WAN DISCRETIZATION OF PDES: BEST APPROXIMATION, STABILIZATION AND ESSENTIAL BOUNDARY CONDITIONS*

SILVIA BERTOLUZZA†, ERIK BURMAN‡, AND CUIYU HE §

Abstract. In this paper, we provide a theoretical analysis of the recently introduced weakly adversarial networks (WAN) method, used to approximate partial differential equations in high dimensions. We address the existence and stability of the solution, as well as approximation bounds. We also propose two new stabilized WAN-based formulas that avoid the need for direct normalization. Furthermore, we analyze the method's effectiveness for the Dirichlet boundary problem that employs the implicit representation of the geometry. We also devise a pseudo-time XNODE neural network for static PDE problems, yielding significantly faster convergence results than the classical DNN.

Key words. Weak Adversarial Network, Cea's Lemma, Numerical PDE, Pseudo-time XNODE

MSC codes. 65M12, 65N12

1. Introduction.

Recently there has been a vast interest in approximating partial differential equations (PDE) using neural networks and machine learning techniques. In this note, we will consider the weak adversarial networks (WAN) method introduced by Zang et al. [22]. The idea is to rewrite the weak form of the PDE as a saddle point problem whose solution is obtained by approximating both the trial (primal) and the test (adversarial) space through neural networks. In [22], the method was tested on various PDEs, tackling different challenging issues such as high dimension, nonlinearity, and nonconvexity of the domain. It was subsequently applied for the inverse problems in high dimension [1] and for the parabolic problems [16], with quite promising results.

However, as often happens for neural network methods for numerical PDEs, rigorous theoretical results on the capability of WANs to approximate the solution of a given PDE still need to be improved. The most critical issues must be addressed are the discrete solution's existence, stability, and approximation properties. Due to the inherent nature of neural network function classes, even the issue of the existence of a discrete solution is far from a trivial one. Indeed, fixed architecture neural network classes are generally neither convex nor closed [18, 14]. Therefore, a global minimum for a cost functional in one of such classes might not exist. Unsurprisingly, as we are ultimately dealing with a saddle point problem, a suitable choice of the test (adversarial) network class will play a vital role in the analysis. The lack of linearity of the trial (primal) network class will imply the need for a strengthened inf-sup condition (see (2.14) in the following), which, however, will not, in general, be enough to guarantee the existence and uniqueness of a global minimizer. Indeed, due to the non-closedness of neural network classes, it might not be possible to attain the minimum with an element belonging to the class.

What we can prove, under suitable assumptions (see (2.6) and (2.14)), in a general

*Submitted to SIAM Journal of Scientific Computing.

Funding: EB was partially funded by EPSRC grants EP/W007460/1, EP/V050400/1 and EP/T033126/1.

†Istituto di Matematica Applicata e Tecnologie Informatiche, CNR, Italy, (silvia.bertoluzza@imati.cnr.it).

‡Department of Mathematics, University College London, UK, (e.burman@ucl.ac.uk).

§Department of Mathematics, Oklahoma State University, 401 Stillwater, OK, 74078 (cuiyu.he@okstate.edu).// Department of Mathematics, University of Georgia, Athens, GA 30602 (cuiyu.he@uga.edu)

abstract framework, is (1) the existence of at least one weakly converging minimizing sequence for the WAN cost functional, and, (2) that all weak limits of weakly converging minimizing sequences satisfy a quasi-optimality bound similar to Céa's Lemma. More importantly, we further prove that a similar approximation bound will hold for the elements of the minimizing sequences sufficiently close to convergence. Combined with approximation bounds by the deep neural networks [10], this will guarantee that the WAN can, in principle, provide an arbitrarily good approximation to the continuous PDE solution. Another crucial issue relates to the convergence of the optimization scheme used to solve the minimization problem. Also this task is made difficult by the inherent topological properties of neural network classes. It is worth mentioning (see [18]) that the function class of Deep Neural Networks (DNN) lacks inverse stability in the L^p and $W^{s,p}$ norms. In simple terms, the norm of the elements of the DNN function class does not control the norm of the associated parameter vector. As the optimization schemes indirectly act on the function class through the parameter space, this will negatively affect the minimization process. In particular, when, the weak limit of the minimizing sequence does not belong to the function class, it can be proved that the sequence of the Euclidean norms of the corresponding parameter vectors explodes [18].

In the WAN framework, aforementioned problems are integrated with the problems related to the inexact evaluation of the cost functional, which is defined as a supremum over the elements of the adversarial network and requires solving an optimization problem that, for the classical WAN method, become ill-posed due to the presence of direct normalization, and is therefore subject to a possibly relevant error. If this error becomes comparable, or even dominant, compared to the value of the cost functional itself, the overall optimization procedure will lose effectiveness and likely display oscillations. To mitigate this phenomenon, developing more stable and accurate methods for evaluating the operator norm is crucial. In the framework of WAN, we propose two alternative ways of evaluating the operator, that avoid direct normalization and improve the overall convergence of the minimization procedure.

We then exploit the results for the second-order elliptic PDEs with essential boundary conditions. These are notoriously challenging as the construction of neural networks exactly vanishing on the boundary of a domain is extremely difficult, if not impossible. On the other hand, standard techniques, such as Nitsche's method, that impose Dirichlet boundary conditions weakly, rely on inverse inequalities that do not generally hold in the neural network framework. Adapting a strategy introduced, for finite elements, in [7], we propose to approximate the test space $H_0^1(\Omega)$ with a class of functions obtained by multiplying the elements of a given neural network class with a level set type weight, thus strongly enforcing the homogeneous boundary conditions on the test function class. Non-homogeneous boundary conditions are then imposed by penalization with a suitable boundary norm. We can show that the resulting discrete schemes fall in our abstract setting, thus obtaining Céa's Lemma type quasi-optimal H^1 error bounds.

As the architecture of neural networks plays a crucial role in their performance, we test the newly proposed methods on different function classes of various structures. In particular, besides DNN, we focus on residual-related networks, whose usage [11] was initially proposed to enhance image processing capabilities. These networks have also found application in various domains, including numerical PDEs [13, 23]. In a recent work by Oliva et al. [16], the XNODE network is proposed to solve parabolic equations. Numerical experiments have demonstrated that, compared to classical DNN networks, XNODE, can substantially reduce the number of iterations required for optimization. This rapid convergence can be attributed to the structure of the XNODE model, which emulates a residual network, and to the direct incorporation of the initial condition into the model. Besides testing our framework with XNODE architecture on a parabolic problem, we also introduce a new variant of the XNODE network, which we refer to as the pseudo-time XNODE method for stationary problems. Remarkably fast convergence is observed in the numerical results, even for nonlinear and high-dimensional static elliptic PDEs.

The paper is organized as follows. In section 2, we prove quasi-best approximation results and in Section 3 we propose two more stable equivalent formulations. In section 4, we leverage our approach to allow for Dirichlet boundary conditions. Finally, the numerical results are provided in section 6. We devote the remaining part of this section to discuss the standard WAN in an abstract setting. Our framework covers a large class of problems without symmetry or coercivity assumptions, allowing for standard well-posed problems and certain non-standard data assimilation problems. We also cover a very general class of discretization spaces: while we have in mind neural networks, the only a priori assumptions that we make on our trial and test function classes is that they are function sets containing the identically vanishing function so that our results potentially applies to a much wider range of methods, provided that the inf-sup conditions (2.6) and (2.14) hold.

Throughout the paper, we assume that all forms, linear and bilinear, are evaluated exactly and that the resulting nonlinear optimization problems can be solved with sufficient accuracy. Needless to say, these problems are crucial for the actual performance of the method. Nevertheless, the quasi-best approximation results proved herein are a cornerstone for its reliability.

1.1. The abstract setting. We consider a PDE set in some open, connected set $\Omega \subset \mathbb{R}^d$ ($d \geqslant 1$). We assume that the problem can be cast in the following general abstract weak form. Let W and V be two reflexive separable Banach spaces. Define a bounded bilinear form $\mathcal{A} : W \times V \mapsto \mathbb{R}$, satisfying

$$\mathcal{A}(w, v) \leqslant M \|w\|_W \|v\|_V, \quad \forall w \in W, v \in V, \tag{1.1}$$

and let $\mathcal{F} : V \mapsto \mathbb{R}$ be a bounded linear form. We consider the abstract problem: find $u \in W$ such that

$$\mathcal{A}(u, v) = \mathcal{F}(v), \quad \forall v \in V. \tag{1.2}$$

As in [1], we rewrite (1.2) as the following minimization problem

$$u = \operatorname*{argmin}_{w \in W} \sup_{v \in V, v \neq 0} \frac{\mathcal{F}(v) - \mathcal{A}(w, v)}{\|v\|_V} = \operatorname*{argmin}_{w \in W} \|u - w\|_{op}, \tag{1.3}$$

where we define

$$\|w\|_{op} := \sup_{v \in V, v \neq 0} \frac{\mathcal{A}(w, v)}{\|v\|_V}. \tag{1.4}$$

We assume (1.2) admits a unique solution, satisfying the following stability estimate

$$\|u\|_W \leqslant C \|\mathcal{F}\|_{V'}. \tag{1.5}$$

This is for instance the case if the form satisfies the assumptions of the Banach-Necas-Babuska theorem, or if it satisfies the more general condition of the Lions theorem, complemented by suitable compatibility conditions on $\mathcal{F}$ (see [8, Theorem 2.6 and Lemma A.40]). It is straightforward to show that, under such an assumption, the solution of problem (1.2) coincides with the unique minimizer of (1.3).

In principle, for any function class $W_{\boldsymbol{\theta}}$, parametrized by a parameter set $\mathcal{P}_{\boldsymbol{\theta}}$, we can approximate the solution u by solving the semi-discrete problem:

$$\tilde{u}^*_{\boldsymbol{\theta}} = \underset{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}}{\operatorname{argmin}} \|u - w_{\boldsymbol{\theta}}\|_{op}. \tag{1.6}$$

Remark that allowing the test space V to be different from the space W, where the solution is sought, makes the above formulation extremely flexible, allowing it to cover a wide range of situations, such as the ones where a partial differential equation, written in the form

$$a(u, w) = f(w), \qquad \forall w \in W_0 \subset W, \tag{1.7}$$

(W_0 denoting some closed subspace of W), is complemented by a constraint:

$$b(u, \chi) = g(\chi), \qquad \forall \chi \in X, \tag{1.8}$$

where X is a third reflexive separable Banach space. Such a situation falls in our abstract framework, with $V = W_0 \times X$, if we set, for $v = (w_0, \chi) \in V$,

$$\mathcal{A}(w, v) = a(w, w_0) + \beta b(w, \chi), \qquad \mathcal{F}(v) = f(w_0) + \beta g(\chi),$$

(β being a parameter weighting the constraint with respect to the equation). In such a case, the $\|\cdot\|_{op}$ norm satisfies

$$\|w\|_{op} = \sup_{(v,\chi)\in W_0\times X} \frac{\mathcal{A}(w,(v,\chi))}{(\|v\|^2_{W_0} + \|\chi\|^2_X)^{1/2}} \simeq \sup_{v\in W_0} \frac{a(w,v)}{\|v\|_{W_0}} + \beta \sup_{\chi\in X} \frac{b(w,\chi)}{\|\chi\|_X}.$$

Typically, as we shall see below, (1.8) could represent the imposition of essential boundary conditions. It could also represent some other form of constraint, such as the ones encountered in data assimilation problems subject to the heat or wave equation (see [2] or [3] where b is the L^2-scalar product over some subset $\omega \subset \Omega$ [4, 15]).

1.2. The WAN method. Let W denote, throughout this section, the $H^1(\Omega)$ space with norm $\|v\|_W$ defined as $\|v\|^2_W = (\nabla v, \nabla v)_\Omega + (v, v)_\Omega$, where $(\cdot,\cdot)_\Omega$ denotes the $L^2(\Omega)$ scalar product. We consider an elliptic partial differential equation, endowed with a Dirichlet boundary condition, that we write in the form

$$A(u) = f, \qquad B(u) = g, \tag{1.9}$$

where A is a second order partial differential operator and B is the trace operator. Bao et al. propose in [1] to rewrite (1.9) as a minimization problem in a suitable dual space. To this aim, the so-called operator norm is introduced, defined as

$$\|A(v)\|_{H^{-1}(\Omega),op} := \sup_{\substack{\varphi\in H^1_0(\Omega)\\ \varphi\neq 0}} \frac{a(v,\varphi)}{\|\varphi\|_W}, \tag{1.10}$$

where $a : H^1(\Omega) \times H^1_0(\Omega) \mapsto \mathbb{R}$, $a(w, \varphi) = (A(u), \varphi)_\Omega$, is the bilinear form corresponding to the operator A. We immediately see that, provided the form a is continuous on $H^1(\Omega) \times H^1_0(\Omega)$, such norm is well defined (indeed, it coincides with the standard $H^{-1}(\Omega)$ norm). The idea of [1] was then to combine the residual in such a norm with a boundary penalization term aimed at weakly imposing the boundary conditions (rather than enforcing them exactly), and consider the following minimization problem:

$$u^* = \operatorname*{argmin}_{w \in W} \left(\|A(u-w)\|_{H^{-1}(\Omega),op} + \beta \|g - w\|_{L^2(\partial\Omega)} \right). \tag{1.11}$$

Setting $V = H^1_0(\Omega) \times L^2(\partial\Omega)$, and

$$\mathcal{A}(w,[v,\chi]) = a(w,v) + \beta(w,\chi)_{\partial\Omega}, \qquad \mathcal{F}([v,\chi]) = (f,v)_\Omega + \beta(g,\chi)_{\partial\Omega},$$

this problem can be rewritten in the form (1.3). At the continuous level, problem (1.11) is, in some sense, equivalent to (1.9). Indeed, we observe that the unique solution of (1.9) annihilates both $\|A(u-w)\|_{H^{-1}(\Omega),op}$ and $\|g - w\|_{L^2(\partial\Omega)}$, implying existence. Then, the values of the minimum is zero, and any other $H^1(\Omega)$ function minimizing the boundary penalized residual can be easily seen to be the solution to (1.9), thus obtaining uniqueness. Trivially, as it coincides with the solution of (1.9), the solution of (1.11) satisfies $\|u^*\|_W \lesssim \|f\|_{H^{-1}(\Omega)} + \|g\|_{H^{1/2}(\partial\Omega)} \lesssim \|f\|_{H^{-1}(\Omega)} + C(g)\|g\|_{L^2(\partial\Omega)}$, with $C(g) = \|g\|_{H^{1/2}(\partial\Omega)}/\|g\|_{L^2(\partial\Omega)}$, which is a stability bound of the form (1.5), though with a constant depending on g (whether such a constant is large or not depends on the frequency content of g: if g is not oscillating, such a constant is of order one, but it can be large if g presents high frequency oscillations). However, such a formulation does not entirely fall in the abstract setting of Section 1.1, since, for $V = H^1_0(\Omega) \times L^2(\partial\Omega)$, the bilinear form $\mathcal{A}$ does not satisfy the boundedness assumption (1.1). It is therefore natural to consider the following minimization problem, where the boundary penalization term is measured in the $H^{1/2}(\partial(\Omega)) = (H^{-1/2}(\partial\Omega))'$ norm

$$u^* = \operatorname*{argmin}_{\mathrm{w} \in \mathrm{W}} \left(\|A(u-w)\|_{H^{-1}(\Omega),op} + \beta \|g - w\|_{H^{1/2}(\partial\Omega)} \right). \tag{1.12}$$

It is not difficult to see that also this problem can be written in the form (1.3), this time with $V = H^1_0(\Omega) \times H^{-1/2}(\partial\Omega)$. Thanks to the choice of the correct norm for the boundary penalization term, problem (1.12) falls within our abstract framework of subsection 1.1, it is well posed, and equivalent to (1.9). It will serve as a starting point for the boundary condition treatment we will propose in section 4.

Remark 1.1. In the very first version of the WAN method, see [22], the authors actually proposed a different definition of the operator norm, namely they defined the dual norm involved in the minimization problem as

$$\|A(v)\|_{L^2(\Omega),op} := \sup_{\varphi \in H^1_0(\Omega), \varphi \neq 0} \frac{a(v,\varphi)}{\|\varphi\|_{L^2(\Omega)}}.$$

It should be noted that this norm is not generally well defined at the continuous level, and to remedy this, the different normalization in (1.10) was proposed in [1]. We remark that the notation used for such a norm in [22] was $\|A(v)\|_{op}$, while we use the notation $\|\cdot\|_{op}$ with a different meaning, see (1.4).

Remark 1.2. We remark that replacing the natural norms $H^1(\Omega)$ and $H^{1/2}(\partial\Omega)$ in, respectively, (1.10) and (1.11) with the corresponding L^2-norm results in two "variational crimes" with fairly different features. In both cases, the natural norm is replaced by a weaker norm but in the first case the replacement happens in the denominator. The resulting term $\|A(w)\|_{L^2(\Omega),op}$, $w \in W$, is not necessarily well defined (as this would require $w \in H^2(\Omega)$). This is essentially the same residual quantity as that minimized in so-called PINN methods [5, 19]. In the second case, the "variational crime" is somewhat less severe: all the quantities involved in the minimization problem (1.11) are well defined, though, as we already pointed out, the boundedness assumption (1.1) does not hold.

In the WAN method, the discretization for either (1.11) or (1.12) is performed by replacing the spaces $H^1(\Omega)$ and $H^1_0(\Omega)$ by, respectively, their discrete counterparts $W_{\boldsymbol{\theta}} \subset H^1(\Omega)$ and $V_{\boldsymbol{\eta}} \subset H^1_0(\Omega)$, where $W_{\boldsymbol{\theta}}$ and $V_{\boldsymbol{\eta}}$ are two fixed architecture neural network function classes, parameterized by parameter sets $\mathcal{P}_{\boldsymbol{\theta}}$ and $\mathcal{P}_{\boldsymbol{\eta}}$. The discretization is carried out via a discrete operator norm, defined, for any $w \in H^1(\Omega)$, as

$$\|A(w)\|_{H^{-1}(\Omega),op,\boldsymbol{\eta}} := \sup_{\substack{v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}} \\ \|v_{\boldsymbol{\eta}}\|_V \neq 0}} \frac{a(w, v_{\boldsymbol{\eta}})}{\|v_{\boldsymbol{\eta}}\|_V}. \tag{1.13}$$

The discrete method can then be written, for X being either $L^2(\partial\Omega)$ or $H^{1/2}(\partial\Omega)$,

$$u^*_{\boldsymbol{\theta}} = \operatorname*{argmin}_{\mathrm{w}_{\boldsymbol{\theta}} \in \mathrm{W}_{\boldsymbol{\theta}}} \left(\|A(u - w_{\boldsymbol{\theta}})\|_{H^{-1}(\Omega),op,\boldsymbol{\eta}} + \beta \|w_{\boldsymbol{\theta}} - g\|_X \right). \tag{1.14}$$

Exactly evaluating the functional on the right-hand side is very difficult since functions in $W_{\boldsymbol{\theta}}$ and $V_{\boldsymbol{\eta}}$, may have very different geometric structures. In practice, the integrals are approximated using fixed sample points or a Monte Carlo integration method [12]. The optimization is then performed using a Stochastic gradient descent method, e.g. Adam, over the parameter sets $\mathcal{P}_{\boldsymbol{\theta}}$ and $\mathcal{P}_{\boldsymbol{\eta}}$. We also note that, due to the normalization in (1.13), when w is close to u, the maximization problem in $v_{\boldsymbol{\eta}}$ becomes ill-posed, resulting in increased undesirable oscillations. We will propose a possible remedy in Section 3.

2. Analysis of the WAN method. This section will frame and analyze the WAN method in an abstract framework. We aim to provide insight into choosing the approximation and adversarial networks to ensure the resulting method's stability and optimality. For simplicity, we will perform the analysis based on (1.14) without considering the errors caused by the Monte Carlo and gradient descent methods.

We define the WAN method in the abstract framework as follows. Letting $V_{\boldsymbol{\eta}} \subset V$ denote a function class parametrized by a parameter set $\mathcal{P}_{\boldsymbol{\eta}}$, we introduce the discrete version of the $\|\cdot\|_{op}$ norm on W, defined as

$$\|w\|_{op,\boldsymbol{\eta}} := \sup_{\substack{v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}} \\ \|v_{\boldsymbol{\eta}}\|_V \neq 0}} \frac{\mathcal{A}(w, v_{\boldsymbol{\eta}})}{\|v_{\boldsymbol{\eta}}\|_V}. \tag{2.1}$$

We observe that, for all $w \in W$ we have that

$$\|w\|_{op,\boldsymbol{\eta}} \leqslant \|w\|_{op} \leqslant M\|w\|_W. \tag{2.2}$$

The fully discrete problem then reads

$$u^*_{\boldsymbol{\theta}} = \operatorname*{argmin}_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \|u - w_{\boldsymbol{\theta}}\|_{op,\boldsymbol{\eta}}. \tag{2.3}$$

In our analysis, a key role will be played by the function class of differences of elements of the approximation network $W_{\boldsymbol{\theta}}$:

$$S_{\boldsymbol{\theta}} := \{w_{1,\boldsymbol{\theta}} - w_{2,\boldsymbol{\theta}},\ w_{1,\boldsymbol{\theta}}, w_{2,\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}\}. \tag{2.4}$$

We will first consider the case of coercive problems and then tackle problems only known to satisfy the stability (1.5).

2.1. Coercive problems. Let us at first consider the case $V = W$, and assume that the bilinear form $\mathcal{A}$ is coercive, i.e., there exist $\alpha > 0$ such that

$$\alpha \|\phi\|_W^2 \leqslant \mathcal{A}(\phi, \phi). \tag{2.5}$$

We make the following assumption on the networks $W_{\boldsymbol{\theta}}$ and $V_{\boldsymbol{\eta}}$:

$$W_{\boldsymbol{\theta}} \cup S_{\boldsymbol{\theta}} \subseteq V_{\boldsymbol{\eta}}. \tag{2.6}$$

Observe that if $0 \in W_{\boldsymbol{\theta}}$, we have that $W_{\boldsymbol{\theta}} \cup S_{\boldsymbol{\theta}} = S_{\boldsymbol{\theta}}$. We start by remarking that, as the functional $w \to \|u - w\|_{op,\boldsymbol{\eta}}$, with $u \in W$ given, is bounded from below by 0, we have that

$$\sigma^* := \inf_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \|u - w_{\boldsymbol{\theta}}\|_{op,\boldsymbol{\eta}} \geqslant 0.$$

By the definition of infimum, there exist a sequence $\{w_{\boldsymbol{\theta}}^n\}$ with $w_{\boldsymbol{\theta}}^n \in W_{\boldsymbol{\theta}}$ such that

$$\lim_{n\to\infty} \|u - w_{\boldsymbol{\theta}}^n\|_{op,\boldsymbol{\eta}} = \inf_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \|u - w_{\boldsymbol{\theta}}\|_{op,\boldsymbol{\eta}}. \tag{2.7}$$

We call a sequence satisfying (2.7) a minimizing sequence for (2.3). We have the following lemma, where $\mathrm{cl}_w^{seq}(W_{\boldsymbol{\theta}}) \subseteq W$ denotes the weak sequential closure of $W_{\boldsymbol{\theta}}$ in W(see [17]).

LEMMA 1. *Let $\{w_{\boldsymbol{\theta}}^n\}$ be a minimizing sequence for* (2.3)*. Then, under assumption* (2.6)*, there exists a subsequence weakly converging to an element $u_{\boldsymbol{\theta}}^* \in \mathrm{cl}_w^{seq}(W_{\boldsymbol{\theta}})$ satisfying*

$$\|u - u_{\boldsymbol{\theta}}^*\|_{op,\boldsymbol{\eta}} \leqslant \inf_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \|u - w_{\boldsymbol{\theta}}\|_{op,\boldsymbol{\eta}}.$$

Proof. Thanks to (2.5) and (2.6) it is not difficult to see that the sequence $\{w_{\boldsymbol{\theta}}^n\}$ is bounded in W, and it therefore admits a weakly convergent subsequence $\{\widetilde{w}_{\boldsymbol{\theta}}^n\}$. We let $u_{\boldsymbol{\theta}}^* \in W$ denote the weak limit of $\{\widetilde{w}_{\boldsymbol{\theta}}^n\}$. Let now $A^T : W \to W'$ be defined as $\langle A^T v, w\rangle = \mathcal{A}(w, v)$, with $\langle\cdot,\cdot\rangle$ denoting the duality pairing. We have, by the definition of weak limit,

$$\|u - u_{\boldsymbol{\theta}}^*\|_{op,\boldsymbol{\eta}} = \sup_{\substack{v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}} \\ v_{\boldsymbol{\eta}} \neq 0}} \frac{\langle A^T v_{\boldsymbol{\eta}}, u - u_{\boldsymbol{\theta}}^*\rangle}{\|v_{\boldsymbol{\eta}}\|_V} = \sup_{\substack{v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}} \\ v_{\boldsymbol{\eta}} \neq 0}} \frac{\lim_{n\to\infty} \langle A^T v_{\boldsymbol{\eta}}, u - \widetilde{w}_{\boldsymbol{\theta}}^n,\rangle}{\|v_{\boldsymbol{\eta}}\|_V}. \tag{2.8}$$

Now, for any $v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}$, $v_{\boldsymbol{\eta}} \neq 0$, we have

$$\frac{\lim_{n\to\infty} \langle A^T v_{\boldsymbol{\eta}}, u - \widetilde{w}_{\boldsymbol{\theta}}^n\rangle}{\|v_{\boldsymbol{\eta}}\|_V} \leqslant \lim_{n\to\infty} \sup_{\substack{v'_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}} \\ v'_{\boldsymbol{\eta}} \neq 0}} \frac{\mathcal{A}(u - \widetilde{w}_{\boldsymbol{\theta}}^n, v'_{\boldsymbol{\eta}})}{\|v'_{\boldsymbol{\eta}}\|_V} = \lim_{n\to\infty} \|u - \widetilde{w}_{\boldsymbol{\theta}}^n\|_{op,\boldsymbol{\eta}} = \sigma^*,$$

whence $\|u - u_{\boldsymbol{\theta}}^*\|_{op,\boldsymbol{\eta}} \leqslant \sigma^*$. ∎

We now prove Cea's lemma of best approximation for WAN on coercive problems.

LEMMA 2. *Let assumption* (2.6) *hold, and let u be the solutions to* (1.2) *and $u^*_{\boldsymbol\theta} \in \mathrm{cl}_w^{seq}(W_{\boldsymbol\theta})$ be the weak limit of a weakly convergent minimizing sequence $\{\widetilde{w}^n_{\boldsymbol\theta}\}$ for* (2.3)*. Then we have the following error bound:*

$$\|u - u^*_{\boldsymbol\theta}\|_W \leqslant \left(1 + \frac{2M}{\alpha}\right) \inf_{w_{\boldsymbol\theta} \in W_{\boldsymbol\theta}} \|u - w_{\boldsymbol\theta}\|_W. \tag{2.9}$$

Proof. We start by observing that (2.6) implies that, for any two elements $w_{1,\boldsymbol\theta}$ and $w_{2,\boldsymbol\theta}$ of $W_{\boldsymbol\theta}$ it holds that

$$\alpha \|w_{1,\boldsymbol\theta} - w_{2,\boldsymbol\theta}\|_W \leqslant \sup_{\substack{v_{\boldsymbol\eta} \in V_{\boldsymbol\eta} \\ v_{\boldsymbol\eta} \neq 0}} \frac{\mathcal{A}(w_{1,\boldsymbol\theta} - w_{2,\boldsymbol\theta}, v_{\boldsymbol\eta})}{\|v_{\boldsymbol\eta}\|_W} = \|w_{1,\boldsymbol\theta} - w_{2,\boldsymbol\theta}\|_{op,\boldsymbol\eta}. \tag{2.10}$$

Let now $e^* = u - u^*_{\boldsymbol\theta}$, and let $w_{\boldsymbol\theta}$ be an arbitrary element in $W_{\boldsymbol\theta}$. Letting $(\cdot,\cdot)_W$ denote the scalar product in W and $\mathfrak{R} : W \to W'$ denote the Riesz isomorphism, we have

$$\begin{aligned} \|u^*_{\boldsymbol\theta} - w_{\boldsymbol\theta}\|_W &= \frac{(u^*_{\boldsymbol\theta} - w_{\boldsymbol\theta}, u^*_{\boldsymbol\theta} - w_{\boldsymbol\theta})_W}{\|u^*_{\boldsymbol\theta} - w_{\boldsymbol\theta}\|_W} = \frac{\langle \mathfrak{R}(u^*_{\boldsymbol\theta} - w_{\boldsymbol\theta}), u^*_{\boldsymbol\theta} - w_{\boldsymbol\theta}\rangle}{\|u^*_{\boldsymbol\theta} - w_{\boldsymbol\theta}\|_W} \\ &= \lim_{n\to\infty} \frac{\langle \mathfrak{R}(u^*_{\boldsymbol\theta} - w_{\boldsymbol\theta}), \widetilde{w}^n_{\boldsymbol\theta} - w_{\boldsymbol\theta}\rangle}{\|u^*_{\boldsymbol\theta} - w_{\boldsymbol\theta}\|_W} \leqslant \lim_{n\to\infty} \|\widetilde{w}^n_{\boldsymbol\theta} - w_{\boldsymbol\theta}\|_W \leqslant \alpha^{-1} \lim_{n\to\infty} \|\widetilde{w}^n_{\boldsymbol\theta} - w_{\boldsymbol\theta}\|_{op,\boldsymbol\eta}. \end{aligned}$$

Note that we used (2.10) for the last bound. Adding and subtracting u in the right hand side and using (2.3) and (1.1), we have

$$\begin{aligned} \|u^*_{\boldsymbol\theta} - w_{\boldsymbol\theta}\|_W &\leqslant \alpha^{-1} \lim_{n\to\infty} \|u - \widetilde{w}^n_{\boldsymbol\theta}\|_{op,\boldsymbol\eta} + \alpha^{-1}\|u - w_{\boldsymbol\theta}\|_{op,\boldsymbol\eta} \\ &\leqslant \alpha^{-1} \inf_{w'_{\boldsymbol\theta} \in W_{\boldsymbol\theta}} \|u - w'_{\boldsymbol\theta}\|_{op,\boldsymbol\eta} + \alpha^{-1}\|u - w_{\boldsymbol\theta}\|_{op,\boldsymbol\eta} \leqslant \frac{2}{\alpha}\|u - w_{\boldsymbol\theta}\|_{op,\boldsymbol\eta}. \end{aligned} \tag{2.11}$$

Since $w_{\boldsymbol\theta} \in W_{\boldsymbol\theta}$ is arbitrary, using (2.2) and a triangle inequality we get (2.9). ∎

Generally, the weak solutions to (1.14), defined as the weak limits of minimizing sequences for the right hand side in $W_{\boldsymbol\theta}$, are not necessarily unique. Moreover, the solution of the minimization problem (2.3) itself might not lie in $W_{\boldsymbol\theta}$, but only in its weak sequential closure. In such a case, it can be proven (see [18]) that the sequence of parameters in $\mathcal{P}_{\boldsymbol\theta}$ resulting from the minimization procedure is unbounded, which results in numerical instability. A possible remedy (see [1]) is to restrict both maximization in $V_{\boldsymbol\eta}$ and minimization in $W_{\boldsymbol\theta}$ to subsets of $V_{\boldsymbol\eta}$ and $W_{\boldsymbol\theta}$ corresponding to parameters in $\mathcal{P}_{\boldsymbol\eta}$ and $\mathcal{P}_{\boldsymbol\theta}$ with euclidean norm bounded by a suitable constant B. In such a case, one can apply standard calculus results to prove the existence of a minimizer $w_{\boldsymbol\theta} \in W_{\boldsymbol\theta}$. However, finding an appropriate choice of B remains a challenging problem. A too-small value of B will result in poor approximation regardless of the network's approximation capability, and if B is very large, it ultimately serves no purpose. Lemma 2 does, instead, guarantee that even when multiple weak solutions exist, they all provide a quasi-best approximation of u in W. Moreover, we can obtain a quasi-best approximation to u within the approximation class $W_{\boldsymbol\theta}$ by taking entries of any minimizing sequence sufficiently close to convergence. Indeed, for any minimizing sequence $\{\widetilde{w}^n_{\boldsymbol\theta}\}$, given $\varepsilon > 0$ we can choose k such that

$$\|u - \widetilde{w}^k_{\boldsymbol\theta}\|_{op,\boldsymbol\eta} \leqslant \inf_{w_{\boldsymbol\theta} \in W_{\boldsymbol\theta}} \|u - w_{\boldsymbol\theta}\|_{op,\boldsymbol\eta} + \varepsilon.$$

Then, by (2.9) and (2.11) we have

$$\|u - \widetilde{w}^k_{\boldsymbol\theta}\|_W \leqslant \|u - u^*_{\boldsymbol\theta}\|_W + \|u^*_{\boldsymbol\theta} - \widetilde{w}^k_{\boldsymbol\theta}\|_W \lesssim \inf_{w_{\boldsymbol\theta} \in W_{\boldsymbol\theta}} \|u - w_{\boldsymbol\theta}\|_W + \varepsilon,$$

meaning that any minimizing sequence does approximate the solution u in the norm $\|\cdot\|_W$ within the accuracy allowed by the chosen neural network class architecture in a finite number of steps. It is important to observe that, under proper assumptions, the cost functional is equivalent to the W' norm of the residual, thus providing a reliable a posteriori error bound. Moreover, by (2.11), the cost functional evaluated on $w_{\boldsymbol{\theta}}$ provides an upper bound for the discrepancy, in W, between $w_{\boldsymbol{\theta}}$ and the weak limit w_θ^*, and can then be leveraged to devise a stopping criterion.

Remark 2.1. Since $W_{\boldsymbol{\theta}}$ is a function class and not a function space, (2.6) implies that $V_{\boldsymbol{\eta}}$ should be a richer function class than $W_{\boldsymbol{\theta}}$. When $\mathcal{A}$ is coercive and symmetric, the Deep Ritz method can be interpreted as choosing, in our abstract formulation, $V_{\boldsymbol{\eta}} = u - W_{\boldsymbol{\theta}}$. It is not difficult to check that with such a definition of $V_{\boldsymbol{\eta}}$, if $\mathcal{A}$ is coercive, both Lemma 1 and Lemma 2 still hold. However, in practice, numerical evidence suggests that using a separate and more comprehensive space for $V_{\boldsymbol{\eta}}$ than $u - W_{\boldsymbol{\theta}}$ enhances both numerical efficiency (faster convergence) and accuracy.

Remark 2.2. To fully exploit (2.9), we combine it with approximation results on neural network classes. We refer to [10] for a survey of the different results available in the literature and to the references therein. In particular, we recall that when $W = H^1(\Omega)$ and $W_{\boldsymbol{\theta}}$ is a function class of DNN network with ReLU activation function, it was shown in [9] that for any function $\varphi \in H^m(\Omega), m > 1$ and Ω is Lipschitz,

$$\min_{\varphi_{\boldsymbol{\theta}} \in V_{\boldsymbol{\theta}}} \|\varphi - \varphi_{\boldsymbol{\theta}}\|_{H^1(\Omega)} \leqslant C(m,d) N_{\boldsymbol{\theta}}^{-(m-1)/d} \|\varphi\|_{H^m(\Omega)}, \tag{2.12}$$

where $C(m,d) \geqslant 0$ is a function depends on (m,d) and $N_{\boldsymbol{\theta}}$ is the number of neurons in the DNN network. Combining such a bound with the quasi-best approximation estimates allows us to deduce a priori error estimates of the WAN schemes.

Remark 2.3. While we focused our analysis on linear problems, the WAN method can be, and is, applied also in the non linear framework. Indeed, under suitable assumptions on the operator A (for instance, if A is monotone and Lipschitz continuous) the existence of weakly converging minimizing sequences whose weak limit satisfies the estimate of Lemma 2 carries over to the nonlinear case. A proof in the case of monotone operators is given in the online supplementary material. Beyond that, also in cases where monotonicity does not hold, numerical results will show the effectiveness of our approach (see subsection 6.1 and subsection 6.2)

2.2. PDE without coercivity. We now drop the assumption that $V = W$, and we assume instead that there exists an operator $\mathfrak{R} : V \to W'$ such that

$$\inf_{w \in W} \sup_{\substack{v \in V \\ v \neq 0}} \frac{\langle \mathfrak{R}v, w\rangle}{\|w\|_W \|v\|_V} \geqslant \alpha^* > 0, \qquad \|\mathfrak{R}v\|_{W'} \leqslant M^* \|v\|_V. \tag{2.13}$$

Remark that, as we assume that problem (1.2) is well posed, a possible choice for $\mathfrak{R}$ is $\mathfrak{R} = A^T$, but choices with better stability constants α^* might exist. Moreover assume that $V_{\boldsymbol{\eta}} \subset V$ can be chosen so that we have the discrete inf-sup condition:

$$\kappa \|w_{\boldsymbol{\theta}}\|_W \leqslant \sup_{\substack{v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}} \\ v_{\boldsymbol{\eta}} \neq 0}} \frac{\mathcal{A}(w_{\boldsymbol{\theta}}, v_{\boldsymbol{\eta}})}{\|v_{\boldsymbol{\eta}}\|_V} \quad \forall w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}} \cup S_{\boldsymbol{\theta}}, \tag{2.14}$$

with $S_{\boldsymbol{\theta}}$ defined in (2.4). It is easy to see that Lemma 1 holds with proof unchanged also in this case, which gives us the existence of a (possibly not unique) element $u_{\boldsymbol{\theta}}^* \in \mathrm{cl}_w^{seq}(W_{\boldsymbol{\theta}})$, weak limit of a minimizing sequence $\{\widetilde{w}_{\boldsymbol{\theta}}^n\}$ of elements of $W_{\boldsymbol{\theta}}$, satisfying

$$\|u - u_{\boldsymbol{\theta}}^*\|_{op,\boldsymbol{\eta}} \leqslant \|u - w_{\boldsymbol{\theta}}\|_{op,\boldsymbol{\eta}} \qquad \forall w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}.$$

LEMMA 3. *Let $V_{\boldsymbol{\eta}}$ be chosen in such a way that assumption* (2.14) *is satisfied for some constant $\kappa > 0$, possibly depending on $V_{\boldsymbol{\eta}}$. Let u be the solutions to* (1.2) *and let $u^*_{\boldsymbol{\theta}} \in \mathrm{cl}^{seq}_w(W_{\boldsymbol{\theta}})$ be the weak limit of a weakly convergent minimizing sequence $\{\widetilde{w}^n_{\boldsymbol{\theta}}\}$ for* (2.3)*. Then we have the following error bound:*

$$\|u - u^*_{\boldsymbol{\theta}}\|_W \leqslant \left(1 + 2\frac{M^*}{\alpha_*}\frac{M}{\kappa}\right) \inf_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \|u - w_{\boldsymbol{\theta}}\|_W. \tag{2.15}$$

Proof. Let $w_{\boldsymbol{\theta}}$ be an arbitrary element of $W_{\boldsymbol{\theta}}$. Thanks to (2.13) and (2.14) we can write

$$\begin{aligned} \alpha_*\|u^*_{\boldsymbol{\theta}} - w_{\boldsymbol{\theta}}\|_W &\leqslant \sup_{\substack{v \in V \\ v \neq 0}} \frac{\langle \mathfrak{R} v, u^*_{\boldsymbol{\theta}} - w_{\boldsymbol{\theta}}\rangle}{\|v\|_V} = \lim_{n \to \infty} \sup_{\substack{v \in V \\ v \neq 0}} \frac{\langle \mathfrak{R} v, \widetilde{w}^n_{\boldsymbol{\theta}} - w_{\boldsymbol{\theta}}\rangle}{\|v\|_V} \\ &\leqslant M^* \lim_{n\to\infty} \|\widetilde{w}^n_{\boldsymbol{\theta}} - w_{\boldsymbol{\theta}}\|_W \leqslant \frac{M^*}{\kappa} \lim_{n\to\infty} \|\widetilde{w}^n_{\boldsymbol{\theta}} - w_{\boldsymbol{\theta}}\|_{op,\boldsymbol{\eta}}. \end{aligned} \tag{2.16}$$

By the same argument used for the proof of Lemma 2, we then obtain that

$$\|u^*_{\boldsymbol{\theta}} - w_{\boldsymbol{\theta}}\|_W \leqslant 2\frac{M^*}{\alpha_*}\frac{1}{\kappa}\|u - w_{\boldsymbol{\theta}}\|_{op,\boldsymbol{\eta}}, \tag{2.17}$$

and, consequently, by the triangle inequality,

$$\|u - u^*_{\boldsymbol{\theta}}\|_W \leqslant \left(1 + 2\frac{M^*}{\alpha_*}\frac{M}{\kappa}\right)\|u - w_{\boldsymbol{\theta}}\|_W, \tag{2.18}$$

which, thanks to the arbitrariness of $w_{\boldsymbol{\theta}}$, gives (2.15). ∎

Like the coercive case, we can have an almost best approximation in a finite number of steps of any weakly converging minimizing sequence $\{\widetilde{w}^n_{\boldsymbol{\theta}}\}$. More precisely, by the same argument as for the coercive case, for all $\varepsilon > 0$ there exists a k such that

$$\|u - \widetilde{w}^k_{\boldsymbol{\theta}}\|_W \lesssim \inf_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \|u - w_{\boldsymbol{\theta}}\|_W + \varepsilon.$$

We conclude this section by the following observation: let $\mathcal{J}(\cdot)$ denote any functional on W equivalent to the $\|\cdot\|_{op,\boldsymbol{\eta}}$ norm:

$$c_*\|w\|_{op,\boldsymbol{\eta}} \leqslant \mathcal{J}(w) \leqslant C^*\|w\|_{op,\boldsymbol{\eta}}, \qquad \forall w \in W, \tag{2.19}$$

and consider the problem

$$u^*_{\boldsymbol{\theta}} = \operatorname*{argmin}_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \mathcal{J}(u - w_{\boldsymbol{\theta}}). \tag{2.20}$$

Then there exists a possibly not unique $w^\flat_\theta \in \mathrm{cl}^{seq}_w(W_{\boldsymbol{\theta}})$ such that

$$\mathcal{J}(u - u^\flat_{\boldsymbol{\theta}}) \leqslant \inf_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \mathcal{J}(u - w_{\boldsymbol{\theta}}).$$

Moreover for all $u^\flat_{\boldsymbol{\theta}}$ such that $u^\flat_{\boldsymbol{\theta}}$ is the weak limit of a minimizing sequence $\{\widetilde{w}^n_{\boldsymbol{\theta}}\}$ for (2.20), it holds that

$$\|u - u^\flat_{\boldsymbol{\theta}}\|_W \leqslant \left(1 + 2\frac{C^*}{c_*}\frac{M^*}{\alpha_*}\frac{M}{\kappa}\right) \inf_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \|u - w_{\boldsymbol{\theta}}\|_W.$$

Indeed, by (2.19), all minimizing sequences are bounded with respect to the $\|\cdot\|_{op,\boldsymbol{\eta}}$ norm, and, hence, with respect to the $\|\cdot\|_W$ norm. Any minimizing sequence does then weakly converge to an element $u^\flat_{\boldsymbol{\theta}}$. Moreover, initially proceeding as in (2.16), thanks to (2.19), we have, for $w_{\boldsymbol{\theta}}$ arbitrary,

$$\begin{aligned}\alpha_*\|u^*_{\boldsymbol{\theta}} - w_{\boldsymbol{\theta}}\|_W &\leqslant \frac{M^*}{\kappa}\lim_{n\to\infty}\|\widetilde{w}^n_{\boldsymbol{\theta}} - w_{\boldsymbol{\theta}}\|_{op,\boldsymbol{\eta}} \leqslant \frac{M^*}{\kappa}\left(\lim_{n\to\infty}\|\widetilde{w}^n_{\boldsymbol{\theta}} - u\|_{op,\boldsymbol{\eta}} + \|u - w_{\boldsymbol{\theta}}\|_{op,\boldsymbol{\eta}}\right)\\ &\leqslant \frac{M^*}{\kappa c_*}\left(\lim_{n\to\infty}\mathcal{J}(\widetilde{w}^n_{\boldsymbol{\theta}} - u) + \mathcal{J}(u - w_{\boldsymbol{\theta}})\right) \leqslant \frac{2MM^*C^*}{\kappa c_*}\|u - w_{\boldsymbol{\theta}}\|_W.\end{aligned}$$

3. Two novel stabilized loss functions.

To mitigate undesirable oscillations during the optimization procedure, resulting from the inexact solution of the maximization problem involved in the definition of the operator norm, we propose two alternative definitions of the cost functional that yields the same minimum, while avoiding direct normalization. More precisely, we introduce two new functionals on the product space $W \times V$, such that the supremum over $v \in V$, for $w \in W$ fixed, also gives, up to a possible rescaling and translation, the operator norm of w, while yielding a more favorable optimization problem under discretization.

3.1. Stabilized WAN method.

Define

$$\|\!|w|\!\|^2_{op} = \sup_{v\in V}\left(\mathcal{A}(w,v) - \frac{\gamma_d}{2}\|v\|^2_V\right), \quad \|\!|w|\!\|^2_{op,\boldsymbol{\eta}} = \sup_{v_{\boldsymbol{\eta}}\in V_{\boldsymbol{\eta}}}\left(\mathcal{A}(w,v_{\boldsymbol{\eta}}) - \frac{\gamma_d}{2}\|v_{\boldsymbol{\eta}}\|^2_V\right), \tag{3.1}$$

where $\gamma_d > 0$ is a constant, and consider the following problem:

$$u^\sharp_{\boldsymbol{\theta}} = \underset{w_{\boldsymbol{\theta}}\in W_{\boldsymbol{\theta}}}{\operatorname{argmin}}\|\!|u - w_{\boldsymbol{\theta}}|\!\|_{op,\boldsymbol{\eta}}. \tag{3.2}$$

The following lemma shows that the norms defined in (3.1) coincide with the operator norms defined in (1.4) and (2.1), up to a constant dependent on γ_d.

Lemma 4. *Assume that $v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}$ implies $\lambda v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}$ for all $\lambda \in \mathbb{R}^+$. Then, for any $w \in W$, there holds*

$$\frac{1}{2\gamma_d}\|w\|^2_{op} = \|\!|w|\!\|^2_{op}, \qquad \frac{1}{2\gamma_d}\|w\|^2_{op,\boldsymbol{\eta}} = \|\!|w|\!\|^2_{op,\boldsymbol{\eta}}. \tag{3.3}$$

Remark 3.1. From this lemma, it seems reasonable to choose, e.g., $\gamma_d = 1$. However, experiments have shown that adjusting the value of γ can effectively control the oscillations in experiments.

Proof. We prove the second of the two equalities. The first can be proven by the same argument. For any fixed $w \in W_{\boldsymbol{\theta}}$ with $w \neq 0$, and for all $\varepsilon > 0$, there exists a $\overline{\varphi}^\varepsilon_w \in V_{\boldsymbol{\eta}}$ (depending on ε) with $\|\overline{\varphi}^\varepsilon_w\|_V = 1$, such that

$$\mathcal{A}(w, \overline{\varphi}^\varepsilon_w) \geqslant (1-\varepsilon)\|w\|_{op,\boldsymbol{\eta}},$$

which, setting $\varphi_w = \gamma_d^{-1}\|w\|_{op,\boldsymbol{\eta}}\overline{\varphi}^\varepsilon_w$, yields

$$\begin{aligned}&\sup_{\varphi_{\boldsymbol{\eta}}\in V_{\boldsymbol{\eta}}}\left(\mathcal{A}(w,\varphi_{\boldsymbol{\eta}}) - \frac{\gamma_d}{2}\|\varphi_{\boldsymbol{\eta}}\|^2_V\right) \geqslant \mathcal{A}(w,\varphi_w) - \frac{\gamma_d}{2}\|\varphi_w\|^2_V\\ &=\gamma_d^{-1}\|w\|_{op,\boldsymbol{\eta}}\mathcal{A}(w,\overline{\varphi}^\varepsilon_w) - \frac{1}{2\gamma_d}\|w\|^2_{op,\boldsymbol{\eta}} \geqslant \left(\frac{1}{2} - \varepsilon\right)\frac{1}{\gamma_d}\|w\|^2_{op,\boldsymbol{\eta}}.\end{aligned} \tag{3.4}$$

By the arbitrariness of ε we then obtain that $|||w|||^2_{op,\boldsymbol{\eta}} \geqslant \|w\|^2_{op,\boldsymbol{\eta}}/(2\gamma_d)$. To prove the converse inequality, using Young's equality gives

$$\sup_{\varphi_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}} \Big(\mathcal{A}(w, \varphi_{\boldsymbol{\eta}}) - \frac{\gamma_d}{2} \|\varphi_{\boldsymbol{\eta}}\|^2_V \Big) \leqslant \sup_{\varphi_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}} \Big(\|w\|_{op,\boldsymbol{\eta}} \|\varphi_{\boldsymbol{\eta}}\|_V - \frac{\gamma_d}{2} \|\varphi_{\boldsymbol{\eta}}\|^2_V \Big) \leqslant \frac{1}{2\gamma_d} \|w\|^2_{op,\boldsymbol{\eta}}.$$

The first part of (3.3) can be proved in a similar way. ∎

The analysis of the minimization problem (2.3) carries then over to the minimization problem (3.2). Then, there exists at least a minimizing sequence in $W_{\boldsymbol{\theta}}$ weakly converging to a limit $u^\sharp_{\boldsymbol{\theta}} \in \mathrm{cl}^{seq}_w(W_{\boldsymbol{\theta}})$, and all weak limits of minimizing sequences satisfy either bound (2.9) or bound (2.15), depending on whether $\mathcal{A}$ is coercive or not.

Remark 3.2. When w approaches the true solution, we have that

$$v^*(w) = \operatorname*{argmax}_{v \in V} \big(\mathcal{A}(u - w, v) - \frac{\gamma_d}{2} \|v\|^2_V \big) \to 0.$$

As a consequence, depending on how large the space $W_{\boldsymbol{\theta}}$ is, the problem

$$u^*_\theta = \operatorname*{argmin}_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \big(\mathcal{A}(u - w_{\boldsymbol{\theta}}, v^*(w_{\boldsymbol{\theta}})) - \frac{\gamma_d}{2} \|v^*(w_{\boldsymbol{\theta}})\|^2_V \big)$$

might be close to the problem $u^*_\theta = \operatorname*{argmin}_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \mathcal{A}(u - w_{\boldsymbol{\theta}}, v^*(w_{\boldsymbol{\theta}}))$, which, in turn, if $\|u - w_{\boldsymbol{\theta}}\|_W$ is small, might be ill-posed and too sensitive to the errors in evaluating $v^*(w_{\boldsymbol{\theta}})$. The following subsection introduces a further stabilized loss function that mitigates this issue.

3.2. A further stabilized WAN method. We now define the following alternative operator norm $|||\cdot|||^+$ as follows:

$$(3.5)\quad \begin{aligned} \big(|||w|||^+_{op}\big)^2 &:= \sup_{v \in V} \Big(\mathcal{A}(w, v) - \frac{\gamma_d}{2} \|v\|^2_V + \|v\|_V \Big), \\ \big(|||w|||^+_{op,\boldsymbol{\eta}}\big)^2 &:= \sup_{v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}} \Big(\mathcal{A}(w, v_{\boldsymbol{\eta}}) - \frac{\gamma_d}{2} \|v_{\boldsymbol{\eta}}\|^2_V + \|v_{\boldsymbol{\eta}}\|_V \Big), \end{aligned}$$

where $\gamma_d > 0$ is a constant, and we define the following minimization problem:

$$(3.6)\quad u^\ddagger_{\boldsymbol{\theta}} = \operatorname*{argmin}_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} |||u - w_{\boldsymbol{\theta}}|||^+_{op,\boldsymbol{\eta}}.$$

The following lemma states the relation between the norms defined in (3.5) and the operator norms.

LEMMA 5. *Assume that $v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}$ implies $\lambda v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}$ for all $\lambda \in \mathbb{R}^+$. For any $w \in W$, there holds*

$$(3.7)\quad \frac{1}{\sqrt{2\gamma_d}} (\|w\|_{op} + 1) = |||w|||^+_{op}, \qquad \frac{1}{\sqrt{2\gamma_d}} (\|w\|_{op,\boldsymbol{\eta}} + 1) = |||w|||^+_{op,\boldsymbol{\eta}}.$$

Proof. We prove the second of the two equalities, the first can be proven by the same argument. For any fixed $w \in W$ with $w \neq 0$, and for all $\varepsilon > 0$, there exists $\overline{\varphi}^\varepsilon_w \in V_{\boldsymbol{\eta}}$ with $\|\overline{\varphi}^\varepsilon_w\|_V = 1$ that satisfies

$$(3.8)\quad \mathcal{A}(w, \overline{\varphi}^\varepsilon_w) \geqslant (1 - \varepsilon) \|w\|_{op,\boldsymbol{\eta}},$$

which, setting $\varphi_w = \gamma_d^{-1} s\|w\|_{op,\boldsymbol{\eta}}\overline{\varphi}_w^{\varepsilon}$, for some $s > 0$ to be chosen, yields

$$\begin{aligned}\sup_{\varphi_{\boldsymbol{\eta}}\in V_{\boldsymbol{\eta}}} \Big(\mathcal{A}(w,\varphi_{\boldsymbol{\eta}}) - \frac{\gamma_d}{2}\|\varphi_{\boldsymbol{\eta}}\|_V^2 + \|\varphi_{\boldsymbol{\eta}}\|_V\Big) &\geqslant \mathcal{A}(w,\varphi_w) - \frac{\gamma_d}{2}\|\varphi_w\|_V^2 + \|\varphi_w\|_V \\ &=\gamma_d^{-1}s\|w\|_{op,\boldsymbol{\eta}}\mathcal{A}(w,\overline{\varphi}_w^{\varepsilon}) - \frac{s^2}{2\gamma_d}\|w\|_{op,\boldsymbol{\eta}}^2 + \gamma_d^{-1}s\|w\|_{op,\boldsymbol{\eta}} \\ &\geqslant(1-\varepsilon)\frac{s}{\gamma_d}\|w\|_{op,\boldsymbol{\eta}}^2 - \frac{s^2}{2\gamma_d}\|w\|_{op,\boldsymbol{\eta}}^2 + \frac{s}{\gamma_d}\|w\|_{op,\boldsymbol{\eta}}.\end{aligned}$$

We can choose s that maximizes the term on the right hand side. By direct computations, we have

$$\max_{s\in\mathbb{R}^+}\left((1-\varepsilon)\frac{s}{\gamma_d}\|w\|_{op,\boldsymbol{\eta}}^2 - \frac{s^2}{2\gamma_d}\|w\|_{op,\boldsymbol{\eta}}^2 + \frac{s}{\gamma_d}\|w\|_{op,\boldsymbol{\eta}}\right) = \frac{1}{2\gamma_d}((1-\varepsilon)\|w\|_{op,\boldsymbol{\eta}}+1)^2.$$

Above we have used the fact that $\max_{s\in\mathbb{R}^+}(-as^2+bs) = b^2/4a$. By the arbitrariness of ε we then obtain $(|\!|\!|w|\!|\!|_{op,\boldsymbol{\eta}}^+)^2 \geqslant (\|w\|_{op,\boldsymbol{\eta}}+1)^2/(2\gamma_d)$. The converse inequality can be proved by once again using Young's inequality. ∎

Again, the analysis of the minimization problem (2.3), including the existence of a weakly converging minimization sequence and a best approximation bound for all weak limits of minimizing sequences, carries over to the minimization problem (3.2).

4. Imposition of Dirichlet boundary conditions. Herein we will adapt to the case of WANs the technique introduced in [7] for dealing with Dirichlet boundary conditions. The idea is to weigh the elements of the test adversarial network by multiplying a cutoff function ϕ (see [21] and references therein), so that the resulting test functions are forced to be zero on the boundary. The Dirichlet boundary condition can then be imposed on the primal network using a penalty without violating the consistency of the equation. For simplicity, we assume that the problem is a symmetric second-order static elliptic PDE. We also assume that the boundary $\partial\Omega$ is smooth (C^3 to be precise).

We first present the ideas in the simple framework of the Deep Ritz method for the case of homogeneous boundary conditions. We assume the operator $\mathcal{A}$ of (1.2) to be symmetric under homogeneous Dirichlet boundary conditions. Then the continuous Ritz method may be written as

$$u = \operatorname*{argmin}_{v\in H_0^1(\Omega)} \left(0.5\mathcal{A}(v,v) - (f,v)_\Omega\right).$$

Under the smoothness assumptions on $\partial\Omega$ we know that, provided f is sufficiently smooth, $u \in H^m(\Omega)$ for some $m \geqslant 3$ and $\|u\|_{H^3(\Omega)} \lesssim \|f\|_{H^1(\Omega)}$. Assuming that $w_{\boldsymbol{\theta}} \in H_0^1(\Omega)$ for all $w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}$, the Deep Ritz method takes the form

$$u_{\boldsymbol{\theta}}^* = \operatorname*{argmin}_{w_{\boldsymbol{\theta}}\in W_{\boldsymbol{\theta}}} \left(0.5\mathcal{A}(w_{\boldsymbol{\theta}},w_{\boldsymbol{\theta}}) - f(w_{\boldsymbol{\theta}})\right).$$

As we already mentioned, the problem with this formulation is that it appears to be very difficult to design networks that satisfy boundary conditions by construction. Instead, typically, a penalty term of the form $\lambda\|Tw_{\boldsymbol{\theta}}\|_{\partial\Omega}$ is added to the functional on the right hand side [6]. The convergence to the solution $u \in H_0^1(\Omega)$ of the continuous problem is obtained by letting $\lambda \to \infty$ and enriching the network space. In the classical numerical methods, e.g., the Finite Element Method, λ is proportional to h^{-s}, with

h being the mesh size and $s > 0$ a carefully chosen exponent. With neural network methods, it is, however, not obvious how to match the dimension of the space to the rate by which λ grows. In general, either the accuracy or the conditioning of the nonlinear system suffers.

Our idea is to build the boundary conditions into the formulation by weighting the network functions with the level set function ϕ, where $\phi|_{\partial\Omega} = 0$, $\phi|_{\Omega} > 0$, and ϕ behaves as a distance function in the vicinity of $\partial\Omega$. The solution we look for then takes the form $\phi w_{\boldsymbol{\theta}}$ with $w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}$. The Cut Deep Ritz method reads

$$\nu^*_{\boldsymbol{\theta}} = \operatorname*{argmin}_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \left(0.5\mathcal{A}(\phi w_{\boldsymbol{\theta}}, \phi w_{\boldsymbol{\theta}}) - f(\phi w_{\boldsymbol{\theta}})\right).$$

It is straightforward to show that this is equivalent to

$$\nu^*_{\boldsymbol{\theta}} = \operatorname*{argmin}_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \left(\mathcal{A}(u - \phi w_{\boldsymbol{\theta}}, u - \phi w_{\boldsymbol{\theta}})\right) = \operatorname*{argmin}_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \left(\sup_{v_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \frac{\mathcal{A}(u - \phi w_{\boldsymbol{\theta}}, u - \phi v_{\boldsymbol{\theta}})}{\sqrt{\mathcal{A}(u - \phi v_{\boldsymbol{\theta}}, u - \phi v_{\boldsymbol{\theta}})}} \right). \tag{4.1}$$

Following Remark 2.1, and assuming, for the sake of simplicity, that the minimization problem (4.1) has a unique solution $\nu^*_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}$ we then have that

$$\|u - \phi\nu^*_{\boldsymbol{\theta}}\|_{H^1(\Omega)} \leqslant C \inf_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \|u - \phi w_{\boldsymbol{\theta}}\|_{H^1(\Omega)}.$$

It remains to show that $\phi w_{\boldsymbol{\theta}}$ is capable of approximating u in $H^1_0(\Omega)$. To this end let $\mathcal{O}$ be some domain such that $\Omega \subset \mathcal{O}$, where $\mathcal{O}$ is a box in $\mathbb{R}^d$ and let $\tilde{u}$ denote a stable extension of u to $\mathcal{O}$ [20]. We assume the following on the boundary $\partial\Omega$ and ϕ.

ASSUMPTION 4.1. *Let Ω be a bounded domain in $\mathbb{R}^d$. The boundary $\partial\Omega$ can be covered by open sets $\mathcal{O}_i, i = 1, \cdots, I$, and one can introduce on every $\mathcal{O}_i$ local coordinates $\xi_1, \cdots, \xi_d$ with $\xi_d = \phi$ such that all the partial derivatives $\partial\xi_i^\alpha/\partial^\alpha x$ and $\partial x^\alpha/\partial^\alpha \xi$ up to order $k+1$ are bounded by some $C_0 > 0$. Moreover, ϕ is of class C^{k+1} on $\mathcal{O}$, where $k+1 \geqslant 3$ is the smoothness of the domain, and $|\phi| \geqslant M_0$ on $\mathcal{O} \setminus \cup \mathcal{O}_i$ with some $m > 0$, and in $\cup\mathcal{O}_i$, ϕ is a signed distance function to $\partial\Omega$.*

We further need the following Hardy-type inequality (see [7, Lemma 3.1]).

LEMMA 6. *We assume that the domain Ω is defined by the zero level set of the smooth function ϕ and that Assumption* 4.1 *is satisfied. Then for any $v \in H^{k+1}(\mathcal{O})$ such that $v|_{\partial\Omega} = 0$, there holds*

$$\|v/\phi\|_{H^k(\Omega)} \leqslant C\|v\|_{H^{k+1}(\mathcal{O})}.$$

Then, as by assumption $\|\phi\|_{W^{1,\infty}(\Omega)} < C$, combining the quasi best approximation bound given by Lemma 2.9 with (2.12) we obtain the following estimate for the Cut Deep Ritz method, with ReLU activation function:

$$\|u - \phi\nu^*_{\boldsymbol{\theta}}\|_{H^1(\Omega)} \lesssim N_{\boldsymbol{\theta}}^{-(m-2)/d}|u|_{H^m}.$$

In particular, when $m = 3$, using elliptic regularity, we can bound the $H^1(\Omega)$ norm of the error with $N_{\boldsymbol{\theta}}^{-1/d}\|f\|_{H^1(\Omega)}$. This shows that the method typically requires one order more regularity of the data than typically expected.

We now introduce the Cut weak adversarial network (CutWAN) method for problems not necessarily coercive and with non-homogeneous Dirichlet boundary conditions. We let

$$|\!|\!| w |\!|\!|_{op,\phi,\boldsymbol{\eta}} := \sup_{\varphi_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}} \left(\mathcal{A}(w, \phi\varphi_{\boldsymbol{\eta}}) - \frac{\gamma_d}{2}\|\phi\varphi_{\boldsymbol{\eta}}\|^2_{H^1(\Omega)} \right) \tag{4.2}$$

and set

$$u_{\boldsymbol{\theta}}^{\diamond} = \operatorname*{argmin}_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \left(\| u - w_{\boldsymbol{\theta}} \|_{op,\phi,\boldsymbol{\eta}} + \| w_{\boldsymbol{\theta}} - g \|_{H^{1/2}(\partial\Omega)} \right). \tag{4.3}$$

Note that the cut-off function ϕ only multiplies the test functions in the CutWAN network method. The boundary condition is weakly imposed by adding a penalty term on the primal network $w_{\boldsymbol{\theta}}|_{\partial\Omega}$. It is not difficult to check that this problem falls in the abstract formulation considered at the end of subsection 2.2. Here, the role of adversarial test network is played by the product space $\phi V_{\boldsymbol{\theta}} \times H^{-1/2}(\partial\Omega)$, and the inf-sup condition (2.14) becomes

$$\| w \|_W \lesssim \| w \|_{op,\phi,\boldsymbol{\eta}} + \| w \|_{H^{1/2}(\partial\Omega)}, \qquad \forall w \in S_{\boldsymbol{\theta}}. \tag{4.4}$$

In particular, under such an assumption, we have the following best approximation results for the CutWAN method.

LEMMA 7. *Assume that the inf-sup condition* (4.4) *holds. Let* $u_{\boldsymbol{\theta}}^{\diamond}$ *be the weak limit of a minimizing sequence for problem* (4.3)*. Then there holds*

$$\| u - u_{\boldsymbol{\theta}}^{\diamond} \|_W \lesssim \inf_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \| u - w_{\boldsymbol{\theta}} \|_W. \tag{4.5}$$

Note that the CutWAN method achieves optimal convergence rates even though the test function class is multiplied by ϕ. So the difficulty handled by the Hardy inequality in the Cut Deep Ritz method does not appear. Indeed the difficulty of controlling the levelset weighted test function is hidden in the inf-sup assumption (4.4). A study of this condition will be the topic of future work.

Similarly, we can also define the following algorithm. Define

$$\| w \|_{op,\phi,\boldsymbol{\eta}}^{+} := \sup_{\varphi_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}} \left(\mathcal{A}(w, \phi\varphi_{\boldsymbol{\eta}}) - \frac{\gamma_d}{2} \| \phi\varphi_{\boldsymbol{\eta}} \|_V^2 + \| \phi\varphi_{\boldsymbol{\eta}} \|_V \right), \tag{4.6}$$

and let

$$u_{\boldsymbol{\theta}}^{\eth} = \operatorname*{argmin}_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \left(\| u - w_{\boldsymbol{\theta}} \|_{op,\phi,\boldsymbol{\eta}}^{+} + \| w_{\boldsymbol{\theta}} - g \|_{H^{1/2}(\partial\Omega)} \right). \tag{4.7}$$

We refer to the above method as the shifted CutWAN method. One can also prove the best approximation results for the shifted CutWAN method similarly as in Lemma 7.

Remark 4.2. For computational convenience, the $H^{1/2}(\partial\Omega)$ norm in (4.3) and (4.7) can be replaced by a suitable combination of the L^2 norm of the function and of its tangential derivative. Indeed, using the Gagliardo-Nirenberg inequality we have that

$$\| w_{\boldsymbol{\theta}} - g \|_{H^{1/2}(\partial\Omega)} \leqslant \| w_{\boldsymbol{\theta}} - g \|_{L^2(\partial\Omega)}^{1/2} \| w_{\boldsymbol{\theta}} - g \|_{H^1(\partial\Omega)}^{1/2}. \tag{4.8}$$

It is not difficult to ascertain that if we replace the $H^{1/2}$ norm in (4.3) and (4.7) with the right hand side of (4.8), the analysis of Section 2 holds with minor changes, resulting in an error bound of the form

$$\| u - u_{\boldsymbol{\theta}}^{\diamond} \| \lesssim \inf_{w_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}} \left(\| u - w_{\boldsymbol{\theta}} \|_W + \| u - w_{\boldsymbol{\theta}} \|_{L^2(\partial\Omega)}^{1/2} \| u - w_{\boldsymbol{\theta}} \|_{H^1(\partial\Omega)}^{1/2} \right).$$

We observe that an analogous result would hold for the plain $L^2(\partial\Omega)$ penalization as originally proposed by [22, 1], if an inverse estimate were to hold for $W_{\boldsymbol{\theta}}$, allowing to bound the $H^1(\partial\Omega)$ norm of the boundary residual with its $L^2(\partial\Omega)$ norm times a constant depending on $W_{\boldsymbol{\theta}}$. Unfortunately this is generally not true, and Lemma 2 does not hold when using such a stabilization, which is, however, computationally convenient, and which we will test extensively in the forthcoming sections. We point out that, thanks to the combination of (4.8) with a Cauchy-Schwartz inequality, simply adding $\|g - w_{\boldsymbol{\theta}}\|_{H^1(\partial\Omega)}$ to the $L^2(\partial\Omega)$ penalized functional yields an a posteriori error estimator. This can be evaluated upon convergence of the optimization procedure, to check if the solution obtained with the cheaper L^2 penalized functional, is satisfactory. If not, it can serve, in a two stage strategy, as starting point for an additional optimization procedure relying on the more expensive functionals for which our theoretical error analysis applies.

5. Neural Network Structures.

5.1. Deep Neural Network (DNN) Structure. A DNN structure is the composition of multiple linear functions and nonlinear activation functions. We will use the DNN structure for $V_{\boldsymbol{\eta}}$. Specifically, the first component of DNN is a linear transformation $\boldsymbol{T}^l : \mathbb{R}^{n_l} \to \mathbb{R}^{n_{l+1}}$, $l = 1, \cdots, L$, defined as follows,

$$\boldsymbol{T}^l(\boldsymbol{x}^l) = \boldsymbol{W}^l \boldsymbol{x}^l + \boldsymbol{b}^l, \text{ for } \boldsymbol{x}^l \in \mathbb{R}^{n_l},$$

where $\boldsymbol{W}^l = (w^l_{i,j}) \in \mathbb{R}^{n_{l+1}\times n_l}$ and $\boldsymbol{b}^l \in \mathbb{R}^{n_{l+1}}$ are parameters in the DNN. The second component is an activation function $\psi : \mathbb{R} \to \mathbb{R}$ to be chosen, and typical examples of the activation functions are tanh, Sigmoid, and ReLU. Application of ψ to a vector $\boldsymbol{x} \in \mathbb{R}^n$ is defined component-wisely, i.e., $\psi(\boldsymbol{x}) = (\psi(x_i))$, $i = 1, 2, \cdots, n$. The l-th layer of the DNN is defined as the composition of the linear transform $\boldsymbol{T}^l$ and the nonlinear activation function ψ, i.e.,

$$\mathcal{N}^l(\boldsymbol{x}^l) := \psi(\boldsymbol{T}^l(\boldsymbol{x}^l)), \quad l = 1, \cdots, L-1.$$

For an input $\boldsymbol{x} \in \mathbb{R}^{n_1}$, a general L-layer DNN is defined as follows,

$$\mathcal{N}\mathcal{N}(\boldsymbol{x}; \boldsymbol{\theta}) := \boldsymbol{T}^L \circ \mathcal{N}^{L-1} \circ \cdots \circ \mathcal{N}^2 \circ \mathcal{N}^1(\boldsymbol{x}), \tag{5.1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^N$ stands for all the parameters in the DNN, i.e., $\boldsymbol{\theta} = \{\boldsymbol{W}^l, \boldsymbol{b}^l\}_{l=1}^L$. For a fully connected DNN, the number of parameters corresponding to $\boldsymbol{\theta}$ is $N_{\boldsymbol{\theta}} := \sum_{l=1}^L n_{l+1}(n_l + 1)$. We will refer $\mathcal{N}^1$ as the input layer, $\mathcal{N}^i, 1 < i < L$ as the hidden layers, and T^L as the output layer. We assume that every DNN neural network has an input, an output, and at least one hidden layer. Note that for the outer layer, there is no followed activation function. Figure 1 shows an example of a DNN model with 5 hidden layers with $[n_1, n_2, \cdots, n_7] = [6, 20, 10, 10, 10, 20, 1]$.

5.1.1. The recursive DNN model. In the case of a DNN model with consecutive hidden layers having an equal number of neurons, the weights and biases for those hidden layers can be easily shared due to the same data structure. We define the recursive DNN model as DNN models that share the parameters for all consecutive hidden layers with the same number of neurons. Therefore, A recursive DNN model could have significantly fewer total parameters than the corresponding non-recursive DNN model. For instance, the non-recursive DNN model described in Figure 1 has 931 parameters, while its corresponding recursive model has only 711 parameters. The contrast will become more pronounced when the number of hidden layers and hidden neurons increases. Our numerical results show that a recursive DNN model can benefit PDE solving.

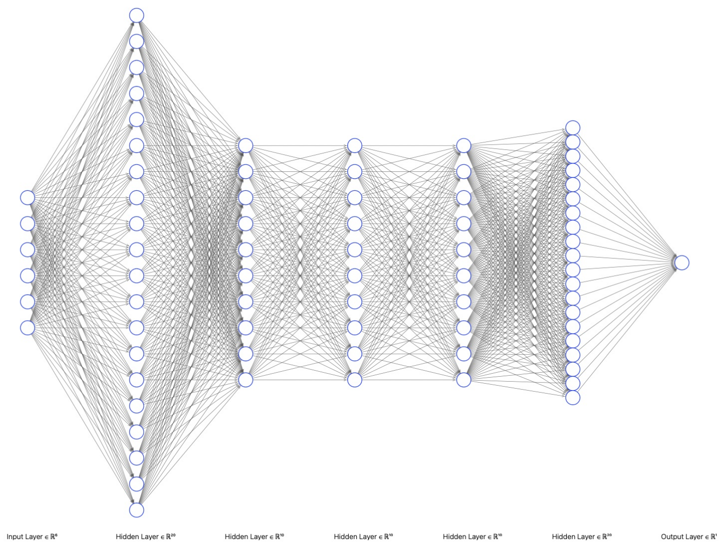


Fig. 1: A DNN network structure with 5 hidden layers

5.1.2. Comments about DNN. Although DNNs have been widely used as the primary neural network for solving PDE problems, their performance often falls short of expectations. When using DNNs within the Physically Informed Neural Networks (PINN) and Deep Ritz methods, achieving the desired accuracy typically requires thousands of iterations due to oscillations and stagnation. The method of WAN helps the algorithm escape local minima. However, despite this improvement, the number of iterations remains in the range of several thousand, as reported in [22] and demonstrated in our numerical results in Section 6. To enhance convergence, we explore different neural structures that approximate the trial functions with more efficacy.

5.2. XNODE model for parabolic PDE.. It has been demonstrated in [16] that for time-dependent parabolic problems, the XNODE model achieves much faster convergence than traditional deep neural networks. We believe this rapid convergence is attributed to the structure of the XNODE model, which emulates the residual network, and the direct embedding of the initial condition in the model.

Consider the following parabolic PDE defined on an arbitrary bounded domain $\mathcal{D} \subset [0,T] \times \mathbb{R}^d$, possibly representing a time dependent spatial domain,

$$
\begin{cases} \partial_t u - \nabla \cdot A(t,\boldsymbol{x})\nabla u + \boldsymbol{b}(t,\boldsymbol{x})\nabla u + c(u,\boldsymbol{x})u - f(\boldsymbol{x}) = 0 & \text{for } (t,\boldsymbol{x}) \in \mathcal{D}, \\ u(t,\boldsymbol{x}) = g(t,\boldsymbol{x}) & \text{on } \partial\mathcal{D}, \\ u(0,\boldsymbol{x}) - h(\boldsymbol{x}) = 0 & \text{on } \Omega(0). \end{cases} \tag{5.2}
$$

where $A = \{a_{ij}\}$, $\boldsymbol{b} = \{b_1, b_2, \cdots, b_n\}$, $f : \mathcal{D} \to \mathbb{R}$, $c : \mathbb{R} \times \mathcal{D} \to \mathbb{R}$ and $h : \Omega(0) \to \mathbb{R}$ are given, with $\Omega(t) := \{\boldsymbol{x} | (t,\boldsymbol{x}) \in \mathcal{D}\}$ denote the spatial domain of $\mathcal{D}$ when restricting time to be t. Note that c can be a non-linear function with respect to the first argument.

We now briefly introduce the XNODE model in [16]. For simplicity, we consider a time-independent domain in this paper, i.e., $\mathcal{D} = [0,T] \times \Omega$, where $\Omega \subset \mathbb{R}^d$ is bounded.

The XNODE model maps an arbitrary input $\boldsymbol{x} \in \mathbb{R}^d$ to the output $o_{\boldsymbol{x}}(t)_{t\in[0,T]} \in \mathcal{C}([0,T]$ by solving the following ODE problem:

$$\begin{cases} \frac{\mathrm{d}\boldsymbol{h}(t)}{\mathrm{d}t} = \mathcal{N}^{\mathrm{vec}}_{\boldsymbol{\theta}_2}(\boldsymbol{h}(t), t, \boldsymbol{x}), \quad \boldsymbol{h}(0) = \mathcal{N}^{\mathrm{init}}_{\boldsymbol{\theta}_1}(h(\boldsymbol{x})) \in \mathbb{R}^h. \\ o_{\boldsymbol{x}}(t) = \mathcal{L}_{\boldsymbol{\theta}_3}(\boldsymbol{h}(t)). \end{cases} \tag{5.3}$$

where $\mathcal{N}^{\mathrm{vec}}_{\boldsymbol{\theta}_2}$ and $\mathcal{N}^{\mathrm{vec}}_{\boldsymbol{\theta}_1}$ are DNN neural networks fully parameterized by $\mathcal{P}_{\boldsymbol{\theta}_2}$ and $\mathcal{P}_{\boldsymbol{\theta}_1}$ for the vector fields and the initial condition $\boldsymbol{h}(0)$ respectively. $\mathcal{L}_{\boldsymbol{\theta}_3}$ is a single linear layer parameterized by $\mathcal{P}_{\boldsymbol{\theta}_3}$. By $\Theta = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3)$ we denote the set of all trainable model parameters of the proposed XNODE model. Finally define

$$u_{\boldsymbol{\Theta}}(t, \boldsymbol{x}) := o_{\boldsymbol{x}}(t) \approx u(t, \boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Omega. \tag{5.4}$$

5.3. Pseudo-time XNODE model for static PDEs. In this subsection, we expand the XNODE model to handle stationary PDE problems. To simplify matters, we will focus on the following form of stationary PDE problem.

$$\begin{cases} -\nabla \cdot A(\boldsymbol{x})\nabla u(x) + \boldsymbol{b}(\boldsymbol{x}) \cdot \nabla u(\boldsymbol{x}) + c(u, \boldsymbol{x})u - f(\boldsymbol{x}) = 0 & \boldsymbol{x} \in \Omega = [0,1]^d, \\ u(\boldsymbol{x}) = g(\boldsymbol{x}) & \text{on } \partial\Omega. \end{cases} \tag{5.5}$$

The idea is to introduce a pseudo-time variable, which we choose from one of the spatial variables, x_i, to compensate for the absence of t, i.e., we let $t = x_i$ for some prefixed i. For simplicity, we choose $i = 1$ without loss of generality. The remaining variables $x_i, i = 2, \cdots, d$ will form the spatial variables in the XNODE model. More precisely, the spatial input point for the pseudo-time XNODE model should be modified as $\tilde{\boldsymbol{x}} = \{x_2, \cdots, x_d\}$. Similar to (5.4), we now define

$$u_{\boldsymbol{\Theta}}(\boldsymbol{x}) = u_{\boldsymbol{\theta}}(x_1, \tilde{\boldsymbol{x}}) := o_{\tilde{\boldsymbol{x}}}(x_1) \approx u(x_1, \tilde{\boldsymbol{x}}), \tag{5.6}$$

where $o_{\tilde{\boldsymbol{x}}}(x_1)$ is the numerical solution of (5.3).

5.4. Loss functions. We first recall the classical WAN loss function used in [16]:

$$L_{\mathrm{wan}}(\boldsymbol{\theta}, \boldsymbol{\eta}) := \log\left(\frac{|(\mathcal{A}(u_{\boldsymbol{\theta}}) - f, \phi v_{\boldsymbol{\eta}})|^2}{\|\phi v_{\boldsymbol{\eta}}\|^2_{L^2(\mathcal{D})}}\right) + \alpha L^2_{\mathrm{init}}(\boldsymbol{\theta}) + \beta L^2_{\mathrm{bdry}}(\boldsymbol{\theta}), \tag{5.7}$$

where α, γ are hyperparameters as penalty terms and

$$L_{\mathrm{init}}(\boldsymbol{\theta}) = \|u_{\boldsymbol{\theta}}(0, \boldsymbol{x}) - h(\boldsymbol{x})\|_{L^2(\Omega)}, \quad L_{\mathrm{bdry}}(\boldsymbol{\theta}) = \|u_{\boldsymbol{\theta}}(t, \boldsymbol{x}) - g(t, \boldsymbol{x})\|_{L^2([0,T]\times\partial\Omega)},$$

and $\phi(\boldsymbol{x})|_{\partial\Omega} = 0$. Here $u_{\boldsymbol{\theta}} \in W_{\boldsymbol{\theta}}$ and $v_{\boldsymbol{\eta}} \in V_{\boldsymbol{\eta}}$ where $W_{\boldsymbol{\theta}}$ and $V_{\boldsymbol{\eta}}$ are neural network function classes parameterized by $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$, respectively. In this paper, we use the classical DNN function class for $V_{\boldsymbol{\eta}}$. For $W_{\boldsymbol{\theta}}$, we will utilize and compare different neural network structures, which will be specified in each experiment. When the PDE problem is static, α is set to 0.

We also define the loss functions for the respective Cut-WAN and shifted Cut-WAN methods,

$$\begin{aligned} L_{\mathrm{cwan}}(\boldsymbol{\theta}, \boldsymbol{\eta}) &= |(\mathcal{A}(u_{\boldsymbol{\theta}}) - f, \phi v_{\boldsymbol{\eta}})| - \gamma_d \|\phi v_{\boldsymbol{\eta}}\|^2_V + \alpha L^2_{\mathrm{init}}(\boldsymbol{\theta}) + \beta L^2_{\mathrm{bdry}}(\boldsymbol{\theta}), \\ L_{\mathrm{scwan}}(\boldsymbol{\theta}, \boldsymbol{\eta}) &= L_{\mathrm{cwan}}(\boldsymbol{\theta}, \boldsymbol{\eta}) + \|\phi v_{\boldsymbol{\eta}}\|_{H^1_0(\Omega)}. \end{aligned} \tag{5.8}$$

During computation, the integrals are estimated using the Monte-Carlo sampling.

6. Numerical Results. The authors carried out the numerical results on a personal CPU device (Apple M1 Max chip with 32 GB memory and 10 total cores). The Adam optimization method is used for all presented numerical experiments.

6.1. Parabolic equations.

EXAMPLE 1. *Following the numerical example in [22, 16], we consider the following non-linear PDE problem in the form of a d-dimensional nonlinear diffusion-reaction equation (Eq.* (6.1)*) defined on a bounded domain* $\mathcal{D} \subset [0,1] \times [-1,1]^d$*:*

$$\begin{cases} \partial_t u - \triangle u - u^2 - f = 0 & \text{for } (t, \boldsymbol{x}) \in \mathcal{D} \\ u - g = 0 & \text{on } \partial\mathcal{D} \\ u(0, \boldsymbol{x}) - h(\boldsymbol{x}) = 0 & \text{on } \Omega(0), \end{cases} \tag{6.1}$$

where the exact solution is given by

$$u(t, \boldsymbol{x}) = 2 \sin\left(\frac{\pi}{2} x_1\right) \cos\left(\frac{\pi}{2} x_2\right) e^{-t}. \tag{6.2}$$

The hyperparameters for the XNODE model for u and DNN model for v used in these experiments are listed in Table 1, and their meanings are explained in Appendix A. The same hyperparameters were maintained across all experiments in Example 1 for the XNODE model. The recursive (nonrecursive) XNODE model $u_{\boldsymbol{\theta}}$ has 1501 (2161) trainable parameters, while the recursive (non-recursive) model of $V_{\boldsymbol{\eta}}$ has 5902 (23351) trainable parameters.

d	N_r	N_b	n_T	K_u	K_ϕ	α	β
5	4000	4000	20	2	1	10^7	10^5
ϵ	l_θ	l_η	u_{layers}	$u_{\text{hid-dim1}}$	$u_{\text{hid-dim2}}$	v_{layers}	$v_{\text{hid-dim}}$
10^{-2}	.015	.04	8	20	10	9	50

Table 1: Hyper parameter setting for Example 1

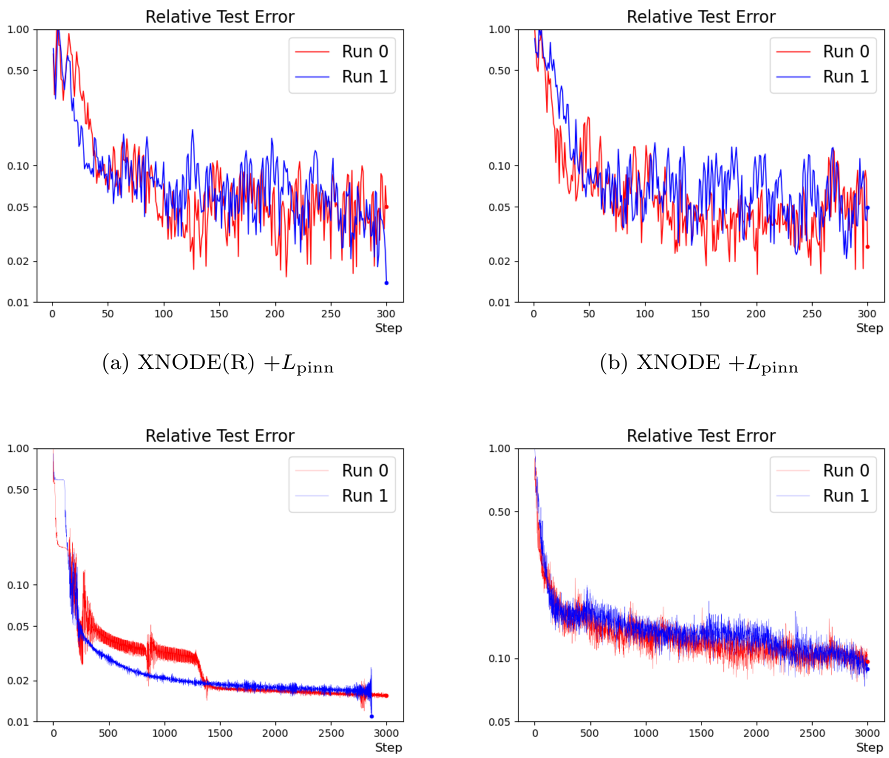
From Table 1, large penalty constants for α and β are utilized. We hypothesize that larger penalty constants can help strongly enforce initial and boundary conditions, which is beneficial in PDE solving using neural networks.

When utilizing the XNODE model to compute $u_{\boldsymbol{\theta}}$, the training process ceases either when the relative training error drops below 1% or after a maximum of 300 iterations. Conversely, if the DNN model is used to compute $u_{\boldsymbol{\theta}}$, the maximum number of iterations is set to 3000.

For a comparison, we first train the models using the PINN type loss function defined as follows:

$$L_{\text{pinn}}(\theta) = \|\partial_t u_{\boldsymbol{\theta}} - \triangle u_{\boldsymbol{\theta}} - u_{\boldsymbol{\theta}}^2 - f\|_{L^2(\Omega)} + \alpha L_{\text{init}}(\boldsymbol{\theta}) + \beta L_{\text{bdry}}(\boldsymbol{\theta}). \tag{6.3}$$

The L_{pinn} type loss function was initially introduced in the physics-informed neural network by Raissi et al. (2019) [19]. We have conducted experiments on L_{pinn} with the random initialization, and the results are displayed in Figure 2. We utilized the XNODE and DNN models for both the recursive and non-recursive versions. We note that the PINN loss function requires computing higher-order derivatives, which poses potential challenges. Firstly, the loss function becomes invalid when there is no strong solution, and secondly, computing these derivatives increases the computational time. In each step, the relative error in Figure 2 and subsequent figures is calculated



(a) XNODE(R) $+L_{\text{pinn}}$ (b) XNODE $+L_{\text{pinn}}$

(c) DNN(R)$+L_{\text{pinn}}$ (d) DNN$+L_{\text{pinn}}$

Fig. 2: Example 1: Relative L^2 Error versus Step for models using L_{pinn}

using a randomly chosen test set, denoted as X_{test}, that is the same size as the training data sets. More precisely, the relative error is computed as

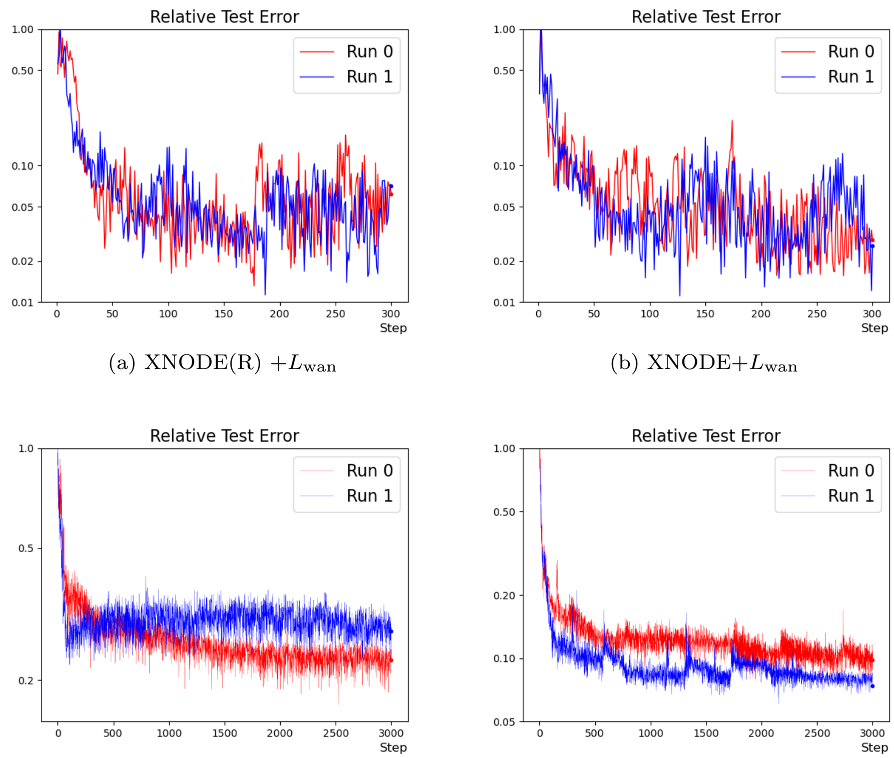
$$\sum_{\boldsymbol{x_i} \in X_{test}} \frac{\sum_i (u(\boldsymbol{x_i}) - u_\theta(\boldsymbol{x_i}))^2}{\sum_i u(\boldsymbol{x_i})^2}$$

It's worth noting that the test set is separate from the training set but has the same size.

In Figure 2, the training time for the DNN and XNODE models is about 2 and 8 seconds per step. The "(R)" after the model denotes the recursive model. In terms of L_{pinn}, it is evident that the DNN model exhibits slower convergence than the XNODE model, whether in recursive or non-recursive scenarios. When we compare figures in the right column from the left column, it is apparent that the recursive DNN model produces comparable results.

When using the XNODE model, from Figure 2a and Figure 2b, in both cases, relative errors approached the 7% threshold within the first 50 iterations. However, the errors then oscillate with large amplitude, requiring various steps to achieve the next level of accuracy.

In Figure 3, we then train the models using L_{wan} using the same models as in Figure 2. The training time for the DNN and XNODE models is about 2 and



(a) XNODE(R) $+L_{\text{wan}}$

(b) XNODE$+L_{\text{wan}}$

(c) DNN(R) $+L_{\text{wan}}$

(d) DNN $+L_{\text{wan}}$

Fig. 3: Example 1: Relative L^2 Error versus Step for models using L_{wan}

6 seconds per step. After analyzing both Figure 3c and Figure 3d, it is apparent that the utilization of (L_{wan}+ DNN) produces less desirable results compared to (L_{pinn}+ DNN) based on Figure 2. However, the combination of (L_{wan}+ XNODE) produces comparable results with (L_{pinn}+ XNODE). This indicates that XNODE is less sensitive to the chosen objective function.

Based on the observations from Figure 2 and Figure 3, we can deduce that the utilization of the XNODE network for $u_{\boldsymbol{\theta}}$ outperforms the DNN network in both the L_{wan} and L_{pinn} scenarios. However, it is noteworthy that when employing the XNODE network, the loss function during the training exhibits significant oscillation after reaching a certain level of accuracy for both L_{wan} and L_{pinn}. Additionally, in every scenario presented, the recursive model delivers comparable outcomes to its non-recursive counterpart.

We now evaluate the XNODE network using the loss functions L_{cwan} and L_{scwan} defined in (5.8), with the same models as shown in Figure 2. In each sub-figure in Figure 4, we present the results of five out of six consecutive and randomly initialized experiments under the specific setting to show generality. Each training step takes about 6 seconds.

Overall, after comparing Figure 2 and Figure 3 with Figure 4, we have noticed that L_{cwan} and L_{scwan} show uniformly faster and numerically more stable, i.e., less oscillations, convergence than L_{wan}. Moreover, we observe consistent/robust performance regardless of random initialization. In almost all experiments, the training relative error reaches the 1% relative error all within 200 steps.

When we compare the data in the right column to that of the left column, we notice that the recursive model performs just as well, if not better. Specifically, in the experiments depicted in Figure 4a, the stopping criteria were met at an average of 144 steps, with individual results of $137, 85, 177, 132$, and 190 for experiments 0 to 4, respectively. On the other hand, the non-recursive counterpart met the stopping criteria at an average of 164 steps, with individual results of $168, 125, 210, 153$, and 162 for experiments 0 to 4, respectively, as shown in Figure 4b. We also note that for the L_{scwan}, the comparison results for $\gamma_d = 0.5$ and $\gamma_d = 0.001$ are similar in this example. However, with $\gamma_d = 0.001$, we notice slightly more oscillaltions than the case of $\gamma_d = 0.5$.

In summary, the utilization of the XNODE network and the cutWAN and shifted CutWAN loss functions, i.e., L_{scwan} and L_{cscwan}in (5.8), has demonstrated a highly competitive model for solving high-dimensional parabolic PDE problem. In particular, solving the 5 dimensional non-linear parabolic problem in (6.1) takes only about 15 minutes for the training to reach the 1% relative error on a personal computer. Furthermore, the recursive model necessitates fewer parameters in contrast to non-recursive models. In comparison to classical numerical techniques such as the finite element method, which grows exponentially in the number of unknowns as the dimension expands, the potential benefit of our approach becomes more prominent as the disk space on a personal computer can rapidly become restricted with the classical approach.

We now consider the effect using the $H^{1/2}$ norm on the boundary based on (4.8). Define

$$\tilde{L}_{\text{bdry}}(\boldsymbol{\theta}) = \|u\|^{1/2}_{L^2(0,T,\partial\Omega)} \|\nabla_\Gamma u\|^{1/2}_{L^2(0,T,\partial\Omega)}$$

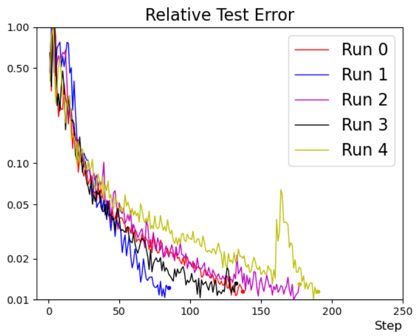
where $\nabla_\Gamma u = \nabla_{\boldsymbol{x}} u - (\nabla_{\boldsymbol{x}} u \cdot \boldsymbol{n})\boldsymbol{n}$ is the tangential gradient of u and $\boldsymbol{n}$ is the unit outer normal of Ω. We test the results replacing L_{bdry} in (5.7) by $\tilde{L}_{\text{bdry}}$ using the loss function L_{scwan} with $\gamma_d = 1/2$ (see Figure 5). The results in Figure 5a (average iteration number = 151) and Figure 5b (average iteration number = 173) are comparable to Figure 4c (average iteration number = 132) and Figure 4d (average iteration number = 161) . However, using $\tilde{L}_{\text{bdry}}$ resulted in an additional duration of approximately 1 second per iteration. For simplicity, we will use L_{bdry} for future experiments. It is worth noting that one can use L_{bdry} for the former iterations and switch to the more accurate $\tilde{L}_{\text{bdry}}$ for better accuracy and time efficiency. This can be necessary when g is of high frequency.

Remark 6.1 (How does $V_{\boldsymbol{\eta}}$ affect the method's performance?). In the proof, we require $V_{\boldsymbol{\eta}}$ to be rich enough to satisfy the stability condition. In this example, we tested multiple configurations for the $V_{\boldsymbol{\eta}}$ network, experimenting with different hidden layers and varying numbers of neurons. The results are all consistent with Figure 4. This indicates that the model is robust with $V_{\boldsymbol{\eta}}$ for this example.

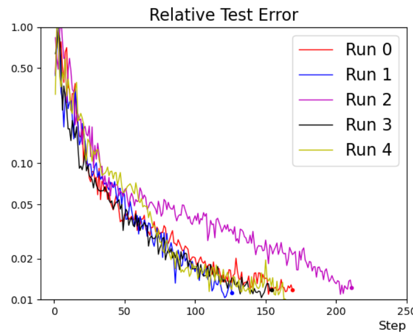
6.2. Stationary PDE problems.

EXAMPLE 2. *We now test the following high-dimensional problem as in [22].*

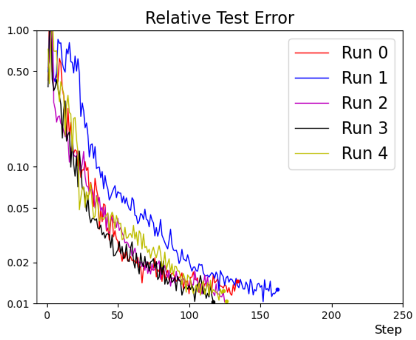
$$\begin{cases} -\triangle u(\boldsymbol{x}) = f & \boldsymbol{x} \in \Omega \\ u(\boldsymbol{x}) = g(\boldsymbol{x}) & \boldsymbol{x} \textit{ on } \partial\Omega, \end{cases}$$



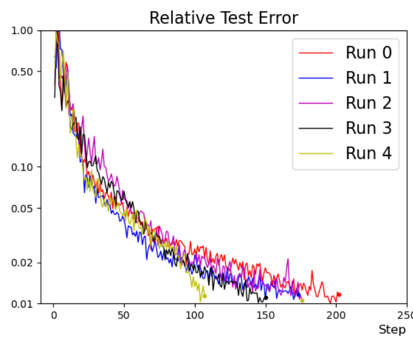
(a) XNODE(R) + L_{cwan} ($\gamma_d = 10^{-3}$)



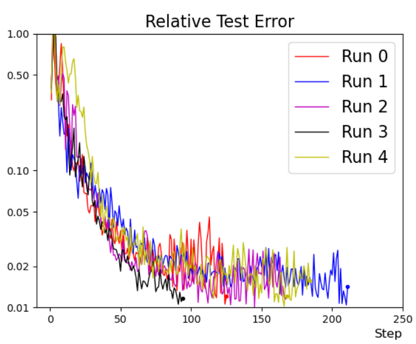
(b) XNODE + L_{cwan} ($\gamma_d = 10^{-3}$)



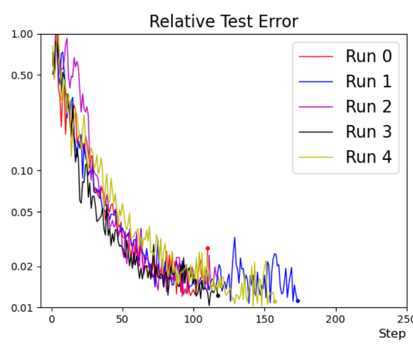
(c) XNODE(R) +L_{scwan}($\gamma_d = 0.5$)



(d) XNODE +L_{scwan}($\gamma_d = 0.5$)



(e) XNODE(R) +L_{scwan}($\gamma_d = 10^{-3}$)

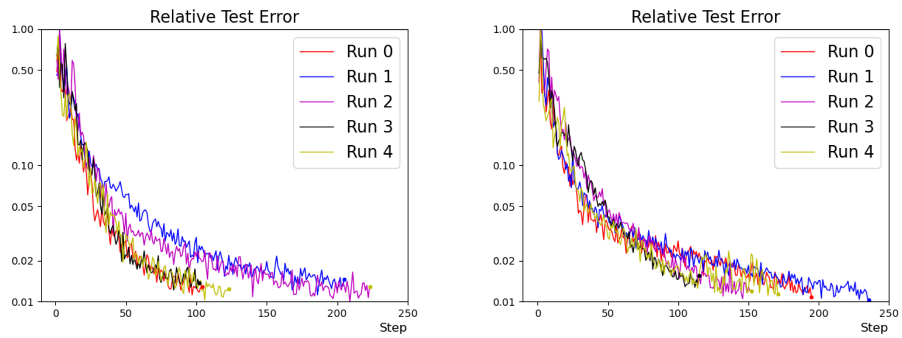


(f) XNODE +L_{scwan}($\gamma_d = 10^{-3}$)

Fig. 4: Example 1: Relative L^2 Error for XNODE models on L_{cwan} and L_{scwan}

where the true solution renders $u(\boldsymbol{x}) = \sum_{i=1}^{d} \sin\left(\frac{\pi}{2} x_i\right)$.

Observe that the boundary condition on the plane $x_1 = 0$ and $x_1 = 1$ now serves as the initial and terminal conditions in the pseudo-time XNODE model. Subsequently,



(a) XNODE(R) $+L_{\text{scwan}}(\gamma_d = 0.5)$ (b) XNODE $+L_{\text{scwan}}(\gamma_d = 0.5)$

Fig. 5: Example 1: The effect using $\tilde{L}_{\text{bdry}}(\boldsymbol{\theta})$

we adjust the initial loss and introduce the terminal loss as:

$$
\begin{aligned}
L_{\text{init}}(\theta) &= ||u_\theta(x_1 = 0) - g(x_1 = 0)||_{L^2(\Omega(0))}, \\
L_{\text{last}}(\theta) &:= ||u_\theta(x_1 = 1) - g(x_1 = 1)||_{L^2(\Omega(1))},
\end{aligned}
\tag{6.4}
$$

where $\Omega(t) := \{\boldsymbol{x} \in \Omega,\ x_1 = t\}$. We shall utilize the following loss functions for the pseudo-time XNODE model.

$$
\tilde{L}_{\text{wan,cwan,scwan}}(\theta, \eta) = L_{\text{wan,cwan,scwan}}(\theta, \eta) + \gamma L_{\text{last}}(\theta).
\tag{6.5}
$$

We experimented with testing the pseudo-time XNODE model with $d = 5$, using the same parameters as in Table 1 except for the penalty parameters. A grid search was performed to tune the hyperparameters α, β, and γ, which were restricted to the range $[10, 10^9]$. A optimal values found were $\alpha = \gamma = 10^5$ and $\beta = 10^7$.

We established a stopping criteria that ensured the relative training error was below 1% or the maximum iteration number less than 300. Each training step takes about 8 seconds.

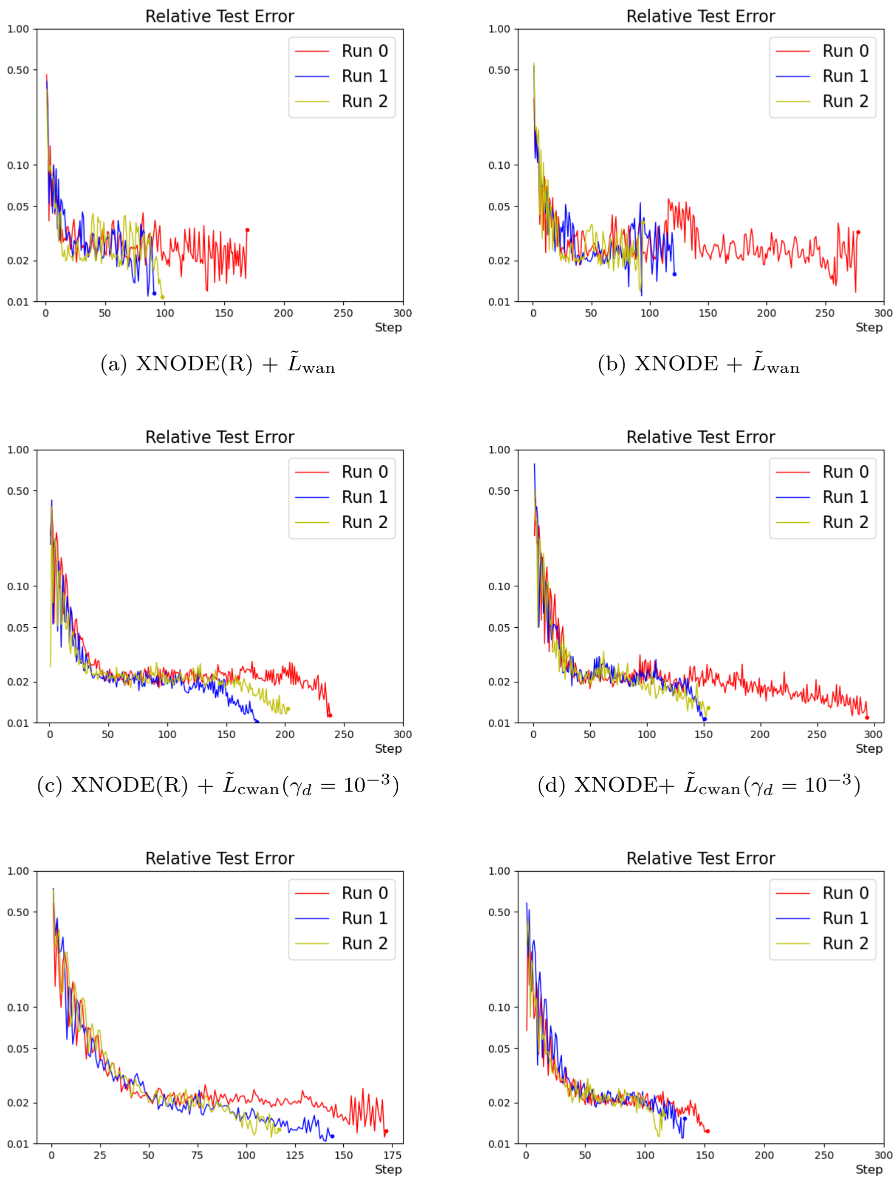
We have analyzed the loss functions $\tilde{L}_{\text{wan}}$, $\tilde{L}_{\text{scwan}}$ with $\gamma_d = 0.5$, and $\tilde{L}_{\text{cwan}}$ with $\gamma_d = 0.001$, as described in (6.5). We conducted tests on both the recursive and non-recursive models for each setting, and the outcomes are displayed in Figure 6. Each sub-figure in Figure 6 showcases the results of three consecutive experiments that were initialized randomly. The recursive and non-recursive models are utilized in the left and right columns respectively.

For $\tilde{L}_{\text{wan}}$, the recursive model in Figure 6a reached the stopping criteria at steps $169, 98$, and 91 (with an average of 120). Meanwhile, the non-recursive model in Figure 6b reached the stopping criteria at steps $277, 119$, and 93 (with an average of 163), based on three experiments for each.

For $\tilde{L}_{\text{cwan}}$, the recursive model in Figure 6c reached the stopping criteria at steps $237, 174$, and 200 (with an average of 203). Meanwhile, the non-recursive model in Figure 6d reached the stopping criteria at steps $293, 149$, and 153 (with an average of 198), based on three experiments for each.

For $\tilde{L}_{\text{scwan}}$, the recursive model in Figure 6e reached the stopping criteria at steps $172, 144$, and 117 (with an average of 144). Meanwhile, the non-recursive model in

Figure 6f reached the stopping criteria at steps 151, 132, and 115 (with an average of 132), based on three experiments for each.



(a) XNODE(R) + $\tilde{L}_{\text{wan}}$

(b) XNODE + $\tilde{L}_{\text{wan}}$

(c) XNODE(R) + $\tilde{L}_{\text{cwan}}(\gamma_d = 10^{-3})$

(d) XNODE+ $\tilde{L}_{\text{cwan}}(\gamma_d = 10^{-3})$

(e) XNODE+ $\tilde{L}_{\text{scwan}}(\gamma_d = 0.5)$

(f) XNODE(R)+ $\tilde{L}_{\text{Scwan}}(\gamma_d = 0.5)$

Fig. 6: Example 2. Relative L^2 Error versus Step using pseudo-time XNODE

In all XNODE experiments, the relative error quickly reached the 2% threshold within the first 35 iterations. Although the relative error oscillations generated by $\tilde{L}_{\text{wan}}$ are still greater than those of $\tilde{L}_{\text{cwan}}$ and $\tilde{L}_{\text{scwan}}$, it is worth noting that, in this particular case, the stopping criteria was achieved with slightly fewer iterations on average. We believe this faster convergence takes place thanks to the Poisson type PDE used in this example. For the Poisson problem, it is easy to see that $\|\cdot\|_{op}$ is the most natural norm to minimize. It has also been observed that the recursive model's performance is almost comparable to that of the non-recursive models in this example.

EXAMPLE 3.

$$-\nabla\cdot(a(x)\nabla u)+\frac{1}{2}|\nabla u|^2=f(x),\ \ in\ \Omega=[0,1]^d,\quad u(x)=g(x)on\ \partial\Omega. \tag{6.6}$$

where $a(x)=1+\|x\|^2$. *The true solution* $u(x)=\sin(0.5\pi x_1^2+0.5x_2^2)$.

In this problem, the non-linear term $\frac{1}{2}|\nabla u|^2$ presents a significant challenge. We will use the hyper-parameter set from Subsection 6.2 in all numerical tests with the pseudo-time XNODE model. Our objective in this example is to evaluate the performance of the pseudo-time XNODE model using various loss functions. We established the stopping criteria for the maximum number of iterations to be less than 600. The duration of each iteration is approximately 8.5 seconds.

We first test the loss function $\tilde{L}_{\text{wan}}$ by conducting three consecutive experiments with random initialization. The results are presented in Figure 7. The left/ right figure in Figure 7 shows the relationship between the number of steps and the L^2 relative error/ minimal L^2 relative error based on test sets. After the stopping criteria have been met, the minimal relative training L^2 error is 0.024, 0.056, and 0.023, respectively. We have observed a slower convergence rate compared to previous examples, which can be attributed to the more challenging non-linear term.

d	N_r	N_b	n_T	K_u	K_ϕ	α	β	γ
5	4000	4000	20	2	1	6×10^7	12×10^7	12×10^7
ϵ	l_θ	l_η	u_{layers}	$u_{\text{hid-dim1}}$	$u_{\text{hid-dim2}}$	v_{layers}	$v_{\text{hid-dim}}$	
10^{-2}	.015	.03	12	20	10	9	50	

Table 2: Hyper parameter setting for Example 3

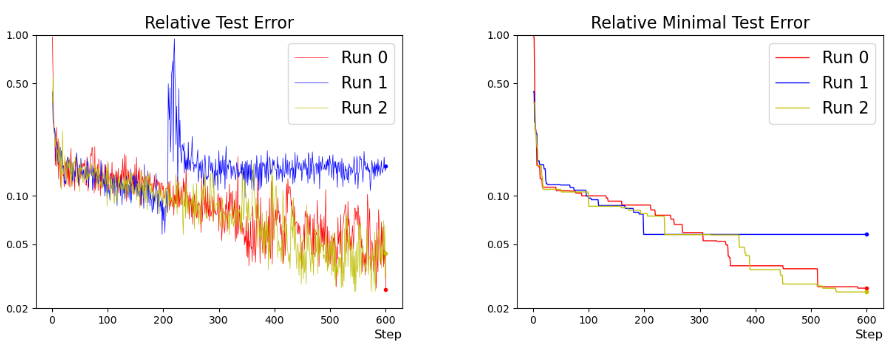


Fig. 7: Example 3. Pseudo time XNODE + $\tilde{L}_{\text{wan}}$

The results for $\tilde{L}_{\text{cwan}}$ and $\tilde{L}_{\text{scwan}}$ both with $\gamma_d=0.5$ are provided in Figure 8 and Figure 9, respectively. For all three experiments, the minimal relative error calculated from $\tilde{L}_{\text{cwan}}$ was $0.013, 0.016$, and 0.016. Meanwhile, the minimal relative error calculated from $\tilde{L}_{\text{scwan}}$ for the same experiments were $0.011, 0.012$, and 0.016. Therefore, in comparison to Figure 7, using $\tilde{L}_{\text{cwan}}$ and $\tilde{L}_{\text{scwan}}$ provides a slight advantage over $\tilde{L}_{\text{wan}}$. Due to the highly nonlinear nature of this problem, we believe that a more refined approach to tuning the hyperparameters is necessary to achieve greater accuracy in the results. We will consider this as a future work.

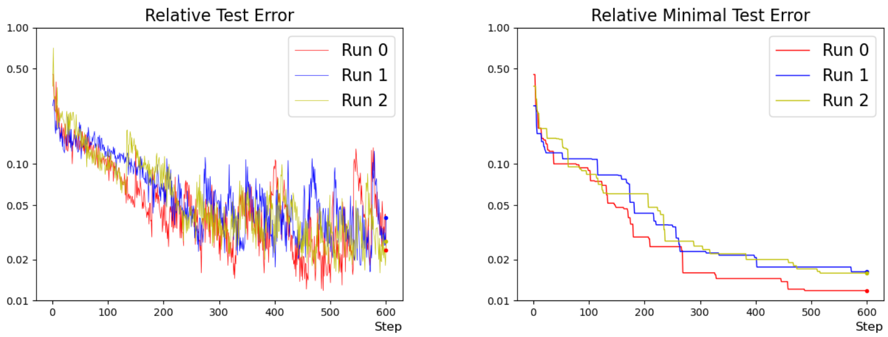


Fig. 8: Example 3. Pseudo time XNODE + $\tilde{L}_{\text{cwan}}$ ($\gamma_d = 0.5$)

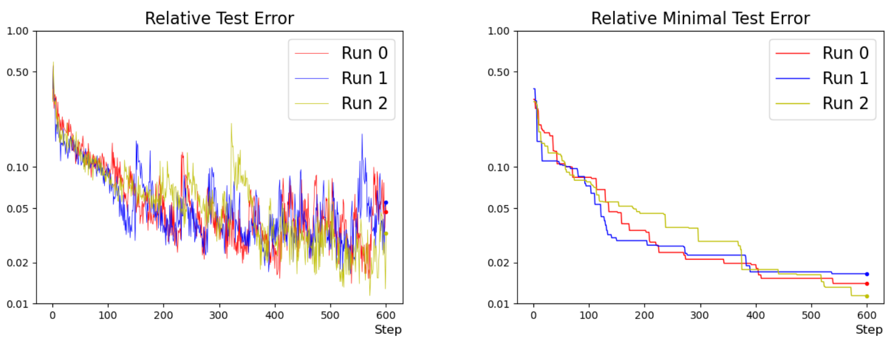


Fig. 9: Example 3. Pseudo time XNODE + $\tilde{L}_{\text{scwan}}$ ($\gamma_d = 0.5$)

7. Acknowledgement. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

Appendix A. Model set up XNODE-WAN Algorithm. The hyper-parameters for the neural networks are explained in the following table. For $V_{\boldsymbol{\eta}}$, we use a classical DNN network. The activation is set to be Tanh for the last hidden layer and ReLU for other hidden layers. Note that there is no activation function for the output layer. $\mathcal{N}^{\text{init}}_{\boldsymbol{\theta}_1}$ has one input layer, one hidden layer, and one output layer. The activation function after both the input and hidden layer is ReLU. $\mathcal{N}^{\text{vec}}_{\boldsymbol{\theta}_2}$ has u_{layers} of hidden dimensions and Tanh as the activation function.

REFERENCES

[1] G. Bao, X. Ye, Y. Zang, and H. Zhou, Numerical solution of inverse problems by weak adversarial networks, Inverse Problems, 36 (2020), pp. 115003, 31, https://doi.org/10.

Notation	Meaning
d	dimension for the physical domain (not including the time domain)
N_r	Number of sampled collocation points of the spatial domain
N_b	Number of sampled collocation points of the spatial domain boundary
n_T	Number of sampled time partition
K_u	Inner iteration to update weak solution u_θ or u_θ
K_ϕ	Inner iteration to update test function ϕ_η
α	Weight parameter of boundary loss L_{bdry}
γ	Weight parameter of initial and terminal losses L_{init} and L_{last}
ϵ	relative error tolerance
l_θ	Learning rate for the primal network
l_η	Learning rate for network parameter η of test function ϕ_η
u_{layers}	The number of hidden layers for $\mathcal{N}_{\theta_2}^{\text{vec}}$
$u_{\text{hid-dim1}}$	intermediate and output dimension for $\mathcal{N}_{\theta_1}^{\text{init}}$, input dimension for $\mathcal{N}_{\theta_2}^{\text{vec}}$
$u_{\text{hid-dim2}}$	intermediate dimension for $\mathcal{N}_{\theta_2}^{\text{vec}}$
v_{layers}	The number of hidden layers for $V_{\boldsymbol{\eta}}$
$v_{\text{hidd-dim}}$	intermediate and output dimension for $V_{\boldsymbol{\eta}}$.

Table 3: List of hyperparameters.

1088/1361-6420/abb447, https://doi.org/10.1088/1361-6420/abb447.

[2] E. Burman, A. Feizmohammadi, A. Münch, and L. Oksanen, Space time stabilized finite element methods for a unique continuation problem subject to the wave equation, ESAIM Math. Model. Numer. Anal., 55 (2021), pp. S969–S991, https://doi.org/10.1051/m2an/2020062, https://doi.org/10.1051/m2an/2020062.

[3] E. Burman and L. Oksanen, Data assimilation for the heat equation using stabilized finite element methods, Numer. Math., 139 (2018), pp. 505–528, https://doi.org/10.1007/s00211-018-0949-3, https://doi.org/10.1007/s00211-018-0949-3.

[4] A. N. Deshmukh, M. Deo, P. K. Bhaskaran, T. B. Nair, and K. Sandhya, Neural-network-based data assimilation to improve numerical ocean wave forecast, IEEE Journal of Oceanic Engineering, 41 (2016), pp. 944–953.

[5] M. W. M. G. Dissanayake and N. Phan-Thien, Neural-network-based approximations for solving partial differential equations, Communications in Numerical Methods in Engineering, 10 (1994), pp. 195–201, https://doi.org/https://doi.org/10.1002/cnm.1640100303, https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.1640100303, https://arxiv.org/abs/https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.1640100303.

[6] C. Duan, Y. Jiao, Y. Lai, X. Lu, Q. Quan, and J. Z. Yang, Analysis of deep ritz methods for laplace equations with dirichlet boundary conditions, 2021, https://arxiv.org/abs/2111.02009.

[7] M. Duprez and A. Lozinski, ϕ-FEM: a finite element method on domains defined by level-sets, SIAM J. Numer. Anal., 58 (2020), pp. 1008–1028, https://doi.org/10.1137/19M1248947, https://doi.org/10.1137/19M1248947.

[8] A. Ern and J.-L. Guermond, Theory and practice of finite elements, vol. 159, Springer, 2004.

[9] I. Gühring, G. Kutyniok, and P. Petersen, Error bounds for approximations with deep relu neural networks in w s, p norms, Analysis and Applications, 18 (2020), pp. 803–859.

[10] I. Gühring, M. Raslan, and G. Kutyniok, Expressivity of deep neural networks, Cambridge University Press, 2022.

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

[12] Q. Hong, J. W. Siegel, and J. Xu, A priori analysis of stable neural network solutions to numerical pdes, arXiv preprint arXiv:2104.02903, (2021).

[13] S. Karumuri, R. Tripathy, I. Bilionis, and J. Panchal, Simulator-free solution of high-dimensional stochastic elliptic partial differential equations using deep neural networks, Journal of Computational Physics, 404 (2020), p. 109120.

[14] S. Mahan, E. J. King, and A. Cloninger, Nonclosedness of sets of neural networks in sobolev spaces, Neural Networks, 137 (2021), pp. 85–96.

[15] M. H. Moeini, A. Etemad-Shahidi, V. Chegini, and I. Rahmani, Wave data assimilation using a hybrid approach in the persian gulf, Ocean Dynamics, 62 (2012), pp. 785–797.

[16] P. V. Oliva, Y. Wu, C. He, and H. Ni, Towards fast weak adversarial training to solve high dimensional parabolic partial differential equations using xnode-wan, 2021, https://arxiv.org/abs/2110.07812.

[17] M. Ostrovskii, Weak* sequential closures in Banach space theory and their applications, 2001, https://arxiv.org/abs/arXiv:math/0203139.

[18] P. Petersen, M. Raslan, and F. Voigtlaender, Topological properties of the set of functions generated by neural networks of fixed size, Foundations of computational mathematics, 21 (2021), pp. 375–444.

[19] 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, Journal of Computational Physics, 378 (2019), pp. 686–707.

[20] E. M. Stein, Singular integrals and differentiability properties of functions, Princeton Mathematical Series, No. 30, Princeton University Press, Princeton, N.J., 1970.

[21] N. Sukumar and A. Srivastava, Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks, Computer Methods in Applied Mechanics and Engineering, 389 (2022), p. 114333.

[22] Y. Zang, G. Bao, X. Ye, and H. Zhou, Weak adversarial networks for high-dimensional partial differential equations, J. Comput. Phys., 411 (2020), pp. 109409, 14, https://doi.org/10.1016/j.jcp.2020.109409, https://doi.org/10.1016/j.jcp.2020.109409.

[23] S. Zeng, Z. Zhang, and Q. Zou, Adaptive deep neural networks methods for high-dimensional partial differential equations, Journal of Computational Physics, 463 (2022), p. 111232.