] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, 2e edition, 2006.
- [23] A. Schiela and D. Wachsmuth. Convergence analysis of smoothing methods for optimal control of stationary variational inequalities with control constraints. *ESAIM: M2AN*, 47(3):771–787, 2013. <https://doi.org/10.1051/m2an/2012049>.
- [24] R.E. Showalter. *Monotone operators in Banach space and nonlinear partial differential equations*, volume 49. American Mathematical Society, 1997.
- [25] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, volume 112 of *Graduate Studies in Mathematics*. American Mathematical Society, 2010.
- [26] S. Wang and X. Sun. Generalization of hinging hyperplanes. *IEEE Transactions on Information Theory*, 51(12):4425–4431, 2005. <https://doi.org/10.1109/TIT.2005.859246>.
- [27] J. Zowe and S. Kurcyusz. Regularity and stability for the mathematical programming problem in Banach spaces. *Applied Mathematics and Optimization*, 5(1):49–62, Mar 1979. <https://doi.org/10.1007/BF01442543>.