A DEEP FIRST-ORDER SYSTEM LEAST SQUARES METHOD FOR THE OBSTACLE PROBLEM

GABRIEL ACOSTA, EUGENIA BELÉN, FRANCISCO M. BERSETCHE,
AND JUAN PABLO BORTHAGARAY

ABSTRACT. We propose a deep learning approach to the obstacle problem inspired by the first-order system least-squares (FOSLS) framework. This method reformulates the problem as a convex minimization task; by simultaneously approximating the solution, gradient, and Lagrange multiplier, our approach provides a flexible, mesh-free alternative that scales efficiently to high-dimensional settings. Key theoretical contributions include the coercivity and local Lipschitz continuity of the proposed least-squares functional, along with convergence guarantees via Γ-convergence theory under mild regularity assumptions. Numerical experiments in dimensions up to 20 demonstrate the method's robustness and scalability, even on non-Lipschitz domains.

1. INTRODUCTION

Variational inequalities arise in the modeling of several nonlinear phenomena such as contact, friction, and plasticity [9, 23, 31]. A classic example is the equilibrium position of an elastic membrane constrained to remain above a given obstacle. These problems are inherently challenging for numerical approaches due to the low regularity of solutions and the a priori unknown contact boundary. While extensive work has been done on numerical methods for such problems, including finite element approaches (see, for example, [10, 17, 34]), the exploration of least-squares formulations remains relatively limited, particularly in contexts beyond finite element methods. We may mention [16] as one of the first works to formalize least-squares formulations for the obstacle problem. The first-order system least-squares (FOSLS) approach proposed in that reference reformulates the obstacle problem by introducing auxiliary variables, such as the Lagrange multiplier and the gradient of the solution. This transforms the obstacle problem into a convex minimization problem subject to linear constraints. The FOSLS framework has demonstrated strong potential for providing stable approximations, facilitating error estimation, and enabling adaptivity.

In parallel, deep learning has emerged as a powerful tool for approximating solutions to partial differential equations (PDEs) by minimizing loss functionals that enforce the governing equations and boundary conditions [15, 18, 26, 27, 30, 32, 36]. While not always competitive with classical methods in low-dimensional settings, neural networks show promise in high-dimensional contexts and for problems with complex geometries [14, 35]. Importantly, neural network formulations based on first-order systems [6, 26, 27, 28] circumvent the need for second-order derivatives in the cost functional, streamlining computation and enabling the approximation of weak solutions.

This paper bridges the gap between the two aforementioned frameworks by proposing a deep learning approach based on the FOSLS methodology for the obstacle problem. Specifically, we adapt the Deep FOSLS framework [6] to the first-order system formulation introduced in [16]. This approach provides a mesh-free alternative to traditional discretization methods that leverages the flexibility and scalability of neural networks. By simultaneously approximating the solution, the gradient, and the Lagrange multiplier, we demonstrate the efficiency and adaptability of this methodology in solving obstacle problems in high-dimensional instances. Moreover, the formulation within the Deep FOSLS framework enables the derivation of convergence results under appropriate regularity assumptions.

1.1. **Related work.** Recent advancements in the literature have explored the approximation of solutions to the obstacle problem through machine learning techniques. In [8], the obstacle problem is reformulated as a penalized minimization problem whose solutions converge to those of the original problem as the penalty parameter approaches infinity. The authors employ neural networks to approximate these minimizers, discretizing the cost functional through Monte Carlo integration. By decomposing the total error into three components—(i) an approximation error (dependent on network depth and width), (ii) a statistical error (governed by sample size), and (iii) an optimization error (related to the empirical loss minimization)—they establish non-asymptotic convergence rates under regularity assumptions on the penalized problem's solutions. In [38], the authors propose two distinct approaches for solving the obstacle problem in one and two dimensions. The first method strictly enforces the obstacle constraint through direct energy minimization, while the second employs a penalized formulation that reduces to solving a linear Dirichlet problem when the constraint is inactive. Both approaches conduct the minimization within a function space parameterized by shallow neural networks. The convergence analysis leverages the universal approximation properties of neural networks under the L^∞-norm. Reference [2] approaches the obstacle problem within a Physics-Informed Neural Networks (PINNs) framework, minimizing the L^2-norm of the PDE residual while strongly enforcing boundary conditions. Similarly, [32] employs an analogous residual-based loss function but incorporates boundary conditions through penalization; this approach is subsequently applied to parabolic obstacle problems arising in American options pricing. Additionally, [12] formulates the obstacle problem as a quadratic programming problem using finite element methods, subsequently solved with a neural network model.

1.2. **Organization of the paper.** The structure of this paper is as follows. Section 2 introduces the reformulation of the obstacle problem as a first-order system. Section 3 outlines the Deep FOSLS approach, detailing the imposition of boundary conditions and the training strategy. In Section 4, we establish the convergence of the method through Γ-convergence theory. Section 5 concludes with numerical experiments in dimensions up to 20, demonstrating the method's scalability and robustness even on non-Lipschitz domains.

2. Variational Formulation and First-Order Reformulation

This section offers a brief theoretical background on the obstacle problem and discusses the theoretical foundation for reformulating it as a first-order system, which a critical step in developing our deep learning methodology. Let $\Omega \subset \mathbb{R}^d$ be an open, bounded domain with Lipschitz boundary, $g \in H_0^1(\Omega)$ and $f \in L^2(\Omega)$. The obstacle problem consists in finding

$$(2.1) \qquad\qquad u_0 := \arg\min_{v \in K} I(v),$$

where I is the classical Dirichlet energy functional

$$I \colon H_0^1(\Omega) \to \mathbb{R}, \quad I(v) = \frac{1}{2} \int_\Omega |\nabla v|^2 - \int_\Omega fv,$$

and K the convex set

$$K := \{v \in H_0^1(\Omega) : \ v \geq g \text{ a.e. in } \Omega\}.$$

It is well-known that this energy-minimization problem admits a unique solution $u_0 \in K$, and that such a solution satisfies the variational inequality

$$(2.2) \qquad (\nabla u_0, \nabla(v - u_0)) \geq (f, v - u_0) \quad \forall v \in K,$$

where the parentheses stand for the duality pairing in $L^2(\Omega)$. Furthermore, in the following we shall denote the $L^2(\Omega)$-norm by $\|\cdot\|$.

In the space $H_0^1(\Omega)$, we have the well-known Poincaré inequality

$$\|v\| \leq C_P \|\nabla v\| \quad \forall v \in H_0^1(\Omega).$$

The constant C_P only depends on the diameter of Ω. This results allows one to obtain the following simple stability bound for solutions to the obstacle problem.

Lemma 2.1 (H^1-stability). *Let $f \in L^2(\Omega)$, $g \in H_0^1(\Omega)$, and $u_0 \in K$ satisfy (2.2). Then,*

$$(2.3) \qquad \|\nabla u_0\| \leq C \left(\|f\| + \|\nabla g\| \right),$$

with a constant $C = C(\Omega)$.

Proof. We use (2.2) with $v = g$:

$$\|\nabla u_0\|^2 \leq (\nabla u_0, \nabla g) + (f, u_0 - g) \leq \frac{1}{2}\|\nabla u_0\|^2 + \frac{1}{2}\|\nabla g\|^2 + \|f\|\|g\| + \|f\|\|u_0\|.$$

For some $\epsilon > 0$ to be chosen, we apply Young's and Poincaré's inequalities to obtain

$$\|f\|\|u_0\| \leq \frac{\|f\|^2}{4\epsilon} + \epsilon\|u_0\|^2 \leq \frac{\|f\|^2}{4\epsilon} + \epsilon C_P^2 \|\nabla u_0\|^2.$$

Therefore, we deduce

$$\left(\frac{1}{2} - \epsilon C_P^2 \right) \|\nabla u_0\|^2 \leq \frac{1}{2}\|\nabla g\|^2 + \|f\|\|g\| + \frac{\|f\|^2}{4\epsilon}.$$

To conclude, it suffices to fix ϵ such that the term in parenthesis in the left hand side is positive and apply the Poincaré's inequality to $g \in H_0^1(\Omega)$. $\square$

Under additional regularity hypotheses (cf. [23]), the energy minimizer $u_0 \in K$ satisfies the *complementarity-form* equation

$$(2.4) \qquad \begin{cases} \min\{-\Delta u_0 - f, \ u_0 - g\} = 0 \text{ in } \Omega, \\ \qquad\qquad\qquad\quad u_0 = 0 \text{ on } \partial\Omega. \end{cases}$$

In this work, following [16], we consider an equivalent reformulation of the obstacle problem as a first-order least squares system. For that purpose, we introduce two auxiliary variables: the flux density

$$\phi_0 := \nabla u_0 \in L^2(\Omega; \mathbb{R}^d),$$

and the Lagrange multiplier

$$\lambda_0 := -\Delta u_0 - f \in H^{-1}(\Omega) := [H_0^1(\Omega)]'.$$

In the following, we employ angle brackets to denote the $H^{-1}(\Omega)$-$H_0^1(\Omega)$ duality pairing. We consider the space

$$\mathcal{V} := \{\boldsymbol{v} = (u, \boldsymbol{\phi}, \lambda) \in H_0^1(\Omega) \times L^2(\Omega; \mathbb{R}^d) \times H^{-1}(\Omega) : \ \operatorname{div} \boldsymbol{\phi} + \lambda \in L^2(\Omega)\},$$

with norm

$$\|(u, \boldsymbol{\phi}, \lambda)\|_{\mathcal{V}} := \left(\|\nabla u\|^2 + \|\boldsymbol{\phi}\|^2 + \|\operatorname{div} \boldsymbol{\phi} + \lambda\|^2\right)^{\frac{1}{2}},$$

introduce a least-squares functional $\mathcal{J} \colon \mathcal{V} \to \mathbb{R}$,

(2.5) $$\mathcal{J}(\boldsymbol{q}; f, g) := \|\operatorname{div} \boldsymbol{\phi} + \lambda + f\|^2 + \|\nabla u - \boldsymbol{\phi}\|^2 + \langle \lambda, u - g \rangle,$$

and the convex set

$$K^{\mathcal{V}} := \{\boldsymbol{v} = (u, \boldsymbol{\phi}, \lambda) \in \mathcal{V} : \ u \geq g, \ \lambda \geq 0\},$$

and study the minimization problem

(2.6) $$\boldsymbol{q}_0 = \arg\min_{\boldsymbol{v} \in K^{\mathcal{V}}} \mathcal{J}(\boldsymbol{v}; f, g).$$

According to [16, Theorem 1], problems (2.1) and (2.6) are equivalent: the unique solution of (2.6) can be characterized as $\boldsymbol{q}_0 = (u_0, \nabla u_0, -\Delta u_0 - f)$, where $u_0 \in H_0^1(\Omega)$ is the unique solution to (2.1).

Even though the functional $\mathcal{J}$ is suitable for the analysis we pursue in this paper, our approach makes use of Monte Carlo integration; we therefore prefer to avoid the duality pairing in $\mathcal{J}$ and replace it with suitable inner products in $L^2(\Omega)$. For the sake of rewriting $\mathcal{J}$, let us consider a triple $\boldsymbol{v} = (u, \boldsymbol{\phi}, \lambda) \in \mathcal{V}$. Because $\gamma := \operatorname{div} \boldsymbol{\phi} + \lambda \in L^2(\Omega)$, we can integrate by parts to rewrite the last term in (2.5) as

$$\langle \lambda, u - g \rangle = \langle \gamma - \operatorname{div} \boldsymbol{\phi}, u - g \rangle = (\gamma, u - g) + (\boldsymbol{\phi}, \nabla(u - g)).$$

Therefore, we introduce the space

$$\mathcal{W} := H_0^1(\Omega) \times L^2(\Omega; \mathbb{R}^d) \times L^2(\Omega)$$

furnished with the product norm

$$\|(u, \boldsymbol{\phi}, \gamma)\|_{\mathcal{W}} := \left(\|\nabla u\|^2 + \|\boldsymbol{\phi}\|^2 + \|\gamma\|^2\right)^{\frac{1}{2}},$$

the convex set

$$K^{\mathcal{W}} := \{\boldsymbol{w} = (u, \boldsymbol{\phi}, \gamma) \in \mathcal{W} : \ u \geq g, \ \gamma \geq \operatorname{div} \boldsymbol{\phi}\},$$

and the functional $\mathcal{L} \colon \mathcal{W} \to \mathbb{R}$

(2.7) $$\mathcal{L}(\boldsymbol{w}; f, g) := \|\gamma + f\|^2 + \|\nabla u - \boldsymbol{\phi}\|^2 + (\gamma, u - g) + (\boldsymbol{\phi}, \nabla(u - g)).$$

We are interested in the minimization problem

(2.8) $$\boldsymbol{p}_0 = \arg\min_{\boldsymbol{w} \in K^{\mathcal{W}}} \mathcal{L}(\boldsymbol{w}; f, g).$$

It is clear that problems (2.6) and (2.8) are equivalent. Indeed, we can define the bijective, norm-preserving mapping

(2.9) $$\mathcal{V} \ni (u, \boldsymbol{\phi}, \lambda) \mapsto (u, \boldsymbol{\phi}, \operatorname{div} \boldsymbol{\phi} + \lambda) \in \mathcal{W};$$

we point out that the restriction of this map to $K^{\mathcal{V}}$ is onto $K^{\mathcal{W}}$. The equality $\mathcal{J}(u, \boldsymbol{\phi}, \lambda) = \mathcal{L}(u, \boldsymbol{\phi}, \operatorname{div} \boldsymbol{\phi} + \lambda)$ follows immediately from the definitions of both functionals. Thus, $\mathcal{L}$ has a

unique minimizer in $K^{\mathcal{W}}$ and it is given by $\boldsymbol{p}_0 := (u_0, \boldsymbol{\phi}_0, \gamma_0)$, where $\boldsymbol{q}_0 = (u_0, \boldsymbol{\phi}_0, \lambda_0)$ is the minimizer of $\mathcal{J}$ in $K^{\mathcal{V}}$, and $\gamma_0 := \lambda_0 + \operatorname{div} \boldsymbol{\phi}_0$. Recalling that $\boldsymbol{q}_0 = (u_0, \nabla u_0, -\Delta u_0 - f)$, we obtain $\boldsymbol{p}_0 = (u_0, \nabla u_0, -f)$.

The following theorem yields the coercivity of $\mathcal{L}$ over $K^{\mathcal{W}}$.

Proposition 2.1 (coercivity). *Let $\boldsymbol{p}_0 \in K^{\mathcal{W}}$ satisfy (2.8). Then, there exists some $C = C(\Omega) > 0$ such that*

$$\mathcal{L}(\boldsymbol{p}; f, g) \geq C\|\boldsymbol{p} - \boldsymbol{p}_0\|_{\mathcal{W}}^2 \quad \forall \boldsymbol{p} \in K^{\mathcal{W}}.$$

Proof. This is a rewriting of [16, Theorem 1] in terms of the admissibility class $K^{\mathcal{W}}$. Indeed, such a result states that if $\boldsymbol{q}_0 \in K^{\mathcal{V}}$ solves (2.6), then

$$\mathcal{J}(\boldsymbol{q}; f, g) \geq C\|\boldsymbol{q} - \boldsymbol{q}_0\|_{\mathcal{V}}^2 \quad \forall \boldsymbol{q} \in K^{\mathcal{V}}.$$

Given $\boldsymbol{p} = (v, \boldsymbol{\tau}, \xi) \in K^{\mathcal{W}}$, we use (2.9) to introduce $\boldsymbol{q} = (v, \boldsymbol{\tau}, \xi - \operatorname{div} \boldsymbol{\tau}) \in K^{\mathcal{V}}$ and deduce

$$\mathcal{L}(\boldsymbol{p}; f, g) = \mathcal{J}(\boldsymbol{q}; f, g) \geq C\|\boldsymbol{q} - \boldsymbol{q}_0\|_{\mathcal{V}}^2 = C\|\boldsymbol{p} - \boldsymbol{p}_0\|_{\mathcal{W}}^2.$$

$\square$

We next prove the local Lipschitz continuity of $\mathcal{L}$ around the minimizer $\boldsymbol{p}_0$.

Proposition 2.2 (local Lipschitz continuity). *Let $\boldsymbol{p}_0 \in K^{\mathcal{W}}$ satisfy (2.8). Then, there exists some $C = C(\Omega, f, g) > 0$ such that*

$$\mathcal{L}(\boldsymbol{p}; f, g) \leq C(1 + \|\boldsymbol{p} - \boldsymbol{p}_0\|_{\mathcal{W}}) \|\boldsymbol{p} - \boldsymbol{p}_0\|_{\mathcal{W}} \quad \forall \boldsymbol{p} \in K^{\mathcal{W}}.$$

Proof. Let $\boldsymbol{p} = (v, \boldsymbol{\tau}, \xi) \in K^{\mathcal{W}}$. Then,

$$\mathcal{L}(\boldsymbol{p}; f, g) = \|\xi + f\|^2 + \|\nabla v - \boldsymbol{\tau}\|^2 + (\xi, v - g) + (\boldsymbol{\tau}, \nabla(v - g)).$$

The unique minimizer of $\mathcal{L}$ in $K^{\mathcal{W}}$ is

$$\boldsymbol{p}_0 := (u_0, \boldsymbol{\phi}_0, \gamma_0) = (u_0, \nabla u_0, -f),$$

where $u_0 \in H_0^1(\Omega)$ is the solution of the obstacle problem, and therefore

$$(-\operatorname{div} \boldsymbol{\phi}_0 + \gamma_0)(u_0 - g) = 0 \quad \text{a.e. in } \Omega.$$

Using these properties, we can write

$$\mathcal{L}(\boldsymbol{p}; f, g) = \|\xi - \gamma_0\|^2 + \|\nabla(v - u_0) - (\boldsymbol{\tau} - \boldsymbol{\phi}_0)\|^2$$
$$+ (\xi, v - g) - (\gamma_0, u_0 - g) + (\operatorname{div} \boldsymbol{\phi}_0, u_0 - g) + (\boldsymbol{\tau}, \nabla(v - g)).$$

On the one hand, it is clear that

$$\|\xi - \gamma_0\|^2 + \|\nabla(v - u_0) - (\boldsymbol{\tau} - \boldsymbol{\phi}_0)\|^2 \leq C\|\boldsymbol{p} - \boldsymbol{p}_0\|_{\mathcal{W}}^2.$$

On the other hand, we can bound

$$|(\xi, v - g) - (\gamma_0, u_0 - g)| = |(\xi, v - u_0) + (\xi - \gamma_0, u_0 - g)| \leq \|\xi\|\|v - u_0\| + \|\xi - \gamma_0\|\|u_0 - g\|$$
$$\leq \|\xi - \gamma_0\|\|v - u_0\| + \|\gamma_0\|\|v - u_0\| + \|\xi - \gamma_0\|\|u_0 - g\|$$
$$\leq \|\boldsymbol{p} - \boldsymbol{p}_0\|_{\mathcal{W}}^2 + (\|\gamma_0\| + \|u_0 - g\|) \|\boldsymbol{p} - \boldsymbol{p}_0\|_{\mathcal{W}},$$

and similarly

$$|(\operatorname{div} \boldsymbol{\phi}_0, u_0 - g) + (\boldsymbol{\tau}, \nabla(v - g))| \leq \|\boldsymbol{p} - \boldsymbol{p}_0\|_{\mathcal{W}}^2 + (\|\boldsymbol{\phi}_0\| + \|u_0 - g\|) \|\boldsymbol{p} - \boldsymbol{p}_0\|_{\mathcal{W}}.$$

We recall $\gamma_0 = -f$ and $\phi_0 = -\nabla u_0$. Because $u_0 \in H_0^1(\Omega)$, we can collect the estimates above and apply the Poincaré inequality together with (2.3) to conclude the desired result. $\square$

3. Description of the method

We propose a method to approximate the unique minimizer (u_0, ϕ_0, γ_0) of the functional in (2.7) by using neural networks. A natural, direct approach involves seeking a set of parameters $\boldsymbol{\Theta}_0 \in \mathbb{R}^m$ such that

$$\mathcal{L}(u_{\boldsymbol{\Theta}_0}, \phi_{\boldsymbol{\Theta}_0}, \gamma_{\boldsymbol{\Theta}_0}) = \min_{\boldsymbol{\Theta} \in \mathbb{R}^m} \mathcal{L}(u_{\boldsymbol{\Theta}}, \phi_{\boldsymbol{\Theta}}, \gamma_{\boldsymbol{\Theta}}),$$

with functions $(u_{\boldsymbol{\Theta}}, \phi_{\boldsymbol{\Theta}}, \gamma_{\boldsymbol{\Theta}})$ belonging to a suitable neural network space. However, a critical challenge arises: the neural network approximations $(u_{\boldsymbol{\Theta}}, \phi_{\boldsymbol{\Theta}}, \gamma_{\boldsymbol{\Theta}})$ are not guaranteed a priori to satisfy the admissibility constraints $u_{\boldsymbol{\Theta}} \geq g$, $\gamma_{\boldsymbol{\Theta}} \geq \operatorname{div} \phi_{\boldsymbol{\Theta}}$ and the homogeneous Dirichlet boundary condition on $u_{\boldsymbol{\Theta}}$. It is a common practice in neural network approaches to weakly impose such constraints through penalty terms added to the loss functional. In contrast, our method enforces these conditions strongly by design, ensuring strict adherence throughout the optimization process. For boundary conditions, we adopt the strategy proposed in [5], where pre-training the network to explicitly satisfy boundary data reduces the number of iterations needed for convergence.

As discussed in Section 3.1 below, we initially create certain auxiliary functions to ensure $(u_{\boldsymbol{\Theta}}, \phi_{\boldsymbol{\Theta}}, \gamma_{\boldsymbol{\Theta}}) \in K^{\mathcal{W}}$ for all $\boldsymbol{\Theta} \in \mathbb{R}^m$. The optimization procedure involves sampling N points $\{x_k\}_{k=1}^N \subset \Omega$ and approximating

$$\mathcal{L}(u_{\boldsymbol{\Theta}}, \phi_{\boldsymbol{\Theta}}, \gamma_{\boldsymbol{\Theta}}) \approx L_N(u_{\boldsymbol{\Theta}}, \phi_{\boldsymbol{\Theta}}, \gamma_{\boldsymbol{\Theta}}),$$

at each step of a gradient descent algorithm, where L_N is defined as

(3.1)
$$L_N(u, \phi, \gamma) := \frac{|\Omega|}{N} \sum_{k=1}^N \left(\gamma + f(x_k)\right)^2 + \left|\nabla u(x_k) - \phi(x_k)\right|^2$$
$$+ \gamma(x_k)\left(u(x_k) - g(x_k)\right) + \phi(x_k) \cdot \left(\nabla u(x_k) - \nabla g(x_k)\right).$$

We expose the details below.

3.1. **Strong imposition of boundary and admissibility conditions.** We follow ideas from [5, 6, 33] about the imposition of the boundary conditions and restrictions. Instead of trying to directly compute either u or γ and incorporate boundary conditions or restrictions by a penalization term, we shall enforce them in the construction of the neural network approximations. For that purpose, we make use of the following notion.

Definition 3.1 (surrogate distance function)**.** *Let $\Gamma \subset \overline{\partial\Omega}$ be a closed set. We say that a Lipschitz continuous function $d_\Gamma \colon \overline{\Omega} \to \mathbb{R}$ is a* surrogate distance function *if there exist constants $c_1, c_2 \in \mathbb{R}_{>0}$ such that $c_1 d_\Gamma(x) \leq dist(x, \Gamma) \leq c_2 d_\Gamma(x)$ for all $x \in \overline{\Omega}$.*

Notice that the function $dist(x, \Gamma)$ is itself a surrogate distance function. However, since an explicit expression for $dist(x, \Gamma)$ is seldom available, we prefer to state our results in terms of a computable substitute $d_\Gamma(x)$. Keeping that in mind, we briefly comment on the use of surrogate distance functions in the strong imposition of admissibility and Dirichlet boundary conditions at a continuous level. Consider a Lipschitz continuous function $a \colon \mathbb{R} \to [0, \infty)$ such that $a(t) = 0$ if $t = 0$. In the computation of u in either (2.5) or (2.7), we restrict the class of functions to be

(3.2)
$$u(x) := g(x) + d_{\partial\Omega}(x)\, a(v(x)),$$

where the unknown is the function $v \colon \Omega \to \mathbb{R}$, g is the obstacle, and $d_{\partial\Omega}$ is a surrogate distance function to $\partial\Omega$. In a similar fashion, we can incorporate the condition $\lambda \geq 0$ by considering

$$(3.3) \qquad \gamma(x) := \operatorname{div} \boldsymbol{\phi}(x) + a(\eta(x)),$$

and the new unknown is the function $\eta \colon \Omega \to \mathbb{R}$. Note that, in this setting, $a(\eta)$ plays the role of the Lagrange multiplier λ. Therefore, we can seek $\boldsymbol{y} = (v, \boldsymbol{\psi}, \eta)$ such that the corresponding tuple $(u, \boldsymbol{\phi}, \gamma)$, with u given by (3.2), $\boldsymbol{\psi} = \boldsymbol{\phi}$, and γ according to (3.3), minimizes the loss function $\mathcal{L}$. The distinction in notation for the flux variable –employing $\boldsymbol{\psi}$ rather than $\boldsymbol{\phi}$– serves to differentiate its role within two distinct contexts. Specifically, $\boldsymbol{\psi}$ denotes a component of the triple $(v, \boldsymbol{\psi}, \eta)$, which belongs to an unconstrained function space, whereas the notation $(u, \boldsymbol{\phi}, \gamma)$ is reserved for admissible triples satisfying suitable constraints.

3.2. **Computational aspects.** In the following, we assume the user has provided a surrogate distance function as in Definition 3.1. For a comprehensive analysis on the construction of such functions, even with higher differentiability properties than those required in this work, we refer to [33]. The surrogate distance can also be approximated by a neural network, as described in [6]. We consider a set of random points $\{x_k\}_{k=1}^N \subset \Omega$ and aim to minimize the cost functional L_N introduced in (3.1). The trainable parameters $\boldsymbol{\Theta}$ emerge in the computation of the auxiliary functions v and η from the construction of u and γ (see (3.2) and (3.3)).

In broad terms, the method we propose can be summarized as follows. Until some stop criterion is reached, do:

- Sample random points $\{x_k\}_{k=1}^N \subset \Omega$.
- Update parameters: $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \ell\nabla_{\boldsymbol{\Theta}} L_N(u_{\boldsymbol{\Theta}}, \boldsymbol{\phi}_{\boldsymbol{\Theta}}, \gamma_{\boldsymbol{\Theta}})$.
- Adjust learning rate ℓ.

The computation of $L_N(u, \boldsymbol{\phi}, \eta)$ requires evaluating the derivatives of u and $\boldsymbol{\phi}$ with respect to the input variables at $\{x_k\}_{k=1}^N$. Since all the involved functions are smooth, these derivatives can be efficiently computed by using back propagation.

For the numerical examples, we implemented our algorithm within the PyTorch library. We utilized one- to three-layer neural networks with a SoftPlus activation function for the main variables v and $\boldsymbol{\psi}$. A piecewise constant neural network was used for the variable η and a three-layer network with a ReLU activation function was employed for high-dimensional examples. The ADAM [24] optimization algorithm demonstrated good results in numerical experiments. Further details about the implementation of the method can be found in Section 5.

4. CONVERGENCE ANALYSIS

We consider a set of unconstrained neural network functions

$$(4.1) \qquad C_m := \left\{ (v_{\boldsymbol{\Theta}}, \boldsymbol{\psi}_{\boldsymbol{\Theta}}, \eta_{\boldsymbol{\Theta}}) : v_{\boldsymbol{\Theta}} : \mathbb{R}^d \to \mathbb{R}, \ \boldsymbol{\psi}_{\boldsymbol{\Theta}} : \mathbb{R}^d \to \mathbb{R}^d, \ \eta_{\boldsymbol{\Theta}} : \mathbb{R}^d \to \mathbb{R} \right\},$$

where we collect all the parameters in a vector $\boldsymbol{\Theta} \in \mathbb{R}^m$. The choice of the neural network architecture is intentionally left ambiguous at this point. In what follows, we will assume certain approximation properties of this set and later focus on a specific instance of C_m and a variant from it.

Assuming that we are able to construct a surrogate auxiliary function $d_{\partial\Omega}$ as in Definition 3.1, and considering a suitable choice of $a : \mathbb{R} \to [0, \infty)$, we define the set of discrete admissible functions

$$
(4.2) \quad W_m := \Big\{ \boldsymbol{p_\Theta} = (u_\Theta, \boldsymbol{\phi_\Theta}, \gamma_\Theta) : u_\Theta = g + d_{\partial\Omega} a(v_\Theta), \ \boldsymbol{\phi_\Theta} = \boldsymbol{\psi_\Theta},
$$
$$
\gamma_\Theta = \operatorname{div} \boldsymbol{\psi_\Theta} + a(\eta_\Theta), \quad (v_\Theta, \boldsymbol{\psi_\Theta}, \eta_\Theta) \in C_m \Big\},
$$

which we regard as a subset of $\mathcal{W} = H_0^1(\Omega) \times L^2(\Omega; \mathbb{R}^d) \times L^2(\Omega)$ and therefore furnish with the norm $\|(u_\Theta, \boldsymbol{\phi_\Theta}, \gamma_\Theta)\|_{\mathcal{W}} = (\|\nabla u_\Theta\|^2 + \|\boldsymbol{\phi_\Theta}\|^2 + \|\gamma_\Theta\|^2)^{\frac{1}{2}}$. We remark that the fulfillment of the boundary and admissibility conditions is guaranteed within the set W_m, in the sense that $u_\Theta = 0$ whenever $d_{\partial\Omega} = 0$, and moreover $u_\Theta \geq g$, and $\gamma_\Theta - \operatorname{div} \boldsymbol{\phi_\Theta} \geq 0$.

4.1. Γ-convergence.

We seek to prove the convergence of the neural network approximations obtained through our method to the minimizers of the least-squares functional $\mathcal{L}$ in (2.7). To this end, we employ Γ-convergence theory, which guarantees that if a sequence of functionals Γ-converges and their minimizers converge, then solutions to the limit problem exist, along with the convergence of both minimum values and minimizers. Below, we provide a brief review of Γ-convergence and some key results relevant to this work. We refer to [7] for a comprehensive treatment of the subject.

Definition 4.1 (sequential Γ-convergence). *Let X be a metric space and let F_n, $F : X \to \overline{\mathbb{R}}$, where $\overline{\mathbb{R}} := [-\infty, +\infty]$. We say that F_n Γ-converges to F (and write $F_n \xrightarrow{\Gamma} F$) if, for every $x \in X$ we have*

- *(lim-inf inequality) for every sequence $\{x_n\}_{n\in\mathbb{N}} \subset X$ converging to x,*

$$
F(x) \leq \liminf_{n\to\infty} F_n(x_n);
$$

- *(lim-sup inequality) there exists a sequence $\{x_n\}_{n\in\mathbb{N}}$ converging to x such that*

$$
F(x) \geq \limsup_{n\to\infty} F_n(x_n).
$$

Definition 4.2 (equi-coercivity). *Let $\{F_n\}_{n\in\mathbb{N}}$ be a sequence of functions $F_n : X \to \overline{\mathbb{R}}$. We say that $\{F_n\}$ is equi-coercive if for all $t \in \mathbb{R}$ there exists a compact set $K_t \subset X$ such that $\{F_n \leq t\} \subset K_t$.*

Our primary motivation for employing Γ-convergence theory stems from the following result, which states that if a sequence of functionals is equi-coercive and Γ-converges, then any corresponding sequence of minimizers converges to a minimizer of the Γ-limit functional. In essence, Γ-convergence, together with an appropriate compactness condition, ensures the convergence of minimizers.

Theorem 4.1 (fundamental theorem of Γ-convergence). *Let (X, d) be a metric space, $\{F_n\}_{n\in\mathbb{N}}$ be an equi-coercive sequence of functions on X, and F be such that $F_n \xrightarrow{\Gamma} F$. Then,*

$$
\exists \min_X F = \lim_{n\to\infty} \inf_X F_n.
$$

Moreover, if $\{x_n\}_{n\in\mathbb{N}}$ is a precompact sequence in X such that $\lim_{n\to\infty} F_n(x_n) = \lim_{n\to\infty} \inf_X F_n$, then every limit of a subsequence of $\{x_n\}$ is a minimum point for F.

4.2. **Convergence of the method.** Our analysis is based on two generic hypotheses on the neural network functions. In Section 4.3, we describe specific instances of the set C_m in which these assumptions are verified.

Assumption 4.1. *We assume the following conditions. In first place, we state two conditions that we shall assume throughout the remainder of the paper.*

1) *We recall that our method is based on the first-order formulation developed in Section 2, which requires Ω be a bounded, Lipschitz domain, $f \in L^2(\Omega)$, and $g \in H_0^1(\Omega)$.*
2) *We additionally assume that the neural network functions in C_m (cf. (4.1)) are such that*
   - *the activation functions used for $v_{\Theta}, \phi_{\Theta}$ are Lipchitz continuous;*
   - *the activation functions used for η_{Θ} are either bounded or Lipschitz continuous.*
   From these, to construct the classes W_m, we need to fix a surrogate auxiliary function $d_{\partial\Omega}$ (Definition 3.1) and a Lipschitz continuous function $a : \mathbb{R} \to [0, \infty)$.

Besides these two structural assumptions, we list below two conditions on the approximation capability and stability with respect to parameters of the neural network classes. We recall that the neural network class W_m is deemed as a subset of $\mathcal{W} = H_0^1(\Omega) \times L^2(\Omega; \mathbb{R}^d) \times L^2(\Omega)$.

3) *Let $\boldsymbol{p}_0 \in K^{\mathcal{W}}$ be given by (2.8). Then, $\boldsymbol{p}_0$ can be approximated by the neural network spaces, that is, we assume that*

$$\text{(H1)} \qquad d(\boldsymbol{p}_0, W_m) := \inf_{\boldsymbol{p}_{\Theta} \in W_m} \|\boldsymbol{p}_0 - \boldsymbol{p}_{\Theta}\|_{\mathcal{W}} \to 0 \quad \text{as } m \to \infty.$$

4) *Let $R > 0$ be fixed as in (4.3) below. We assume*

$$\text{(H2)} \qquad \text{The map } \overline{B(0, R)} \to W_m \text{ such that } \boldsymbol{\Theta} \mapsto \boldsymbol{p}_{\Theta} \text{ is continuous.}$$

The first two conditions are essential for our analysis, while in Section 4.3, we give examples of constructions of the set C_m in which the latter two assumptions are verified.

Regularization plays an instrumental role in neural network approximations: one trades a small decrease in training accuracy for better generalization, thus reducing overfitting. In the context of numerical PDEs, regularization avoids convergence towards non-physical (in our setting, non-variational), almost everywhere solutions [6]. Given $R > 0$, that shall remain fixed, we consider $L : \mathbb{R}^m \to \mathbb{R}$ as

$$\text{(4.3)} \qquad L(\boldsymbol{\Theta}; R) := \begin{cases} \mathcal{L}(\boldsymbol{p}_{\Theta}) & \text{if } |\boldsymbol{\Theta}| \leq R, \\ +\infty & \text{otherwise.} \end{cases}$$

On the other hand, we recall that in the admissible class W_m we have the representation $u_{\Theta} = g + d_{\partial\Omega}a(v_{\Theta})$, $\boldsymbol{\psi}_{\Theta} = \boldsymbol{\phi}_{\Theta}$, $\gamma_{\Theta} = \operatorname{div}\boldsymbol{\phi}_{\Theta} + a(\eta_{\Theta})$ with $(v_{\Theta}, \boldsymbol{\psi}_{\Theta}, \eta_{\Theta}) \in C_m$, so that we can rewrite the discrete loss functional (3.1) as

$$\text{(4.4)} \qquad L_N(\boldsymbol{\Theta}; f, g) = \frac{|\Omega|}{N} \sum_{k=1}^{N} \sum_{i=1}^{4} G_i(\boldsymbol{\Theta}, x_k),$$

with $G_1, G_2, G_3, G_4 : \mathbb{R}^m \times \Omega \to \mathbb{R}$ given by

$$\text{(4.5)} \qquad \begin{aligned} G_1(\boldsymbol{\Theta}, x) &:= |\operatorname{div}\boldsymbol{\psi}_{\Theta}(x) + a(\eta_{\Theta}(x)) + f(x)|^2, \\ G_2(\boldsymbol{\Theta}, x) &:= |\nabla g(x) + \nabla(d_{\partial\Omega}a(v_{\Theta}))(x) - \boldsymbol{\psi}_{\Theta}(x)|^2, \\ G_3(\boldsymbol{\Theta}, x) &:= [\operatorname{div}\boldsymbol{\psi}_{\Theta}(x) + a(\eta_{\Theta}(x))]\, d_{\partial\Omega}(x)a(v_{\Theta}(x)), \\ G_4(\boldsymbol{\Theta}, x) &:= \boldsymbol{\psi}_{\Theta}(x) \cdot \nabla(d_{\partial\Omega}a(v_{\Theta}))(x). \end{aligned}$$

The four functions G_i correspond to the four terms in the sum (3.1) upon the correspondence $u \mapsto g + d_{\partial\Omega}a(v_{\boldsymbol{\Theta}})$, $\boldsymbol{\phi} \mapsto \boldsymbol{\psi}_{\boldsymbol{\Theta}}$, $\gamma \mapsto \operatorname{div}\boldsymbol{\psi}_{\boldsymbol{\Theta}} + a(\eta_{\boldsymbol{\Theta}})$.

We split the proof of convergence of our method into several steps. We start by proving the following auxiliary lemma, that shows certain boundedness properties of the functions in (4.5) with respect to the neural network parameters. In the following, we restrict the domain of all neural network functions to be Ω.

Lemma 4.1 (boundedness)**.** *Let* $G_1, \ldots, G_4$ *be defined as in* (4.5)*, with discrete admissible functions given by* (4.2)*, and fix* $R > 0$ *according to* (4.3)*. Then, under Assumption 4.1, it holds that* $G_3, G_4 \in L^\infty(B(0, R) \times \Omega)$*. Moreover, there exist* $s_1, s_2 \in L^1(\Omega)$*, depending on* R*, such that* $|G_i(\boldsymbol{\Theta}, x)| \leq s_i(x)$ *for all* $(\boldsymbol{\Theta}, x) \in \overline{B(0, R)} \times \Omega$ *and* $i = 1, 2$.

Proof. Consider a triple $(v_{\boldsymbol{\Theta}}, \boldsymbol{\psi}_{\boldsymbol{\Theta}}, \eta_{\boldsymbol{\Theta}}) \in C_m$, where C_m is the set of unconstrained neural network functions, cf. (4.1). We begin by considering the map $\mathscr{F} \colon \mathbb{R}^m \times \Omega \to \mathbb{R}$, given by $\mathscr{F}(\boldsymbol{\Theta}, x) := v_{\boldsymbol{\Theta}}(x)$. Under Assumption 4.1, namely that $v_{\boldsymbol{\Theta}}$ possesses Lipschitz continuous activation functions, it follows that $\mathscr{F}$ itself is Lipschitz continuous. Consequently, $\mathscr{F}$ is bounded on $B(0, R) \times \Omega$, and its (weak) derivatives are likewise essentially bounded on this domain. Analogously, introducing $\mathscr{G} \colon \mathbb{R}^m \times \Omega \to \mathbb{R}^m$ via $\mathscr{G}(\boldsymbol{\Theta}, x) := \boldsymbol{\psi}_{\boldsymbol{\Theta}}(x)$, we deduce that $\mathscr{G}$ exhibits the same boundedness properties, with essentially bounded derivatives over $B(0, R) \times \Omega$. Furthermore, considering $\mathscr{H} \colon \mathbb{R}^m \times \Omega \to \mathbb{R}$ given by $\mathscr{H}(\boldsymbol{\Theta}, x) := \operatorname{div}\boldsymbol{\psi}_{\boldsymbol{\Theta}}(x) + a(\eta_{\boldsymbol{\Theta}}(x))$, the condition on $\eta_{\boldsymbol{\Theta}}$ from Assumption 4.1 yields its essential boundedness on $B(0, R) \times \Omega$.

The function G_4 can be constructed using $\mathscr{F}$ and $\mathscr{G}$. Given that $d_{\partial\Omega}$ is a surrogate distance function and a is Lipschitz continuous, we immediately deduce the essential boundedness of G_4 on $B(0, R) \times \Omega$. Similarly, because $G_3(\boldsymbol{\Theta}, x) = \mathscr{H}(\boldsymbol{\Theta}, x)d_{\partial\Omega}(x)a(\mathscr{F}(\boldsymbol{\Theta}, x))$, the boundedness of G_3 over the same domain follows.

Next, the relation $G_1(\boldsymbol{\Theta}, x) = |\mathscr{H}(\boldsymbol{\Theta}, x) + f(x)|^2$ yields the estimate

$$G_1(\boldsymbol{\Theta}, x) \leq 2|\mathscr{H}(\boldsymbol{\Theta}, x)|^2 + 2|f(x)|^2 \leq 2M + 2|f(x)|^2 =: s_1(x),$$

for some constant $M > 0$. Thus, the inclusion $s_1 \in L^1(\Omega)$ follows from our assumption $f \in L^2(\Omega)$. A similar argument allows us to conclude that there exists $s_2 \in L^1(\Omega)$ satisfying $G_2(\boldsymbol{\Theta}, x) \leq s_2(x)$ for all $(\boldsymbol{\Theta}, x) \in B(0, R) \times \Omega$. $\qquad\square$

Our goal is to approximate the minimizer $\boldsymbol{p}_0$ of $\mathcal{L}$ in $K^{\mathcal{W}}$, which corresponds to the solution of the obstacle problem, using minimizers of the same functional over the set of admissible neural network functions W_m. However, in practice, the exact evaluation of $\mathcal{L}(\boldsymbol{p}_{\boldsymbol{\Theta}})$ is unattainable due to the reliance on numerical integration. Consequently, instead of approximating $\boldsymbol{p}_0$ through the minimizers of $\mathcal{L}$ in W_m, the following lemma targets accounting for the error introduced in the computation of the neural network minimizers.

Lemma 4.2 (approximation with W_m)**.** *For every* $m \in \mathbb{N}$*, we define the set of neural network quasi-minimizers*

$$\mathcal{I}_m := \{\boldsymbol{p}_{\boldsymbol{\Theta}_m} \in W_m : \mathcal{L}(\boldsymbol{p}_{\boldsymbol{\Theta}_m}) \leq \mathcal{L}(\boldsymbol{p}_{\boldsymbol{\Theta}}) + 1/m \quad \forall \boldsymbol{p}_{\boldsymbol{\Theta}} \in W_m\}.$$

Then, under Assumption 4.1, if $\boldsymbol{p}_0 \in K^{\mathcal{W}}$ *is the unique minimizer of* $\mathcal{L}$ *in* $K^{\mathcal{W}}$*, we have*

$$\sup_{\boldsymbol{p}_{\boldsymbol{\Theta}_m} \in \mathcal{I}_m} \|\boldsymbol{p}_{\boldsymbol{\Theta}_m} - \boldsymbol{p}_0\|_{\mathcal{W}} \to 0 \quad \text{as } m \to \infty.$$

Proof. Let $\boldsymbol{p_\Theta} \in W_m$ be arbitrary and $\boldsymbol{p}_0 \in K^{\mathcal{W}}$ be the minimizer of $\mathcal{L}$. From Propositions 2.1 and 2.2, we have

$$(4.6) \qquad \alpha \|\boldsymbol{p_\Theta} - \boldsymbol{p}_0\|_{\mathcal{W}}^2 \leq \mathcal{L}(\boldsymbol{p_\Theta}; f, g) \leq \beta(1 + \|\boldsymbol{p_\Theta} - \boldsymbol{p}_0\|_{\mathcal{W}}) \, \|\boldsymbol{p_\Theta} - \boldsymbol{p}_0\|_{\mathcal{W}}$$

for some positive constants α, β.

Let $\varepsilon > 0$. By assumption (H1), we consider $m_0 > 0$ such that $d(\boldsymbol{p}_0, W_m) < \varepsilon$ and $1/m < \varepsilon$ for all $m > m_0$. For every $m > m_0$, there exists $\boldsymbol{p_{\Theta^*}} \in W_m$ with $d(\boldsymbol{p}_0, W_m) \geq \|\boldsymbol{p_{\Theta^*}} - \boldsymbol{p}_0\|_{\mathcal{W}} - \varepsilon$, namely, $\|\boldsymbol{p_{\Theta^*}} - \boldsymbol{p}_0\|_{\mathcal{W}} < 2\varepsilon$. Then, for all $m > m_0$ and every neural network quasi-minimizer $\boldsymbol{p_{\Theta_m}} \in \mathcal{I}_m$, we combine the bounds in (4.6) to obtain

$$\alpha \|\boldsymbol{p_{\Theta_m}} - \boldsymbol{p}_0\|_{\mathcal{W}}^2 \leq \mathcal{L}(\boldsymbol{p_{\Theta_m}}; f, g)$$
$$\leq \mathcal{L}(\boldsymbol{p_{\Theta^*}}; f, g) + \varepsilon \leq \beta(1 + \|\boldsymbol{p_{\Theta^*}} - \boldsymbol{p}_0\|_{\mathcal{W}}) \, \|\boldsymbol{p_{\Theta^*}} - \boldsymbol{p}_0\|_{\mathcal{W}} + \varepsilon < 2\beta(1 + 2\varepsilon)\varepsilon + \varepsilon.$$

We have thus shown that for every $\varepsilon > 0$ there exists $m_0 > 0$ such that, for all $m > m_0$, if $\boldsymbol{p_{\Theta_m}} \in \mathcal{I}_m$ then

$$\|\boldsymbol{p_{\Theta_m}} - \boldsymbol{p}_0\|_{\mathcal{W}} \leq C\sqrt{\varepsilon},$$

and thereby the proof is concluded. $\qquad\square$

The previous result is of theoretical interest, as it involves the continuous functional $\mathcal{L}$. In practice, we shall make use of Monte Carlo integration and deal instead with a regularized version of (4.4). More precisely, let $\{X_i\}_{i\in\mathbb{N}}$ be an i.i.d. sequence of random variables on a probability space (Ξ, Σ, P) with $X_i : \Xi \to \Omega \ \forall i \in \mathbb{N}$, with a uniform probability density on Ω. Given $\xi \in \Xi$, $R > 0$ as in (4.3), and $N \in \mathbb{N}$ we set $V_N(\xi) := \cup_{i \leq N}\{X_i(\xi)\}$, and introduce the regularized discrete functional $L_{\xi,N} : \mathbb{R}^m \to \overline{\mathbb{R}}$,

$$(4.7) \qquad L_{\xi,N}(\boldsymbol{\Theta}) := \begin{cases} \dfrac{|\Omega|}{N} \displaystyle\sum_{x \in V_N(\xi)} \sum_{i=1}^{4} G_i(\boldsymbol{\Theta}, x) & \text{if } |\boldsymbol{\Theta}| \leq R, \\ + \infty & \text{otherwise,} \end{cases}$$

where $G_1, \ldots, G_4$ are given by (4.5).

We aim to prove the almost sure Γ-convergence of the sequence of functionals $\{L_{\xi,N}\}_{N\in\mathbb{N}}$ towards L as the number of quadrature points N tends to infinity. An auxiliary tool for this purpose is the fact that such a convergence holds in a pointwise sense.

Lemma 4.3 (almost sure pointwise convergence). *Let $R > 0$, and L be defined by (4.3). Additionally, as in (4.7), consider the functional $L_{\xi,N}$ and $\{X_i\}_{i\in\mathbb{N}}$ an i.i.d. family of random variables on the probability space (Ξ, Σ, P). Then, for all $\boldsymbol{\Theta} \in \mathbb{R}^m$, it holds that $L_{\xi,N}(\boldsymbol{\Theta}) \to L(\boldsymbol{\Theta})$ as $N \to \infty$ P-almost surely.*

Proof. Because the parameter $R > 0$ in the definitions of of L and $L_{\xi,N}$ is the same, we have $L(\boldsymbol{\Theta}) = L_{\xi,N}(\boldsymbol{\Theta}) = +\infty$ whenever $|\boldsymbol{\Theta}| > R$. Therefore, there is nothing to be proven in that case and we can assume $|\boldsymbol{\Theta}| \leq R$. Recalling $V_N(\xi) = \cup_{i \leq N}\{X_i(\xi)\}$ with $\xi \in \Xi$, an application of the strong law of the large numbers gives

$$\frac{|\Omega|}{N} \sum_{x \in V_N(\xi)} G_i(\boldsymbol{\Theta}, x) \xrightarrow[N\to\infty]{a.s.} \int_\Omega G_i(\boldsymbol{\Theta}, x) \, dx.$$

The claim follows immediately. $\qquad\square$

According to Definition 4.1, to prove the almost sure Γ-convergence of $\{L_{\xi,N}\}$ towards L as $N \to \infty$, it suffices to show that, for every $\boldsymbol{\Theta} \in \mathbb{R}^m$, the lim-inf inequality holds and that one can construct a recovery sequence satisfying the lim-sup inequality. Our argument follows the same steps as in [6, Theorem 3.2]; we outline the proof for clarity.

Theorem 4.2 (almost sure Γ-convergence)**.** *Let $R > 0$, L be as in (4.3), and $L_{\xi,N}$ and $\{X_i\}_{i \in \mathbb{N}}$ be an i.i.d. family of random variables defined in the probability space (Ξ, Σ, P) as in (4.7). Then, under Assumption 4.1, it holds that $L_{\xi,N} \xrightarrow{\Gamma} L$ as $N \to \infty$ P-almost surely.*

Proof. Given $\boldsymbol{\Theta} \in \mathbb{R}^m$, we consider the trivial recovery sequence $\{\boldsymbol{\Theta}_N\}_{N \in \mathbb{N}} \subset \mathbb{R}^m$, $\boldsymbol{\Theta}_N \equiv \boldsymbol{\Theta}$. Then, Lemma 4.3 yields the P-almost sure lim-sup inequality.

As for the lim-inf inequality, we first observe that, by hypothesis (H2), if $\boldsymbol{\Theta}_N \to \boldsymbol{\Theta}$ then $\boldsymbol{p}_{\boldsymbol{\Theta}_N} \to \boldsymbol{p}_{\boldsymbol{\Theta}}$ in the norm $\|\cdot\|_{\mathcal{W}}$. This, combined with the Poincaré inequality and the boundedness of Ω, implies $G_i(\boldsymbol{\Theta}_N, x) \to G_i(\boldsymbol{\Theta}, x)$ almost everywhere (up to a subsequence) in Ω ($i \in \{1, \dots, 4\}$) for the functions defined in (4.5). Additionally, from Lemma 4.1, we can bound $G_i(\boldsymbol{\Theta}_N, x) \le s_i(x)$, for some $s_i \in L^1(\Omega)$, $i \in \{1, \dots, 4\}$. By combining this with Lemma 4.3, we can readily reproduce the arguments presented in [6, Theorem 3.2]. $\qquad\square$

Finally, we can combine the previous Γ-convergence result with a suitable form of equi-coercivity to conclude the convergence of minimizers. We emphasize that the following theorem does not assume that one is able to compute the minimum of the discrete loss functional, but instead that one can asymptotically recover such a minimizer as the number of quadrature points tends to infinity.

Theorem 4.3 (convergence)**.** *Let Assumption 4.1 hold and, moreover, assume that, given $m \in \mathbb{N}$ and $R > 0$, we can construct a sequence $\{\boldsymbol{\Theta}_N\}_{N \in \mathbb{N}} \subset \overline{B(0, R)} \subset \mathbb{R}^m$ such that $\lim_{N \to \infty} L_{\xi,N}(\boldsymbol{\Theta}_N) = \lim_{N \to \infty} \inf_{\boldsymbol{\Theta} \in \mathbb{R}^m} L_{\xi,N}(\boldsymbol{\Theta})$, with $L_{\xi,N}$ defined as in (4.7). Let $\boldsymbol{p}_0 = \arg\min_{\boldsymbol{p} \in K^{\mathcal{W}}} \mathcal{L}(\boldsymbol{p})$. Then, for every $\varepsilon > 0$, there P-almost surely exist $m_0 = m_0(\varepsilon) \in \mathbb{N}$, $R = R(m_0) > 0$ and $N_0 = N_0(m_0) \in \mathbb{N}$ such that, if one constructs a sequence $\{\boldsymbol{\Theta}_N\}_{N \in \mathbb{N}}$ as above, then*

$$\|\boldsymbol{p}_N - \boldsymbol{p}_0\|_{\mathcal{W}} \le \varepsilon \quad \text{for all } N > N_0,$$

where $\boldsymbol{p}_N := (u_{\boldsymbol{\Theta}_N}, \phi_{\boldsymbol{\Theta}_N}, \gamma_{\boldsymbol{\Theta}_N}) \in W_{m_0}$ is the neural network function defined by the parameters $\boldsymbol{\Theta}_N$.

Proof. Given $\varepsilon > 0$, we consider the set of neural network quasi-minimizers introduced in Lemma 4.2, $\mathcal{I}_m = \{\boldsymbol{p}_{\boldsymbol{\Theta}_m} \in W_m : \mathcal{L}(\boldsymbol{p}_{\boldsymbol{\Theta}_m}) \le \mathcal{L}(\boldsymbol{p}_{\boldsymbol{\Theta}}) + 1/m \; \forall \boldsymbol{p}_{\boldsymbol{\Theta}} \in W_m\}$. By that lemma, there exists $m_0 > 0$ such that

(4.8) $$\|\boldsymbol{p}_0 - \boldsymbol{p}_{\boldsymbol{\Theta}_{m_0}}\|_{\mathcal{W}} < \varepsilon/2 \quad \text{for all } \boldsymbol{p}_{\boldsymbol{\Theta}_{m_0}} \in \mathcal{I}_{m_0}.$$

Next, we fix $R_0 > 0$ large enough so that there exists a quasi-minimizer $\boldsymbol{p}_{\boldsymbol{\Theta}^*} \in \mathcal{I}_{m_0}$ with $\boldsymbol{\Theta}^* \in \overline{B(0, R_0)}$. Because the functional L defined in (4.3) coincides with $\mathcal{L}$ over $\overline{B(0, R_0)}$, the reasoning above implies that $\boldsymbol{p}_{\boldsymbol{\Theta}_{m_0}} \in \mathcal{I}_{m_0}$ for all $\boldsymbol{\Theta}_{m_0} \in \arg\min_{\boldsymbol{\Theta} \in B(0, R_0)} L(\boldsymbol{\Theta})$.

For this choice of m_0 and R_0, from Theorem 4.2 we have $L_{\xi,N} \xrightarrow{\Gamma} L$ P-almost surely as $N \to \infty$. From the definition (4.7) of $L_{\xi,N} : \mathbb{R}^{m_0} \to \overline{\mathbb{R}}$, it follows immediately that $\{L_{\xi,N}\}_{N \in \mathbb{N}}$ is an equi-coercive sequence, according to Definition 4.2.

Therefore, using Theorem 4.1 together with the fact that every cluster point in $\{\boldsymbol{\Theta}_N\}$ is a minimum point for L and the continuity of the map $\boldsymbol{\Theta} \mapsto p_{\boldsymbol{\Theta}}$, we deduce that P-almost surely there exists $N_0 > 0$ such that

$$(4.9) \qquad \|\boldsymbol{p}_N - \boldsymbol{p}_{m_0}\|_{\mathcal{W}} < \varepsilon/2$$

for all $N > N_0$ for some $\boldsymbol{p}_{\boldsymbol{\Theta}_{m_0}} \in \mathcal{I}_{m_0}$.

The proof concludes by combining (4.8) and (4.9). □

4.3. **Approximation properties of neural networks.** The convergence results from the previous section are based on the four conditions from Assumption 4.1. The first two conditions therein are *structural*, in the sense that they involve either the problem data or choices about the neural network activation and auxiliary functions. In contrast, the latter two express properties of the neural networks in either terms of approximation capabilities or their stability.

In this section, we show certain constructions of the set W_m that satisfy the structural condition 2) from Assumption 4.1 and, for them, we prove that hypotheses (H1) and (H2) hold.

4.3.1. *A first example.* For the sake of clarity, we shall restrict our analysis to fully-connected, feedforward neural networks having one hidden layer with n neurons. Accordingly, we consider the following set of unconstrained neural network functions,

$$(4.10) \quad C_m := \Big\{ (v_{\boldsymbol{\Theta}}, \boldsymbol{\psi}_{\boldsymbol{\Theta}}, \eta_{\boldsymbol{\Theta}}) : v_{\boldsymbol{\Theta}} = B_v \sigma(A_v x + c_v), \ \boldsymbol{\psi}_{\boldsymbol{\Theta}} = B_{\boldsymbol{\psi}} \sigma(A_{\boldsymbol{\psi}} x + c_{\boldsymbol{\psi}}), \ \eta_{\boldsymbol{\Theta}} = B_\eta \sigma(A_\eta x + c_\eta) \Big\}.$$

Above $A_v, A_{\boldsymbol{\psi}}, A_\eta \in \mathbb{R}^{n \times d}$, $c_v, c_{\boldsymbol{\psi}}, c_\eta \in \mathbb{R}^{n \times 1}$, $B_v, B_\eta \in \mathbb{R}^{1 \times n}$, $B_{\boldsymbol{\psi}} \in \mathbb{R}^{d \times n}$, and $\sigma : \mathbb{R}^n \to \mathbb{R}^n$, and σ a Lipschitz continuous, non-polynomial activation function, applied coordinate-wise. We gather all parameters in $\boldsymbol{\Theta} \in \mathbb{R}^m$, where $m = n(4d + 5)$. Note that, in this setting, stating $m \to \infty$ is equivalent to expressing that the number of neurons n increases without bound. Furthermore, in this section we shall operate either under the choice

$$(4.11) \qquad a(t) := \mathrm{ReLU}(t) = \max\{0, t\}.$$

The approximation properties of our spaces are proved in Proposition 4.2 below. Before addressing this issue, we note that the strong imposition of the homogenous boundary condition in (3.2), calls for an appropriate understanding of the space to which the function $\frac{u-g}{d_\Omega}$ belongs. This is the purpose of the following definition and the subsequent two lemmas; we point out that the Lipschitz condition on Ω introduced in Assumption 4.1 is instrumental for the validity of our discussion.

Definition 4.3 (weighted L^2 space). *Let $d_{\partial\Omega}$ be a surrogate distance function (Definition 3.1). For $\alpha \geq 0$, we introduce the weighted L^2 space*

$$L^2(\Omega, \alpha) := \{w : \ d_{\partial\Omega}^\alpha w \in L^2(\Omega)\},$$

equipped with the norm $\|w\|_{L^2(\Omega,\alpha)} := \|d_{\partial\Omega}^\alpha w\|$ and its Sobolev extension

$$H^1(\Omega, \alpha, \beta) := \{w \in L^2(\Omega, \alpha) : \ \nabla w \in L^2(\Omega, \beta)\},$$

with the induced norm $\|w\|_{H^1(\Omega,\alpha,\beta)} := \|w\|_{L^2(\Omega,\alpha)} + \|\nabla w\|_{L^2(\Omega,\beta)}$.

It is clear that the definition above is not dependent on the particular choice of $d_{\partial\Omega}$, as all resulting spaces turn out to be equivalent. The following result is implicitly given in [25] but not stated in the form needed in this article. For completeness, we outline the proof.

Lemma 4.4 (density). *The set $C^\infty(\overline{\Omega})$ is dense in $H^1(\Omega, 0, 1)$.*

Proof. Taking $k = 1$, $\varepsilon = 2$ and $p = 2$, in [25, Proposition 9.6], we first note that $H^1(\Omega, 0, 1) = H^1(\Omega, 1, 1)$. Combining this with the identity $\overline{C^\infty(\overline{\Omega})}^{\|\cdot\|_{H^1(\Omega, 1, 1)}} = H^1(\Omega, 1, 1)$ shown in [25, Theorem 7.2], we deduce the claimed density. $\qquad\square$

Lemma 4.5 (embedding). *For all $w \in H_0^1(\Omega)$, it holds that $w/d_{\partial\Omega} \in H^1(\Omega, 0, 1)$ and*
$$\|w/d_{\partial\Omega}\|_{H^1(\Omega, 0, 1)} \leq C\|w\|_{H_0^1(\Omega)}$$
with a constant depending on Ω and the choice of $d_{\partial\Omega}$.

Proof. Let $w \in H_0^1(\Omega)$. On the one hand, Hardy's inequality readily gives
$$(4.12) \qquad\qquad \|w/d_{\partial\Omega}\|^2 \leq C(\Omega)\|\nabla w\|^2.$$

On the other hand,
$$\begin{aligned}
\|d_{\partial\Omega}\nabla(w/d_{\partial\Omega})\|^2 &= \int_\Omega d_{\partial\Omega}^2 \left(\frac{\nabla w}{d_{\partial\Omega}} - \frac{w\nabla d_{\partial\Omega}}{d_{\partial\Omega}^2}\right)^2 \\
&\leq 2\int_\Omega |\nabla w|^2 + \frac{w^2|\nabla d_{\partial\Omega}|^2}{d_{\partial\Omega}^2} \\
&\leq C(\Omega, \|\nabla d_{\partial\Omega}\|_{L^\infty(\Omega)})\|\nabla w\|^2,
\end{aligned}$$
where the last inequality follows from (4.12) and the uniform boundedness of $|\nabla d_{\partial\Omega}|$. $\qquad\square$

Our choice of a in (4.11) gives rise to the following well-known result. While its proof is standard, we include it here for completeness.

Lemma 4.6 (convergence of composition). *Let a be given by (4.11) and let $\beta \geq 0$. If $w_m \to w$ in $H^1(\Omega, 0, \beta)$, then $a(w_m) \to a(w)$ in $H^1(\Omega, 0, \beta)$.*

Proof. By the Lipschitz continuity of a, it is easy to observe that $a(w_m) \to a(w)$ in $L^2(\Omega)$:
$$\|a(w_m) - a(w)\|_{L^2(\Omega)} \leq \|w_m - w\|_{L^2(\Omega)}.$$

To prove the convergence of the gradients, we note $\nabla(a(w)) = \chi_{\{w>0\}}\nabla w$ and write
$$(4.13) \qquad \nabla(a(w_m)) - \nabla(a(w)) = \chi_{\{w_m>0\}}\nabla(w_m - w) + \left(\chi_{\{w_m>0\}} - \chi_{\{w>0\}}\right)\nabla w.$$
It is clear that the first term tends to 0 in $L^2(\Omega, \beta)$:
$$\int_\Omega |\chi_{\{w_m>0\}}\nabla(w_m - w)|^2 d_{\partial\Omega}^{2\beta} \leq \int_\Omega |\nabla(w_m - w)|^2 d_{\partial\Omega}^{2\beta} \to 0.$$

For the second term in (4.13), since $\nabla w = 0$ a.e. on the set $\{w = 0\}$, we can write
$$\int_\Omega \left|\chi_{\{w_m>0\}} - \chi_{\{w>0\}}\right| |\nabla w|^2 d_{\partial\Omega}^{2\beta} = \int_{\{w\neq 0\}} \left|\chi_{\{w_m>0\}} - \chi_{\{w>0\}}\right| |\nabla w|^2 d_{\partial\Omega}^{2\beta}.$$

Over the set $\{w \neq 0\}$ and up to a subsequence, we have $w_m \to w$ a.e., and by the continuity of $a'(t) = \chi_{\{t>0\}}$ for $t \neq 0$, we deduce $\chi_{\{w_m>0\}} \to \chi_{\{w>0\}}$ a.e. on $\{w \neq 0\}$. Since the integrand is bounded by $|\nabla w|^2 d_{\partial\Omega}^{2\beta} \in L^1(\Omega)$, we can therefore apply the dominated convergence theorem to conclude that
$$\int_\Omega \left|\chi_{\{w_m>0\}} - \chi_{\{w>0\}}\right| |\nabla w|^2 d_{\partial\Omega}^{2\beta} \to 0.$$
$\qquad\square$

We have collected all the ingredients to show that the choice (4.10)–(4.11) gives rise to property (H1).

Proposition 4.1 (approximation). *Let the structural condition 1) in Assumption 4.1 hold, and the neural network classes be constructed according to (4.2) by using (4.10) and (4.11). Then, for all $\boldsymbol{p} \in K^{\mathcal{W}}$, we have $d(\boldsymbol{p}, W_m) \to 0$ as $m \to \infty$; in particular, (H1) holds.*

Proof. First, we recall that the desired approximation property is expressed with respect to the norm $\|\cdot\|_{\mathcal{W}}$, given by $\|(u, \phi, \gamma)\|_{\mathcal{W}} = \left(\|\nabla u\|^2 + \|\phi\|^2 + \|\gamma\|^2\right)^{1/2}$.

Let $\boldsymbol{p} = (u, \phi, \gamma) \in K^{\mathcal{W}}$. Since $u - g \in H_0^1(\Omega)$, Lemma 4.5 yields $(u - g)/d_{\partial\Omega} \in H^1(\Omega, 0, 1)$. Thanks to Lemma 4.4, using classical results on H^1 approximation [29, Section 6], along with the fact that $\|\cdot\|_{H^1(\Omega,0,1)} \leq C(\Omega)\|\cdot\|_{H^1(\Omega)}$, we can ensure the existence of a sequence of weights $\{\boldsymbol{\Theta}_v^m\}_{m \in \mathbb{N}}$ such that $\|(u - g)/d_{\partial\Omega} - v_{\boldsymbol{\Theta}_v^m}\|_{H^1(\Omega,0,1)} \to 0$ as $m \to \infty$. By Lemma 4.6, we deduce $a(v_{\boldsymbol{\Theta}_v^m}) \to a\left((u - g)/d_{\partial\Omega}\right)$ in $H^1(\Omega, 0, 1)$. But, as $(u - g)/d_{\partial\Omega} \geq 0$ it follows that $a(w_m) \to (u-g)/d_{\partial\Omega}$ in $H^1(\Omega, 0, 1)$. Thus, we can guarantee the existence of a sequence $\{\boldsymbol{\Theta}_v^m\}_{m \in \mathbb{N}}$ such that

$$(4.14) \qquad \|(u - g)/d_{\partial\Omega} - a(v_{\boldsymbol{\Theta}_v^m})\|_{H^1(\Omega,0,1)} \to 0 \quad \text{as } m \to \infty.$$

We now aim to prove that $\|\nabla\left(u - g - d_{\partial\Omega}a(v_{\boldsymbol{\Theta}_v^m})\right)\| \to 0$ as $m \to \infty$. To this end, we write

$$\nabla\left(u - g - d_{\partial\Omega}a(v_{\boldsymbol{\Theta}_v^m})\right) = \nabla d_{\partial\Omega}\left((u-g)/d_{\partial\Omega} - a(v_{\boldsymbol{\Theta}_v^m})\right) + d_{\partial\Omega}\nabla\left((u-g)/d_{\partial\Omega} - a(v_{\boldsymbol{\Theta}_v^m})\right),$$

so that

$$\|\nabla\left(u - g - d_{\partial\Omega}a(v_{\boldsymbol{\Theta}_v^m})\right)\| \leq \|\nabla d_{\partial\Omega}\left((u-g)/d_{\partial\Omega} - a(v_{\boldsymbol{\Theta}_v^m})\right)\| + \|d_{\partial\Omega}\nabla\left((u-g)/d_{\partial\Omega} - a(v_{\boldsymbol{\Theta}_v^m})\right)\|$$
$$\leq C\|(u-g)/d_{\partial\Omega} - a(v_{\boldsymbol{\Theta}_v^m})\|_{H^1(\Omega,0,1)},$$

with a constant $C > 0$ depending on $\|\nabla d_{\partial\Omega}\|_{L^\infty(\Omega)}$. Using (4.14), we deduce that

$$\|\nabla\left(u - g - d_{\partial\Omega}a(v_{\boldsymbol{\Theta}_v^m})\right)\| \to 0 \text{ as } m \to \infty,$$

which, naming $u_{\boldsymbol{\Theta}_v^m} := g + d_{\partial\Omega}a(v_{\boldsymbol{\Theta}_v^m})$ according to (4.2), yields $u_{\boldsymbol{\Theta}_v^m} \to u$ in $H_0^1(\Omega)$.

Following similar but simpler arguments, we construct the sequences $\{\boldsymbol{\Theta}_\psi^m\}_{m \in \mathbb{N}}$ and $\{\boldsymbol{\Theta}_\eta^m\}_{m \in \mathbb{N}}$ such that $\boldsymbol{\psi}_{\boldsymbol{\Theta}_\psi^m} \to \phi$ in $H^1(\Omega; \mathbb{R}^d)$ and $\eta_{\boldsymbol{\Theta}_\eta^m} \to \gamma - \operatorname{div}\phi$ in $L^2(\Omega)$. Clearly, the former yields $\|\phi - \boldsymbol{\psi}_{\boldsymbol{\Theta}_\psi^m}\| \to 0$ and $\|\operatorname{div}\phi - \operatorname{div}\boldsymbol{\psi}_{\boldsymbol{\Theta}_\psi^m}\| \to 0$ as $m \to \infty$. Furthermore, we can estimate

$$(4.15) \qquad \|\gamma - \operatorname{div}\boldsymbol{\psi}_{\boldsymbol{\Theta}_\psi^m} - a(\eta_{\boldsymbol{\Theta}_\eta^m})\| \leq \|\gamma - \operatorname{div}\phi - a(\eta_{\boldsymbol{\Theta}_\eta^m})\| + \|\operatorname{div}\phi - \operatorname{div}\boldsymbol{\psi}_{\boldsymbol{\Theta}_\psi^m}\|.$$

Since $\boldsymbol{p} = (u, \phi, \gamma) \in K^{\mathcal{W}}$, we have $\gamma - \operatorname{div}\phi \geq 0$. Additionally, as $a(t) = \max\{0, t\}$, and $\eta_{\boldsymbol{\Theta}_\eta^m} \to \gamma - \operatorname{div}\phi$ in $L^2(\Omega)$, we deduce that $\|\gamma - \operatorname{div}\phi - a(\eta_{\boldsymbol{\Theta}_\eta^m})\| \to 0$ as $m \to \infty$. Combining this with estimate (4.15), we conclude that $\|\gamma - \operatorname{div}\boldsymbol{\psi}_{\boldsymbol{\Theta}_\psi^m} - a(\eta_{\boldsymbol{\Theta}_\eta^m})\| \to 0$.

By taking $\{\boldsymbol{\Theta}_m\}_{m \in \mathbb{N}} = \{(\boldsymbol{\Theta}_v^m, \boldsymbol{\Theta}_\psi^m, \boldsymbol{\Theta}_\eta^m)\}_{m \in \mathbb{N}}$ and defining

$$\boldsymbol{p}_{\boldsymbol{\Theta}_m} := \left(g + d_{\partial\Omega}a(v_{\boldsymbol{\Theta}_v^m}), \boldsymbol{\psi}_{\boldsymbol{\Theta}_\psi^m}, \operatorname{div}\boldsymbol{\psi}_{\boldsymbol{\Theta}_\psi^m} + \eta_{\boldsymbol{\Theta}_\eta^m}\right),$$

we find that $\boldsymbol{p}_{\boldsymbol{\Theta}_m} \in W^m$ for all $m \in \mathbb{N}$ and $\boldsymbol{p}_{\boldsymbol{\Theta}_m} \to \boldsymbol{p}$ as $m \to \infty$ in $\mathcal{W}$. $\square$

Next, we show the validity of (H2).

Proposition 4.2 (continuity). *Let the neural network classes be constructed according to (4.2) by using (4.10) and (4.11) and $R > 0$ be given according to (4.3). Then, (H2) holds: the map $\overline{B(0, R)} \to W_m$, $\boldsymbol{\Theta} \mapsto \boldsymbol{p}_{\boldsymbol{\Theta}}$ is continuous.*

Proof. We recall that functions in W_m are constructed from neural network functions in C_m by (4.2),

$$u_{\boldsymbol{\Theta}} = g + d_{\partial\Omega}a(v_{\boldsymbol{\Theta}}), \quad \boldsymbol{\phi}_{\boldsymbol{\Theta}} = \boldsymbol{\psi}_{\boldsymbol{\Theta}}, \quad \gamma_{\boldsymbol{\Theta}} = \operatorname{div}\boldsymbol{\psi}_{\boldsymbol{\Theta}} + a(\eta_{\boldsymbol{\Theta}}), \quad (v_{\boldsymbol{\Theta}}, \boldsymbol{\psi}_{\boldsymbol{\Theta}}, \eta_{\boldsymbol{\Theta}}) \in C_m,$$

and that W_m is equipped from the topology from $H_0^1(\Omega) \times L^2(\Omega; \mathbb{R}^d) \times L^2(\Omega)$.

Our choice of C_m in (4.10) readily implies that the map $\boldsymbol{\Theta} \mapsto (v_{\boldsymbol{\Theta}}, \boldsymbol{\psi}_{\boldsymbol{\Theta}}, \eta_{\boldsymbol{\Theta}})$ is continuous with respect to the $W^{1,\infty}(\Omega) \times W^{1,\infty}(\Omega; \mathbb{R}^d) \times W^{1,\infty}(\Omega)$ norm. This readily implies the continuity of the map $\boldsymbol{\Theta} \mapsto \boldsymbol{\phi}_{\boldsymbol{\Theta}}$ with respect to the $L^2(\Omega; \mathbb{R}^d)$ norm. Furthermore, by applying Lemma 4.6, the continuity of the map $\boldsymbol{\Theta} \mapsto \gamma_{\boldsymbol{\Theta}}$ with respect to the $L^2(\Omega)$ norm is straightforward.

Finally, proving the continuity of the map $\boldsymbol{\Theta} \mapsto u_{\boldsymbol{\Theta}}$ in $H_0^1(\Omega)$ requires showing the continuity of the correspondence $\boldsymbol{\Theta} \to \nabla(d_{\partial\Omega}(a(v_{\boldsymbol{\Theta}})))$ in $L^2(\Omega; \mathbb{R}^d)$. This, in turn, can be proven by splitting

$$\nabla(d_{\partial\Omega}(a(v_{\boldsymbol{\Theta}}))) = \nabla d_{\partial\Omega}\, a(v_{\boldsymbol{\Theta}}) + d_{\partial\Omega}\nabla a(v_{\boldsymbol{\Theta}}),$$

using the uniform boundedness of $d_{\partial\Omega}$ and $\nabla d_{\partial\Omega}$, and Lemma 4.6. $\qquad\square$

4.3.2. *A second example.* In proving Proposition 4.2, we relied on classical neural network approximation properties in the H^1 norm, though only L^2 approximation properties were essential for the term $\eta_{\boldsymbol{\Theta}}$; similarly, the continuity of the map $\boldsymbol{\Theta} \mapsto \eta_{\boldsymbol{\Theta}}$ we showed in Proposition 4.2 is only needed with respect to the L^2 norm. Thus, as $\eta_{\boldsymbol{\Theta}}$ involves no differential operator, its architecture necessitates only L^2 stability and approximation capabilities.

We now focus on the specific construction of the neural network to compute the function $\eta_{\boldsymbol{\Theta}}$, which we recall gives rise to the approximation of the Lagrange multiplier λ in the least squares functional (2.5) through the relation $\lambda = a(\eta) - \operatorname{div}\boldsymbol{\phi}$. More precisely, the remainder of this section provides an analysis of L^2 approximation properties and stability –in the sense of hypotheses (H1) and (H2), respectively– of a certain class of Deep Neural Networks (DNN) that we use in our numerical experiments.

Regarding the approximation property (H1), our proof proceeds in an indirect way by showing that the network is capable of generating the space of piecewise constants finite elements. This, in turn, yields, as a by-product, the L^2 approximation property we need. There are several by now classical results [11, 21, 3] regarding the approximation properties of neural networks, although without the incorporation of boundary conditions. We additionally point out to [13, 19, 20, 37] for recent results concerning upper and lower bounds for the expressive power of deep ReLU networks in the context of approximation in Sobolev spaces. References [20, 1] establish the capability of networks to represent simplicial linear finite element functions, which possess good approximation properties in the H^1-norm. In [1], it is shown that at the expense of a large enough number of neurons, any continuous piecewise linear function in $\mathbb{R}^d$ can be written by using at most $\lceil \log_2(d+1) \rceil$ hidden layers. In [19], on the other hand, piecewise linear finite element basis are explicitly written in terms of ReLU DNN functions with a cost proportional to $\lceil \log_2(\kappa_h) \rceil$, being κ_h the maximum number of elements sharing a nodal point. This, in turn, gives a representation of arbitrary linear finite element functions on regular meshes with an overall cost proportional to the number of nodal points. Moreover, in [20], the article [37] is read in the context of hierarchical finite elements and the authors show, in particular, that finite element basis in 2d can be recovered by means of only two hidden layers.

We introduce a type of DNN that can be regarded as a ReLU DNN in which the first activation function is replaced by a step function σ_s (i.e. $\sigma_s = H$ is the Heaviside function). Namely, functions

in this particular kind of hybrid network, which hereafter we denote by $\mathrm{HNN_k}$, are written as

$$f(x) = \Theta^k \circ \sigma \circ \Theta^{k-1} \circ \sigma \cdots \circ \Theta^1 \circ \sigma_s \circ \Theta^0(x).$$

Here, $\Theta^j : \mathbb{R}^{n_j} \to \mathbb{R}^{n_{j+1}}$, are affine mappings and each activation function is applied componentwise. In the neural network jargon, the number $w = \max_{1 \leq i \leq k}\{n_i\}$ is usually referred to as the width of the network, the total number of neurons $s = \sum_{0 \leq i \leq k} n_i$ is called the neural network size, and the number of hidden layers is k. In our case, $\mathbb{R}^{n_0} = \mathbb{R}^d$ and $\mathbb{R}^{n_{k+1}} = \mathbb{R}$.

Despite possessing good approximation properties, function spaces defined by neural networks with piecewise constant activations pose a significant optimization challenge. Standard gradient-based algorithms fail because the activation derivatives are zero almost everywhere. To circumvent this issue, we employ a variant of the Straight-Through Estimator (STE), a common heuristic for training such networks [22]. We elaborate on the implementation of this approach in Section 5.

Next we show, by means of elementary arguments, that for $k \geq \lceil \log_2(d+1) \rceil$, the set $\mathrm{HNN_k}$ contains a space of piecewise constant functions in $\mathbb{R}^d$, a result that, we believe, could be of interest in itself. Given a bounded, Lipschitz domain $\Omega \subset \mathbb{R}^d$, we consider a partition $\mathcal{T}$ of Ω by means of closed convex polytopes, with the property

$$\overline{\Omega} = \bigcup_{\tau \in \mathcal{T}_h} \tau, \quad \tilde{\tau}^o \cap \tau^o = \emptyset \ \text{ for all } \tilde{\tau}, \tau \in \mathcal{T} \text{ with } \tilde{\tau} \neq \tau.$$

We are interested in exploiting the relation to standard finite element spaces with piecewise constant functions, in particular their ability to approximate functions in L^2 spaces. Thus, in the sequel we assume that each τ –which we will refer to as an element– is a simplex in $\mathbb{R}^d$, namely the convex hull of $d+1$ non collinear points $\{v_0, v_1, \cdots, v_d\}$, called henceforth the vertices of τ; nevertheless, our argument holds for any polytopal partition of Ω.

Following standard finite element notation, we denote the size of an element by $h_\tau = diam(\tau)$, while $h = \max_{\tau \in \mathcal{T}} h_\tau$ stands for the maximum element size in the triangulation $\mathcal{T}$. Given Ω and $\mathcal{T}$, the space P_0 consists of functions that take a constant value on each $\tau \in \mathcal{T}$. In particular, here we assume that these functions are defined in $\mathbb{R}^d$ considering their extension by zero outside Ω.

Proposition 4.3 (recovering characteristics). *Let $d \geq 1$ and τ a simplex in $\mathbb{R}^d$. Then, the characteristic function χ_τ of τ can be written by means of an $\mathrm{HNN_k}$ with $k \sim \mathcal{O}(\lceil \log_2(d+1) \rceil)$ and size $s \sim \mathcal{O}(2(d+1))$.*

Proof. For $d = 1$ we write $\tau = [v_0, v_1]$, where we can assume $v_0 < v_1$. Obviously, $\chi_\tau(x) = H(x - v_0) - H(x - v_1)$ where H is the Heaviside function and then $\chi_\tau \in \mathrm{HNN_1}$. Next, we focus on the case $d = 2$, since for an arbitrary spatial dimension d the argument is the same. Let τ be a triangle of vertices $\{v_0, v_1, v_2\}$. We call L_i the line joining v_i and v_{i+1} (indices modulo 3) and ϕ_i a linear function $\phi_i : \mathbb{R}^2 \to \mathbb{R}$ such that $L_i = \{x : \phi_i(x) = 0\}$; we may assume that $\phi_i(x) > 0$ in the interior of τ. We consider $\mu_i = H \circ \phi_i$ and notice that, by construction, $\mu_i(x) = 1$ for $x \in \tau$ and $i = 0, 1, 2$, while if $x \notin \tau$ then there exists $j \in \{0, 1, 2\}$ such that $\mu_j(x) = 0$. Therefore, we can write

$$\chi_\tau(x) = \min_{0 \leq i \leq 2}\{\mu_i(x)\}.$$

Now, following [20], we use the fact that

$$\min\{a, b\} = \frac{a + b}{2} - \frac{|a - b|}{2} = \boldsymbol{v} \cdot \mathrm{ReLU}\left(W \begin{bmatrix} a \\ b \end{bmatrix}\right),$$

where

$$\boldsymbol{v} = \frac{1}{2}\begin{bmatrix} 1 & -1 & -1 & -1 \end{bmatrix}, \qquad W = \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Since

$$\min\{\mu_0(x), \mu_1(x), \mu_2(x)\} = \min\{\mu_0(x), \min\{\mu_1(x), \mu_2(x)\}\},$$

we immediately observe that

$$\chi_\tau(x) = \boldsymbol{v} \cdot \text{ReLU}\left(W\begin{bmatrix} \mu_0(x) \\ \boldsymbol{v} \cdot \text{ReLU}\left(W\begin{bmatrix} \mu_1(x) \\ \mu_2(x) \end{bmatrix}\right) \end{bmatrix}\right) \in \text{HNN}_3.$$

For a general dimension $d \geq 3$, the argument follows along the same lines by noticing that

$$\min\{\mu_0(x), \mu_1(x), \ldots, \mu_d(x)\} = \min\left\{\min\{\mu_0(x)\ldots, \mu_{\lfloor(d+1)/2\rfloor-1}(x)\}, \min\{\mu_{\lfloor(d+1)/2\rfloor}(x)\ldots, \mu_d(x)\}\right\}$$

and iterating the previous procedure. This construction gives rise to a binary tree with total size $s \sim 2^{\log_2(d+1)+1}$. $\qquad\qquad\square$

Remark 1. A well-known result [29, Section 7], which is similar to Proposition 4.3, is that the characteristic function of any convex polytope can be represented by a two-hidden-layer network using the Heaviside function as an activation function.

For a given triangulation $\mathcal{T}$, the dimension of the space of piecewise constant functions associated to it has the same cardinality as $\mathcal{T}$. Thus, we arrive to the following result.

Corollary 4.1 (recovering piecewise constant functions). *Assume $\Omega \subset \mathbb{R}^d$ is a bounded polytope, and let $\mathcal{T}$ be a triangulation of Ω with $N := \#\mathcal{T}$. Then, the space of piecewise constant functions on τ can be generated by an HNN_k with $k \sim (\lceil log_2(d+1)\rceil)$ and size $s \sim 2N(d+1)$.*

Since the space of piecewise constant functions on a triangulation of Ω is capable of approximation in the L^2-norm, we conclude that (H1) remains valid if we use HNN_k networks for the computation of the variable η.

Corollary 4.2 (approximation with HNN_k). *Assume $\Omega \subset \mathbb{R}^d$ is a bounded polytope. Given $\eta \in L^2(\Omega)$ and $k \sim (\lceil log_2(d+1)\rceil)$, it holds that*

$$\lim_{s\to\infty} \inf_{\eta_\Theta \in \text{HNN}_k} \|\eta - \eta_\Theta\| = 0.$$

Our next goal is to show that the class HNN_k also allows to recover hypothesis (H2): we prove the almost everywhere continuity of elements belonging to HNN_k, in the L^2 norm, with respect to the network parameters. In order to state the next result, we introduce the following notation: for each element $\boldsymbol{\Theta} = (\boldsymbol{a}, b) \in \mathbb{R}^d \times \mathbb{R}$, we define $f_\Theta : \Omega \to \mathbb{R}$ by

$$f_\Theta(x) := H(\boldsymbol{a} \cdot x + b),$$

with H the Heaviside function. We notice that if $\boldsymbol{a} = \boldsymbol{0}$ then $f_\Theta(x)$ degenerates to a constant.

Lemma 4.7 (nondegeneracy). *Let $\Omega \subset \mathbb{R}^d$ be a domain of finite measure, then*

1. *for any $\boldsymbol{\Theta}_0 = (\boldsymbol{a}_0, b_0)$ with $\boldsymbol{a}_0 \neq \boldsymbol{0}$ and any sequence $\{\boldsymbol{\Theta}_n\}_{n\in\mathbb{N}} = \{(\boldsymbol{a}_n, b_n)\}_{n\in\mathbb{N}}$ converging to $\boldsymbol{\Theta}_0$, $f_{\Theta_n} \to f_{\Theta_0}$ almost everywhere in Ω;*

2. if, in addition, Ω is bounded, then for any degenerate parameter $\boldsymbol{\Theta}_0 = (\boldsymbol{0}, b_0)$ there exists a non-degenerate $\boldsymbol{\Theta}' = (\boldsymbol{a}', b')$ (i.e. with $\boldsymbol{a}' \neq \boldsymbol{0}$) such that $f_{\boldsymbol{\Theta}_0} \equiv f_{\boldsymbol{\Theta}'}$ in Ω.

Proof. We first prove *1*. Assuming $\boldsymbol{a}_0 \neq \boldsymbol{0}$, we decompose Ω into three disjoint measurable sets:

$$\Pi^0 := \{x \in \Omega : \boldsymbol{a}_0 \cdot x + b_0 = 0\},$$
$$\Pi^+ := \{x \in \Omega : \boldsymbol{a}_0 \cdot x + b_0 > 0\},$$
$$\Pi^- := \{x \in \Omega : \boldsymbol{a}_0 \cdot x + b_0 < 0\},$$

yielding the partition $\Omega = \Pi^0 \cup \Pi^+ \cup \Pi^-$ into disjoint sets. The non-degeneracy condition $\boldsymbol{a}_0 \neq \boldsymbol{0}$ ensures that Π^0 is contained in an affine hyperplane, and therefore has d-dimensional Lebesgue measure zero.

For each $x \in \Pi^+$, the strict positivity $\boldsymbol{a}_0 \cdot x + b_0 > 0$ implies, by continuity of the inner product, that there exists $N_x \in \mathbb{N}$ such that $\boldsymbol{a}_n \cdot x + b_n > 0$ for all $n \geq N_x$. Consequently, $f_{\boldsymbol{\Theta}_n}(x) = H(\boldsymbol{a}_n \cdot x + b_n) \to 1 = f_{\boldsymbol{\Theta}_0}(x)$ for all $x \in \Pi^+$. An identical argument shows that $f_{\boldsymbol{\Theta}_n}(x) \to 0 = f_{\boldsymbol{\Theta}_0}(x)$ for all $x \in \Pi^-$. Since Π^0 is a null set and the convergence holds pointwise on $\Pi^+ \cup \Pi^-$, we conclude that $f_{\boldsymbol{\Theta}_n} \to f_{\boldsymbol{\Theta}_0}$ almost everywhere in Ω with respect to the Lebesgue measure.

Consider now *2*. In this case, $\boldsymbol{\Theta}_0 = (\boldsymbol{0}, b_0)$. Without loss of generality, we may assume $b_0 \geq 0$, which immediately gives $f_{\boldsymbol{\Theta}_0} \equiv 1$ over Ω. Since Ω is bounded, we can construct an explicit non-degenerate counterpart $\boldsymbol{\Theta}' = (\boldsymbol{1}, b')$ by choosing $\boldsymbol{1} = (1, \ldots, 1) \in \mathbb{R}^d$ and $b' > \sup_{x \in \Omega} \|\boldsymbol{1} \cdot x\|_{L^\infty(\Omega)}$. This guarantees the strict positivity

$$\boldsymbol{1} \cdot x + b' > 0 \quad \text{for all } x \in \Omega,$$

and consequently $f_{\boldsymbol{\Theta}'} \equiv 1$. Thus, the equivalence $f_{\boldsymbol{\Theta}_0} \equiv f_{\boldsymbol{\Theta}'}$ holds in Ω. $\qquad\square$

The next theorem establishes the L^2-continuity of any HNN_k-type neural network with respect to the parameters.

Theorem 4.4 (almost everywhere continuity). *Consider a sequence $\{\boldsymbol{\Theta}_n\}_{n \in \mathbb{N}} \subset \mathbb{R}^m$ converging to some $\boldsymbol{\Theta}_0 \in \mathbb{R}^m$, where $v_{\boldsymbol{\Theta}_n}$ and $v_{\boldsymbol{\Theta}_0}$ are HNN_k-type neural networks parameterized by $\boldsymbol{\Theta}_n$ and $\boldsymbol{\Theta}_0$, respectively. For any domain $\Omega \subset \mathbb{R}^d$ with finite measure, there exists a set $\mathcal{N} \subset \mathbb{R}^m$ such that:*

1. if $\boldsymbol{\Theta}_0 \notin \mathcal{N}$, then $v_{\boldsymbol{\Theta}_n} \to v_{\boldsymbol{\Theta}_0}$ in $L^2(\Omega)$;
2. if Ω is bounded, each $\boldsymbol{\Theta} \in \mathcal{N}$ admits $\boldsymbol{\Theta}' \notin \mathcal{N}$ satisfying $v_{\boldsymbol{\Theta}} \equiv v_{\boldsymbol{\Theta}'}$ in Ω.

Moreover, $\mathcal{N}$ is a finite union of subspaces of $\mathbb{R}^m$, each one of dimension $m - d$ (in particular $\mathcal{N}$ is a null set with respect to the Lebesgue measure).

Proof. For any $\boldsymbol{\Theta} \in \mathbb{R}^m$, let $l_{\boldsymbol{\Theta}} : \Omega \to \mathbb{R}^\ell$ denote the first layer of an HNN_k-type neural network, namely

$$l_{\boldsymbol{\Theta}}(x) := \big(H(\boldsymbol{a}_1 \cdot x + b_1), \ldots, H(\boldsymbol{a}_\ell \cdot x + b_\ell)\big),$$

where $\boldsymbol{a}_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$ for $i = 1, \ldots, \ell$. For $1 \leq i \leq \ell$, consider the sets of singular parameters,

$$\mathcal{N}_i := \{\boldsymbol{\Theta} \in \mathbb{R}^m : \boldsymbol{a}_i = \boldsymbol{0}\}.$$

We note that each $\mathcal{N}_i$ is a proper affine subspace of dimension $m - d$ and define

$$\mathcal{N} := \bigcup_{i=1}^{\ell} \mathcal{N}_i.$$

For $\boldsymbol{\Theta}_0 \in \mathbb{R}^m \setminus \mathcal{N}$, Lemma 4.7 guarantees the componentwise convergence $l_{\boldsymbol{\Theta}_n}(x) \to l_{\boldsymbol{\Theta}_0}(x)$ for almost every $x \in \Omega$. As the remaining layers of the HNN_k architecture consist of continuous activation functions, the almost everywhere convergence is preserved, that is

$$(4.16) \qquad v_{\boldsymbol{\Theta}_n}(x) \to v_{\boldsymbol{\Theta}_0}(x) \quad \text{for a.e. } x \in \Omega.$$

With the same procedure as in the proof of Lemma 4.1, we first establish the uniform boundedness property

$$\sup_{\|\boldsymbol{\Theta}\| \leq R} \|v_{\boldsymbol{\Theta}}\|_{L^\infty(\Omega)} < \infty \quad \text{for any } R > 0.$$

Now, if $\boldsymbol{\Theta}_n \to \boldsymbol{\Theta}_0$, consider the compact parameter set $\{\boldsymbol{\Theta}_0\} \cup \{\boldsymbol{\Theta}_n\}_{n \in \mathbb{N}}$, and let R denote its radius. The preceding estimate yields a uniform bound $M > 0$ satisfying

$$\max \left(\|v_{\boldsymbol{\Theta}_0}\|_{L^\infty(\Omega)}, \sup_{n \in \mathbb{N}} \|v_{\boldsymbol{\Theta}_n}\|_{L^\infty(\Omega)} \right) \leq M.$$

Since Ω has finite measure, the constant function M is integrable and therefore, formula (4.16) together with the dominated convergence theorem yield

$$\lim_{n \to \infty} \|v_{\boldsymbol{\Theta}_n} - v_{\boldsymbol{\Theta}_0}\| = 0,$$

and the first assertion of this theorem follows.

When Ω is bounded, any parameter $\boldsymbol{\Theta} \in \mathcal{N}$ must satisfy $\boldsymbol{a}_i = \boldsymbol{0}$ for some non-empty subset of indices $I \subseteq \{1, \ldots, \ell\}$. Through Lemma 4.7, we may construct modified parameters by replacing each zero-weight pair $(\boldsymbol{a}_i, b_i)_{i \in I}$ with non-degenerate alternatives $(\boldsymbol{a}_i', b_i')_{i \in I}$, where $\boldsymbol{a}_i' \neq \boldsymbol{0}$, to a new parameter $\boldsymbol{\Theta}'$. This construction preserves the layer mapping exactly:

$$l_{\boldsymbol{\Theta}}(x) = l_{\boldsymbol{\Theta}'}(x) \quad \forall x \in \Omega.$$

Consequently, the full network implementations coincide:

$$v_{\boldsymbol{\Theta}}(x) = v_{\boldsymbol{\Theta}'}(x) \quad \forall x \in \Omega,$$

with the key property that $\boldsymbol{\Theta}' \notin \mathcal{N}$ avoids the singular set entirely. $\qquad \square$

Remark 2. Theorem 4.4 establishes that the mapping $\boldsymbol{\Theta} \mapsto \eta_{\boldsymbol{\Theta}}$ is continuous almost everywhere in the parameter space. This result yields two key consequences for networks with step activation functions:

(1) Hypothesis (H2) holds for almost every parameter configuration $\boldsymbol{\Theta} \in \mathbb{R}^m$;
(2) Restricting to parameters within the continuity set does not compromise the expressive capacity of the architecture, as each realizable function $\eta_{\boldsymbol{\Theta}}$ admits an equivalent representation $\eta_{\boldsymbol{\Theta}'}$ with $\boldsymbol{\Theta}'$ in the continuity set.

Notably, the second point guarantees the validity of hypothesis (H1) even when the parameter space is restricted to exclude the negligible set of degenerated points.

Remark 3. While we stated Corollary 4.2 and Theorem 4.4 in the L^2 norm, which is the one we are interested in, it is clear that the same results are valid if one considers the L^p norm for an arbitrary $p \in [1, \infty)$.

5. COMPUTATIONAL EXPERIMENTS

In this section, we present numerical results for the method proposed in Section 3. We did not prioritize any specific neural network architecture, employing two- and three-layer fully connected networks with SoftPlus activation functions to construct u_{Θ} and ϕ_{Θ}. For the construction of η_{Θ}, we used one- to three-layer networks with ReLU or step activation functions. Networks with more layers demonstrated better performance in high-dimensional problems. Across all experiments, we utilized $a(t) := t^2$, which has exhibited satisfactory performance. During training, the well-known ADAM algorithm [24] was employed to update the parameters, with the initial learning rate scheduled to decrease linearly to zero by the final iteration.

When step activation functions are used in the construction of η_{Θ}, gradient computation during training relies on the Straight-Through Estimator [4]. In this approach, the derivative of the Heaviside activation function, which vanishes almost everywhere, is replaced by the approximation $\frac{1}{c}\mathbb{I}_{[0,c]}(x)$ for a prescribed parameter $c > 0$.

Although the convergence theory established in Section 4 employs the functional $\mathcal{L}$, defined in (2.7), our experiments show very similar results using both $\mathcal{L}$ and $\mathcal{J}$, defined in (2.5). That is, we interchangeably used the discrete versions of these functionals, with L_N being the discrete functional defined in (3.1), and J_N being the discrete functional defined as follows:

$$J_N(u, \phi) := \frac{|\Omega|}{N} \sum_{k=1}^{N} \big(\operatorname{div} \phi(x_k) + \lambda(x_k) + f(x_k) \big)^2 + \big(\nabla u(x_k) - \phi(x_k) \big)^2$$
$$+ \lambda(x_k)\big(u(x_k) - g(x_k)\big).$$

For this functional, we can compute u_{Θ} and ϕ_{Θ} as in (4.2), and we can simply use $a(\eta_{\Theta})$ to approximate λ.

The code used to implement the experiments in this section is available at `https://github.com/fbersetche/Deep-FOSLS-for-the-Obstacle-Problem-`.

Example 5.1 (A high-dimensional problem). We consider the following high-dimensional instance of problem (2.4). Let $\Omega = B_{R_0}(0) = \{x \in \mathbb{R}^d : |x| < R_0\}$ with $R_0 \in \mathbb{R}_{>0}$, and let $f_d \colon \mathbb{R} \to \mathbb{R}$ be such that $-\Delta f_d(|x|) = \delta_0$, that is, the d-dimensional fundamental solution of the Laplace operator, as a function of the radius, centered at $x = 0$. Let $0 < r_0 < R_0$, and define the polynomial $Q \colon \mathbb{R} \to \mathbb{R}$ as $Q(r) = ar^4 + br^2 + c$, with coefficients determined to satisfy the conditions

$$Q(r_0) = f_d(r_0) - f_d(R_0), \qquad Q'(r_0) = f_d'(r_0), \qquad Q(R_0) = 0.$$

By taking $g \colon \mathbb{R}^d \to \mathbb{R}$ as $g(x) := Q(|x|)$, the solution to problem (2.4) is

$$u(x) = \begin{cases} Q(|x|) & \text{if } |x| \le r_0, \\ f_d(|x|) - f_d(R_0) & \text{if } r_0 < |x| \le R_0. \end{cases}$$

We first test our approach in a 10-dimensional domain ($d = 10$). Figure 5.1 displays the results we obtained by constructing u_{Θ} and ϕ_{Θ} using 3-layer neural networks with 100 SoftPlus activation functions per layer, and a similar architecture for η_{Θ}, but using ReLU activation functions instead of SoftPlus. The SoftPlus activation function was used with parameter $\beta = 100$. At the end of the optimization algorithm, we computed the average value $L_N(\Theta) = 5.193$ and an average L^2-error of 0.2705.

Figure 5.2 corresponds to $d = 20$, where we used the same architecture with 150 neurons per layer. We observed fast convergence in the number of iterations, reaching an average $L_N(\Theta) = 9702.4$ and an average L^2-error of 8.155 by the end of the minimization algorithm.
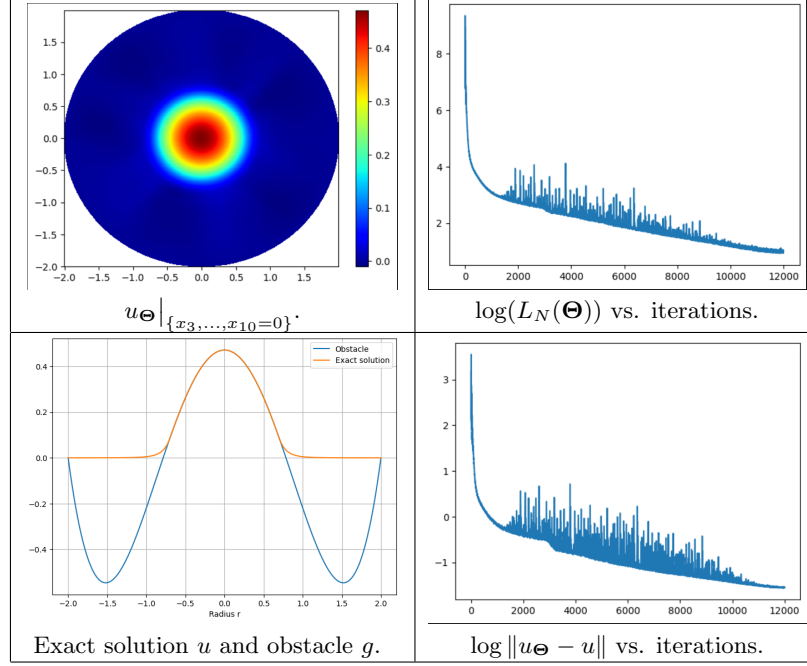

FIGURE 5.1. Top left: A slice of the computational solution u_Θ to example (5.1) in the case $d = 10$, $r_0 = 0.7$, and $R_0 = 2$. In the computation, we used a learning rate $\ell = 0.001$, with 20000 collocation points, 10000 optimization steps, and 65112 degrees of freedom. The bottom left panel exhibits the obstacle and the exact solution as a function of $r = |x|$. We also report the evolution of the loss function (top right) and the L^2-error (bottom right). The L^2-error was estimated using Monte Carlo integration, with a sample of 40000 points.

Example 5.2 (A problem on a slit domain). We consider the non-Lipschitz domain $\Omega = B_1(0) \setminus \{0\} \times [0, 1]$, with $f \equiv 0$, and the a two-peak obstacle function $g : \Omega \to \mathbb{R}$,

$$g(x) = \max\{10e^{-30|x-(-0.4,-0.5)|^2} - 1, 0\} + \max\{15e^{-30|x-(-0.4,0.5)|^2} - 1, 0\}.$$

Figures 5.3 and 5.4 exhibit our computed solutions for this instance of problem (2.4). In this case, we observe a fast convergence towards the solution, reaching $L_N(\Theta) = 42.46$. We employed a two-layer architecture with 150 neurons per layer and a Softplus activation function with parameter $\beta = 100$ for both u_Θ and ϕ_Θ. For η_Θ, we employ a piecewise constant architecture comprising a three-layer HNN_3, where the first and third layers use the ReLU activation function, and the second layer adopts the step activation function (see Proposition 4.3). During training, the STE is used to approximate the gradient with respect to the parameters of η_Θ by substituting the derivative of the Heaviside activation functions with $\frac{1}{c}\mathbb{I}_{[0,c]}(x)$, where $c = 0.5$. Our results indicate that the method
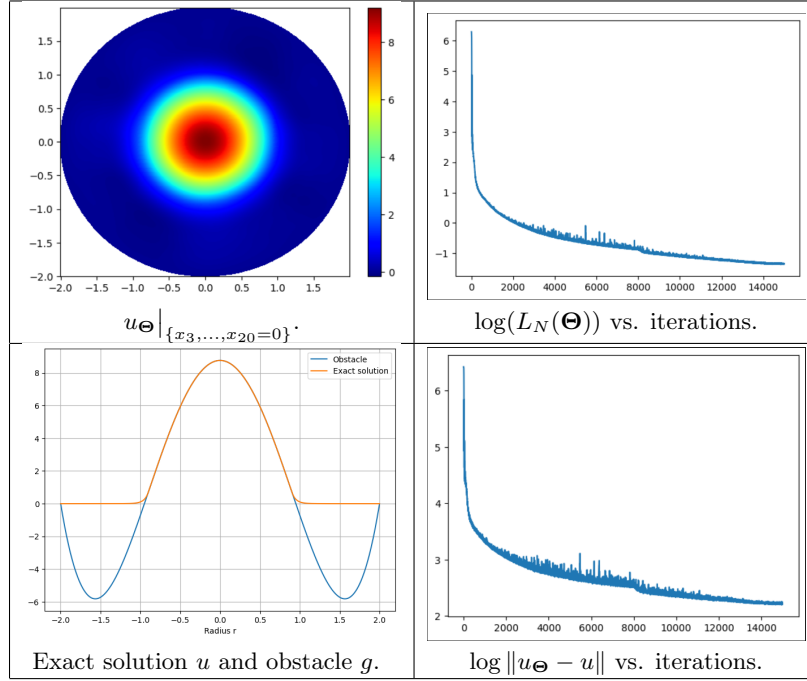
$u_{\boldsymbol{\Theta}}\big|_{\{x_3,\ldots,x_{20}=0\}}.$

$\log(L_N(\boldsymbol{\Theta}))$ vs. iterations.

Exact solution u and obstacle g.

$\log\|u_{\boldsymbol{\Theta}} - u\|$ vs. iterations.

FIGURE 5.2. A slice of the computational solution $u_{\boldsymbol{\Theta}}$ (top left), evolution of the loss function (top right), the exact solution and the obstacle as a function of $r = \|x\|_2$ (bottom left), and evolution of the L^2-error (bottom right) for (5.1) with $d = 20$, $r_0 = 0.9$, and $R_0 = 2$. We used a learning rate $\ell = 0.001$, with 20000 collocation points, 14000 optimization steps, and 148672 degrees of freedom.

performs robustly even on non-Lipschitz domains, which lie outside the scope of the theoretical framework.

REFERENCES

[1] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018.

[2] H. E. Bahja, J. C. Hauffen, P. Jung, B. Bah, and I. Karambal. A physics-informed neural network framework for modeling obstacle-related equations. *arXiv preprint arXiv:2304.03552*, 2023.

[3] A. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[4] Y. Bengio, N. Léonard, and A. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR 2013)*, 2013.

[5] J. Berg and K. Nyström. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing*, 317:28–41, 2018.

[6] F. Bersetche and J. Borthagaray. A deep first-order system least squares method for solving elliptic PDEs. *Computers & Mathematics with Applications*, 129:136–150, 2023.

[7] A. Braides. A handbook of Γ-convergence. In *Handbook of Differential Equations: stationary partial differential equations*, volume 3, pages 101–213. Elsevier, 2006.
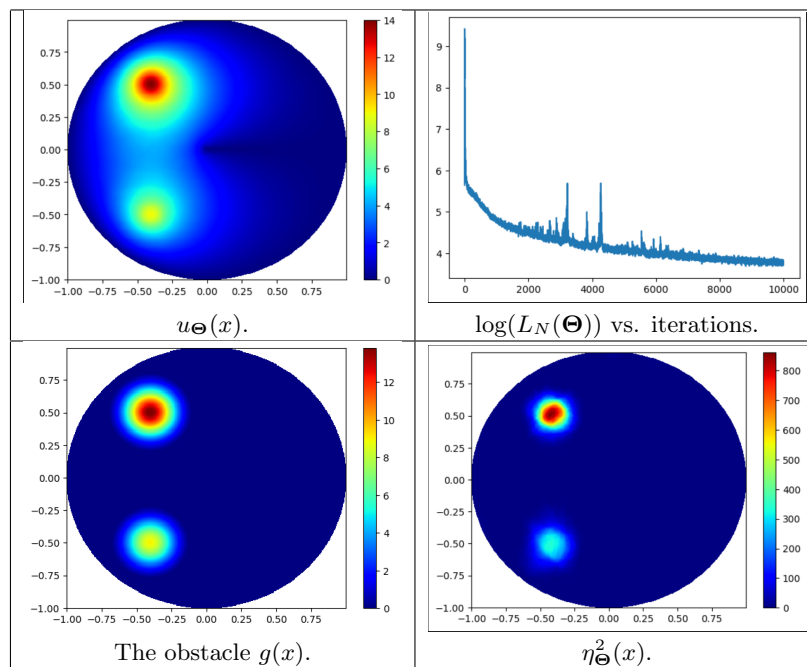
FIGURE 5.3.   Computational solution u_{Θ} (top left), evolution of the loss function (top right), the obstacle function g (bottom left), and the computed contact set η_{Θ}^2 (bottom right). We used a learning rate $\ell = 0.001$, with 4000 collocation points, 10000 optimization steps, and 66654 degrees of freedom. In this case, the surrogate function $d_{\partial\Omega}(x)$ agrees with $d(x, \partial\Omega)$.

[8] X. Cheng, X. Shen, X. Wang, and K. Liang. A deep neural network-based method for solving obstacle problems. *Nonlinear Anal. Real World Appl.*, 72:103864, 2023.

[9] F. Chouly, P. Hild, and Y. Renard. *Finite element approximation of contact and friction in elasticity*. Springer, 2023.

[10] P. G. Ciarlet. *The finite element method for elliptic problems*. SIAM, 2002.

[11] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[12] M. Darehmiraki. A deep learning approach for the obstacle problem. In *Proceedings of Academia-Industry Consortium for Data Science: AICDS 2020*, pages 179–188. Springer, 2022.

[13] R. DeVore, B. Hanin, and G. Petrova. Neural network approximation. *Acta Numer.*, 30:327–444, 2021.

[14] W. E and S. Wojtowytsch. Some observations on high-dimensional partial differential equations with Barron data. In *Mathematical and Scientific Machine Learning*, pages 253–269. PMLR, 2022.

[15] W. E and B. Yu. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[16] T. Führer. First-order least-squares method for the obstacle problem. *Numerische Mathematik*, 144(1):55–88, 2020.

[17] R. Glowinski. *Numerical methods for nonlinear variational problems*. Springer Science & Business Media, 2013.

[18] C. He, X. Hu, and L. Mu. A mesh-free method using piecewise deep neural network for elliptic interface problems. *Journal of Computational and Applied Mathematics*, 412:114358, 2022.

[19] J. He, L. Li, and J. Xu. ReLU deep neural networks from the hierarchical basis perspective. *Comput. Math. Appl.*, 120:105–114, 2022.
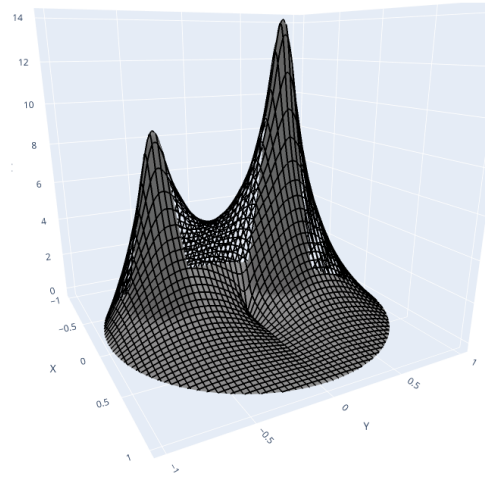
FIGURE 5.4. The computational solution u_Θ (black wire-frame), and the obstacle function g (gray), in Example 5.2.

[20] J. He, L. Li, J. Xu, and C. Zheng. Relu deep neural networks and linear finite elements. *J. Comput. Math.*, 38(3):502–527, 2020.

[21] K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[22] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio. Quantized neural networks: training neural networks with low precision weights and activations. *J. Mach. Learn. Res.*, 18:30, 2018. Id/No 187.

[23] D. Kinderlehrer and G. Stampacchia. *An introduction to variational inequalities and their applications*. SIAM, 2000.

[24] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *In Proceedings of the 3rd InternationalConference for Learning Representations—ICLR*, pages 7–9, San Diego, CA, 2015.

[25] A. Kufner. Weighted Sobolev spaces. Licensed ed. A Wiley-Interscience Publication. Chichester etc.: John Wiley & Sons. 116 p. £ 15.00 (1985)., 1985.

[26] M. Liu and Z. Cai. Adaptive two-layer ReLU neural network: II. Ritz approximation to elliptic PDEs. *Comput. Math. Appl.*, 113:103–116, 2022.

[27] M. Liu, Z. Cai, and J. Chen. Adaptive two-layer ReLU neural network: I. best least-squares approximation. *Comput. Math. Appl.*, 113:34–44, 2022.

[28] J. A. Opschoor, P. C. Petersen, and C. Schwab. First order system least squares neural networks. *arXiv preprint arXiv:2409.20264*, 2024.

[29] A. Pinkus. Approximation theory of the MLP model in neural networks. In *Acta Numerica Vol. 8, 1999*, pages 143–195. Cambridge: Cambridge University Press, 1999.

[30] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.

[31] J.-F. Rodrigues. *Obstacle problems in mathematical physics*. Elsevier, 1987.

[32] J. Sirignano and K. Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

[33] N. Sukumar and A. Srivastava. Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks. *Comput. Methods Appl. Mech. Eng*, 389:114333, 2022.

[34] R. Trémolieres, J.-L. Lions, and R. Glowinski. *Numerical analysis of variational inequalities.* Elsevier, 2011.

[35] S. Wojtowytsch and W. E. Can shallow neural networks beat the curse of dimensionality? A mean field training perspective. *IEEE Trans. Artif. Intell.*, 1(2):121–129, 2020.

[36] J. Xu. Finite neuron method and convergence analysis. *Commun. Comput. Phys.*, 28:1707–1745, 2020.

[37] D. Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

[38] X. E. Zhao, W. Hao, and B. Hu. Two neural-network-based methods for solving elliptic obstacle problems. *Chaos Solitons Fractals*, 161:112313, 2022.

(G. Acosta) Departamento de Matemática, FCEyN, Universidad de Buenos Aires / IMAS, CONICET, Buenos Aires, Argentina

Email address: `gacosta@dm.uba.ar`

(E. Belén) Departamento de Matemática, FCEyN, Universidad de Buenos Aires, Buenos Aires, Argentina

Email address: `beleneugenia.96@gmail.com`

(F.M. Bersetche) Departamento de Matemática, FCEyN, Universidad de Buenos Aires, Buenos Aires, Argentina

Email address: `fbersetche@dm.uba.ar`

(J.P. Borthagaray) Instituto de Matemática y Estadística "Rafael Laguardia", Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

Email address: `jpborthagaray@fing.edu.uy`