Abstract—Prediction error quantification in machine learning has been left out of most methodological investigations of neural networks (NNs), for both purely data-driven and physics-informed approaches. Beyond statistical investigations and generic results on the approximation capabilities of NNs, we present a rigorous upper bound on the prediction error of physics-informed NNs (PINNs). This bound can be calculated without the knowledge of the true solution and only with a priori available information about the characteristics of the underlying dynamical system governed by a partial differential equation (PDE). We apply this a posteriori error bound exemplarily to four problems: the transport equation, the heat equation, the Navier–Stokes equation (NSE), and the Klein–Gordon equation.

Index Terms—A posteriori error estimator, certification, machine learning, Navier–Stokes, physics-informed neural network (PINN).

I. INTRODUCTION

PHYSICS-INFORMED machine learning is applied to numerous highly complex problems, such as turbulence and climate modeling [1], [2], [3], model predictive control [4], [5], and Hamiltonian system dynamics [6]. The systematic study of physics-informed machine learning as a method, however, remains open [7]. Part of this is the question about the quality of the forecast: how close is the prediction of the physics-informed neural network (PINN) [8] to the actual solution?

Beyond statistical evaluations, such as in [9], first steps to answer this question were taken in [10] by establishing rigorous error bounds for PINNs approximating the solution of ordinary differential equations (ODEs). We extend these results by deriving guaranteed upper bounds on the prediction error for problems modeled by linear partial differential equations (PDEs). Our methodology is thereby applicable to, e.g., the previously mentioned use cases of PINNs in computational science and engineering. Similar to [10], the error bound can be computed a posteriori without knowing the actual solution.

Since there are extensive studies on the convergence and approximation properties of (physics-informed) neural networks (NNs) [11], [12], [13], [14], it is clear from a theoretical point of view that a suitable NN can be found with the desired accuracy. These a priori results and the associated error estimates [14], [15], [16] are fundamentally different compared with the computable a posteriori error bounds we present here. Moreover, we like to stress that our analysis does not rely on statistical approaches or the Bayesian perspective but provides a rigorous error bound, which serves as a certificate for the prediction of the NN. The term certification used here differs from robustness certification as in [17].

Close in topic is the derivation of the error estimate based on norm inequalities in [14], which, however, is not directly computable due to unknown constants. In distinction, the error estimator presented here is computable which we demonstrate with several academic examples. Other earlier works [18], [19] also differ from the following work in fundamental aspects, such as that the error bounds derived in them are not guaranteed to hold, are restricted to a certain problem type, or require discretization steps.

At this point, we would like to emphasize that the results presented in Section III are independent of PINNs as the chosen methodology but can be applied to other surrogate modeling techniques. Nevertheless, since PINNs use the norm of the residual as a regularization term during the training process, they appear as a natural candidate to apply our residual-based error estimator.

Notation: We use bold notation for vectors and vector-valued functions and italic letters for operators, especially we use I to denote the identity operator. The notations \( \hat{u} = \partial_t u = (\partial u / \partial t) \) are used interchangeably to denote partial derivatives with respect to time. Similarly, we shorten derivatives with respect to spatial variables and use conventional notation for the divergence \( \nabla \cdot \phi = \text{div}(\phi) \), the gradient \( \nabla u = \text{grad}(u) \), and the Laplace operator \( \Delta u = \text{div}(
abla u) \) with respect to spatial variables only.

For normed spaces \( (X, \| \cdot \|_X) \) and \( (Y, \| \cdot \|_Y) \) and operator \( A : D(A) \subseteq X \rightarrow Y \), we define the induced operator norm \( \| A \|_X,Y := \sup_{x \in D(A), \| x \|_X \neq 0} (\| Ax \|_Y / \| x \|_X) \). In case \( X \subseteq Y \) and \( \| x \|_X = \| x \|_Y \), we shorten the notation to \( \| A \|_X \) and drop the subscript whenever the norm is clear from the context. For the spatial domain \( \Omega \subseteq \mathbb{R}^d \), we denote the boundary by \( \partial \Omega \) and define the size of \( \Omega \) as \( |\Omega| = \int_{\Omega} 1 \, \text{d}x \). The considered temporal domain is denoted by \( T = [0, t_f] \) with \( t_f \in \mathbb{R}^+ \). We use the conventional notation for spaces of \( p \)-integrable functions \( L^p(\Omega) \) and Sobolev spaces \( W^{k,p}(\Omega) \).
and $H^k(\Omega) = W^{k,2}(\Omega)$. We use the function classes $\mathcal{K}, \mathcal{KL}$ according to [20]

$$\mathcal{K} := \{\mu: \mathbb{R}_+ \rightarrow \mathbb{R}_+ | \mu(0) = 0, \mu \text{ cont., strict. incr.}\}$$

$$\mathcal{L} := \{\mu: \mathbb{R}_+ \rightarrow \mathbb{R}_+ | \mu \text{ cont., strict. decr., } \lim_{t \to \infty} \mu(t) = 0\}$$

$$\mathcal{KL} := \{\mu: (\mathbb{R}_+)^2 \rightarrow \mathbb{R}_+ | \mu(\cdot, t) \in \mathcal{K} \forall t \geq 0; \mu(s, \cdot) \in \mathcal{L} \forall s \geq 0\}. \tag{1}$$

II. Problem Description

In the following, we aim at solving the boundary and initial value problem (BIVP): find $u: \mathbb{T} \times \Omega \rightarrow \mathbb{R}^n$ such that

$$\begin{cases}
\partial_t u = Au, & \text{in } \mathbb{T} \times \Omega \\
u = u_0, & \text{in } \{t = 0\} \times \Omega \\
Bu = u_0, & \text{in } \mathbb{T} \times \partial\Omega.
\end{cases} \tag{2}$$

The initial value $u_0$ lies in the domain of definition of the linear differential operator $A: D(A) \subseteq X \rightarrow X$, meaning $u_0 \in D(A)$. The bounded linear operator $B: D(A) \rightarrow U$ represents the type of the boundary condition and maps to the Banach space $U$ which contains the boundary condition $u_0$. As an example, for the Dirichlet boundary condition $B = T \mathbb{I}$ is suitable, wherein $T$ denotes the trace operator.

With this setup given, the objective can be intuitively formulated as follows.

Problem 1: Given an approximate solution $\hat{u}$ for the BIVP (2), which is determined, for example, by approximating the terms $\partial_t u = Au + f$ by computing the residuals $R_0 \in D(A')$ and $R: \mathbb{T} \rightarrow X$. For an approximate solution $\hat{u}$ given by an NN, we can determine the terms $R, R_0$ by computing the residuals $R_0 := \hat{u}(0, \cdot) - u_0$ in $\Omega$ and $R := \partial_t \hat{u} - A\hat{u}$ in $\mathbb{T} \times \partial\Omega$. \tag{4}

This calculation requires $\hat{u}$ to be differentiable in time, which constrains the choice of the activation function of the NN. In addition, we assume the following for the BIVP (3).

Assumption 2: The perturbed BIVP (3) is such that $R$ is Lipschitz continuous in $t \in \mathbb{T}$ and $u_0 + R_0 - B_0u_0 \in D(A)$.

Regarding the approximation by an NN, for most activation functions, the Lipschitz condition is satisfied, and only the second condition requires a more detailed investigation of the corresponding function spaces.

The second problem involves additionally a perturbation in the boundary condition

$$\begin{cases}
\partial_t \hat{u} = A\hat{u} + R, & \text{in } \mathbb{T} \times \Omega \\
\hat{u} = u_0 + R_0, & \text{in } \{t = 0\} \times \Omega \\
B\hat{u} = u_0 + R_b, & \text{in } \mathbb{T} \times \partial\Omega
\end{cases} \tag{5}$$

Remark 1: The results derived below can be generalized directly to systems modeled by inhomogeneous PDEs of type

$$\partial_t u = Au + f$$

with $f(t) \in D(A)$ and $f$ independent of $u$. To simplify notation, we continue all the investigations without $f$.

Similar to [10] for a finite-dimensional setting, the NN approximation of the solution of the BIVP (2) is interpreted as the solution of a perturbed problem. Here, we consider two types of perturbed problems, the first has the same boundary condition as the unperturbed problem, but has perturbation terms in the initial condition and the differential equation. The perturbed problem reads

$$\begin{cases}
\partial_t \hat{u} = A\hat{u} + R, & \text{in } \mathbb{T} \times \Omega \\
\hat{u} = u_0 + R_0, & \text{in } \{t = 0\} \times \Omega \\
B\hat{u} = u_0 + R_b, & \text{in } \mathbb{T} \times \partial\Omega
\end{cases} \tag{3}$$

We consider this system as a boundary control system (BCS), cf., [24] and use the notion of input-to-state-stable (ISS) systems, see [20], to later derive rigorous error bounds.

Definition 1: System (5) is said to be ISS with respect to the boundary term $u_0 + R_b$, if there exist $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}$ such that

$$\|\hat{u}(t, \cdot)\| \leq \beta \left(\|u_0(0, \cdot)\| + \gamma (\|R_b + u_0\|_{L^{\infty}(0, t; \partial\Omega)})\right). \tag{6}$$

In the following, we consider an ISS BCS, in which $u_0 = 0$. As an example, for the heat equation (see Section V-A), we use $\gamma(s) = 1/3 s$. Furthermore, we can use [25] to find our setup and the characteristic constants $M, \omega$ of the semigroup as used in Corollary 1 that $\beta(s, t) = Me^{\omega t} s$.

Remark 2: We present reasoning and theorems based on the notion of ISS; however, all the results can be transferred easily to integral ISS (iISS) systems [20], [25].

Assumption 3: The BCS (5) is ISS with respect to the boundary values $u_0 + R_b$. 

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
With these definitions and assumptions given, we can clarify the intuitive understanding of Problem 1 and present the following rigorous formulation.

**Problem 2:** Assume that the operators $A$ and $B$ in the BIVP (2) satisfy Assumption 1 and let $u$ denote the unique mild solution of (2). Furthermore, let Assumptions 2 and 3 hold true and let $\hat{u}$ denote the mild solution of (3) or (5). Find a certificate $\varepsilon(t)$: $\mathbb{T} \to \mathbb{R}_+$ such that

$$\|\hat{u}(t, \cdot) - u(t, \cdot)\|_X \leq \varepsilon(t)$$

and can be computed a posteriori without knowledge of the true solution $u$.

### III. Error Estimation

We divide this section, which contains the main results of this work, into three sections, whereof the first two are concerned with the two previously introduced perturbed BIVPs. Here, we adapt the terminology of hard and soft boundary constraints in NNs [26], meaning that hard boundary constraints denote the exact fulfilledness of the boundary conditions in BIVP (3), and soft boundary constraints correspond to the perturbed BIVP (5). For the networks used to generate these approximate solutions, we require a network to be designed to fulfill the boundary condition exactly for hard boundary constraints or force the network to approximate the boundary condition by a suitable loss term for soft constraints. In the third subsection, we give an auxiliary result to use numerical integration methods to compute the error bounds.

#### A. Hard Boundary Constraints

An NN approximation with hard boundary constraints can be interpreted as solution to the perturbed BIVP (3) by defining the residuals as in (4).

**Theorem 1:** Let the BIVP (2) be given and suppose that Assumption 1 is fulfilled. Furthermore, let $\hat{u}$ be a mild solution to (3) and presume that the terms $R$ and $R_0$ satisfy Assumption 2. Then

$$\varepsilon(t) := \|\mathcal{S}(t)\|_X \xi_0 + \int_0^t \|\mathcal{S}(t-s)\|_X \xi(s) \, ds$$

(7)

wherein $\|R_0\|_X \leq \xi_0 \in \mathbb{R}_+$ and $\xi: \mathbb{T} \to \mathbb{R}_+$, $\xi$ bounded and continuous with $\|R(t)\|_X \leq \xi(t)$ is a certificate for $\hat{u}$.

**Proof:** The dynamics of the error $e(t, x) := \hat{u}(t, x) - u(t, x)$ is governed by the BIVP

$$\partial_t e = Ae + R \quad \text{in} \quad \mathbb{T} \times \Omega$$
$$e = R_0 \quad \text{in} \quad \{ t = 0 \} \times \Omega$$
$$\mathcal{B}e = 0 \quad \text{in} \quad \mathbb{T} \times \partial \Omega$$

(8)

with $R$ and $R_0$ as defined in (4). Using Assumption 1, the unique mild solution [27, Ch. 4, Corollary 2.11] of BIVP (8) is given by

$$e(t, \cdot) = S(t) R_0 (\cdot) + \int_0^t S(t-s) R(s, \cdot) \, ds$$

(9)

Controlling the residuals from above by $\xi, \xi_0$, one can limit $\|e(t, \cdot)\|_X$ from above by $\varepsilon$ as defined in (7).

**Corollary 1:** Let all the prerequisites for Theorem 1 be fulfilled. Then there exist constants $M \geq 1$ and $\omega \in \mathbb{R}$ such that

$$\varepsilon(t) \leq \tilde{\varepsilon}(t) := M e^{\omega t} \xi_0 + \int_0^t M e^{\omega(t-s)} \xi(s) \, ds$$

(10)

1) If furthermore $\{ S(t) \}_{t \geq 0}$ is an $\alpha$: $\omega$-contraction semigroup, then $M = 1$ and 2) exponentially decaying semigroup, then $\omega < 0$.

**Proof:** By applying general results on strongly continuous semigroups [27, Ch. 1, Th. 2.2] to (7), the existence of $M \geq 1$ and $\omega \geq 0$ for (10) is shown. An $\omega$-semigroup decomposition is characterized by $M = 1$. By definition of an exponentially decaying semigroup, there exist constants $M \geq 1$ and $\mu > 0$ such that $\|S(t)\|_X \leq M e^{-\mu t}$.

Although Corollary 1 only establishes the existence of $M$ and $\omega$, we emphasize that these constants can be computed in several applications; see Section V. For example, we can use the spectral mapping theorem (e.g., [27, Ch. 2, Th. 2.4]) to determine the growth bound $\omega$ to be the real part of the largest eigenvalue of the infinitesimal generator of the semigroup.

#### B. Soft Boundary Constraints

Enforcing boundary values with soft boundary constraints introduces an additional perturbation in the boundary, which is reflected in the perturbed BIVP (5).

**Theorem 2:** Let the BIVP (2) fulfill Assumption 1 and assume that it is ISS according to Assumption 3. Let the solution $\hat{u}$ to the perturbed problem (5) satisfy Assumption 2 with the second part replaced by $u_0 + R_0 - B_0(u_0 + R_0) \in D(A)$, then

$$\varepsilon(t) := Me^{\omega t} \xi_0 + \int_0^t Me^{\omega(t-s)} \xi(s) \, ds + \gamma \left( \|R_b\|_{L^\infty(0,t;\partial \Omega)} \right)$$

with $\|R_0\|_X \leq \xi_0 \in \mathbb{R}_+$ and $\xi: \mathbb{T} \to \mathbb{R}_+$, $\xi$ bounded and continuous with $\|R(t)\|_X \leq \xi(t)$ is a certificate for $\hat{u}$.

**Proof:** We introduce $\tilde{u}$ as a mild solution to (3) with the same perturbation $R(t)$ considered in the BIVP (5) but with $R_0 = 0$. Since a mild solution $u$ to (2) exists, and Assumption 2 is valid, there exists a unique mild solution to (3). Hence, the error can be split into

$$\|\hat{u} - u\|_X \leq \|\hat{u} - \tilde{u}\|_X + \|\tilde{u} - u\|_X$$

According to Corollary 1, we can bound the latter part by (10). For the first contribution, we proceed similar to the proof of Theorem 1 by observing that the dynamics of the error $\tilde{e}(t, x) := \hat{u} - \tilde{u}$ is determined by the BIVP

$$\partial_t \tilde{e} = A\tilde{e} \quad \text{in} \quad \mathbb{T} \times \Omega$$
$$\tilde{e} = R_0 \quad \text{in} \quad \{ t = 0 \} \times \Omega$$
$$\mathcal{B}\tilde{e} = R_b \quad \text{in} \quad \mathbb{T} \times \partial \Omega$$

Now, we use the ISS property with $\beta(s, t) = Me^{\omega t}$ [25] (with $\beta$ as in Definition 1) to conclude

$$\|\tilde{e}(t, \cdot)\|_X \leq Me^{\omega t} \|R_0\| + \gamma \left( \|R_b\|_{L^\infty(0,t;\partial \Omega)} \right)$$

Adding the contributions to the error concludes the proof. □
C. Efficiency of the Error Estimator

The efficiency and sharpness of the error estimator are limited by the estimations made during its derivation. First, the triangle inequality is applied to the norm of (9), i.e.,
\[
\|e(t, \cdot)\|_X \leq \|S(t) R_0(\cdot)\|_X + \left\| \int_{0}^{t} S(t-s) R(s, \cdot) \, ds \right\|_X.
\]
This inequality is sharp when there is a \( k_1 \geq 0 \) such that \( S(t) R_0(\cdot) = k_1 \int_{0}^{t} S(t-s) R(s, \cdot) \, ds \) [28].

Next, the norm of the integral is limited from above by the integral of the norm for the second summand, which is sharp under similar conditions as the triangle inequality. This means that for every \( s_1, s_2 \in (0, t) \), there needs to be a \( k_{s_1,s_2} \geq 0 \) so that the equality \( S(t-s_1) R(s_1, \cdot) = k_{s_1,s_2} S(t-s_2) R(s_2, \cdot) \).

As a third step, we estimate \( \|S(t) R_0(\cdot)\|_X \) and similarly the integrand of the second summand with the operator norm, which by definition is the smallest number \( C(t) \) satisfying \( \|S(t)x\|_X \leq C(t) \|x\|_X \) for all \( x \in X \). Finally, we use the growth bound of the operator semigroup to estimate \( \|S(t)\|_X \), where the efficiency then depends on the characteristics of the semigroup of operators under investigation. In the case when soft boundary constraints are used, the efficiency of the error estimator depends additionally on the sharpness of the ISS gain \( \gamma \) used. These dependences become evident in the numerics section.

A simple 2-D ODE in suitable domain \( R, R_0 \) can be constructed as an example for the case where the error estimator is sharp. Consider the ODE given by
\[
\dot{u} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} u
\]
in \( \mathbb{R}^2 \) equipped with the 2-norm \( \| \cdot \|_2 \) and denote the unit vectors by \( \hat{e}_1 := \begin{bmatrix} 1, 0 \end{bmatrix}^T \) and \( \hat{e}_2 \), respectively. For \( c_1, c_2 > 0 \) assume that \( R_0 = c_1 \hat{e}_2 \) and \( R(t) = c_2 \hat{e}_2 \). Then it is easily verifiable that there exists \( b_1, b_2 \in \mathbb{R} \) such that \( S(t) R_0 = b_1 \hat{e}_2, R(t) = b_2 \hat{e}_2 \), then the estimation step \( \|S(t) R_0(\cdot)\|_X \leq \|S(t)\|_X \|R_0(\cdot)\|_X \) replaces the factor \( e^t \) by \( e^{2t} \), which leads to an increasing overestimation of the error for progressing time.

D. Integral Approximation

Following [10], we give an auxiliary result to apply the previous theorems using numerical integration techniques to compute the integrals in Theorem 1 and 2 and Corollary 1, respectively. Here, we use the composite trapezoidal rule to approximate the integral as
\[
\int_{0}^{t} M(t-s) \xi(s) \, ds \approx M(t) \hat{I}_n \left( 0, t, e^{-\omega s} \xi(\cdot) \right)
\]
using \( n \) equally spaced subintervals \([s_i, s_{i+1}]\), \( 0 = s_0 < \cdots < s_n = t \), meaning \( s_{i+1} - s_i = \Delta s \). Therein the right-hand side is defined via
\[
\hat{I}_n \left( t, e^{-\omega s} \xi(\cdot) \right) := \sum_{i=1}^{n-1} \Delta s \left( e^{-\omega s_i} \xi(s_i) + e^{-\omega s_{i+1}} \xi(s_{i+1}) \right).
\]

Numerical computation of the integral introduces an additional error
\[
\epsilon_{\text{int}} := Me^{\text{out}} K \frac{t^3}{12n^2}
\]
assuming that \( \xi \) is sufficiently smooth and that \( K \) fulfills
\[
\sup_{s \in [0, T]} \left| \frac{d^2}{ds^2} \left( e^{-\omega s} \xi(s) \right) \right| \leq K.
\]
Summarizing the previous discussion yields the following result.

**Proposition 1:** Consider the BIVPs (2) and (3) [or (5)] and suppose that Assumptions 1 and 2 (and 3) are satisfied. Then
\[
\epsilon(t) \leq Me^{\text{out}} \xi_0 + Me^{\text{out}} \hat{I}_n \left( t, Me^{-\omega s} \xi(s) \right) + \epsilon_{\text{int}} + \epsilon_{bc}
\]
with \( \xi_0 \) and \( \xi \) as in Theorem 1 under the additional assumption that \( \xi \in C^3(T, \mathbb{R}) \). Furthermore, the numerical integral (11) and the error of the numerical integration (12) are used assuming \( K \) can be found according to (13). When considering the BIVP (5), \( \epsilon_{bc} = \gamma(\|R_0\|L^\infty(0, t; \mathbb{R})) \), else the term vanishes.

Remark 3: The additional integration error considered in Proposition 1 accounts for numerical integration over time only. The numerical error introduced by integrating over space is already included and part of \( \xi(t) \) and \( \xi_0 \). A critical discussion of the error introduced by numerical integration can be found in [10].

Remark 4: The previous derivations also show that an additional regularization of the derivatives (cf., [26]) of the residual could prove useful to reduce the integration error.

IV. METHODOLOGY: PINNS

We want to apply the results from Section III to PINNs as introduced in [8]. Therein the data-driven loss used during training of feedforward NNs is supplemented by a contribution reflecting the PDE (the physics-informed contribution), yielding
\[
L = \frac{1}{1 + \kappa} L_{\text{eq}} + \frac{\kappa}{1 + \kappa} L_{\text{data}} + \rho L_{\text{space}}.
\]
The parameter \( \kappa \in \mathbb{R}_+ \) defines the relationship between data-driven and physics-informed contributions, where \( \rho \geq 0 \) scales the contribution of soft boundary conditions and further spatial properties; see below for further details. The data-driven contribution is encoded in
\[
L_{\text{data}} = \sum_{i=1}^{N_{\text{data}}} \left\| \hat{u} \left( t_{\text{data},i}, x_{\text{data},i} \right) - u_{\text{data},i} \right\|^2
\]
for a given training dataset
\[
\left( t_{\text{data},i}, x_{\text{data},i}, u_{\text{data},i} \right)_{i=1,\ldots,N_{\text{data}}}
\]
We need to know the true solution \( u_{\text{data},i} = u(t_{\text{data},i}, x_{\text{data},i}) \) for this, and hence we reduce data-driven contributions to data given by the initial condition

\[
L_{\text{data}} = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \| \hat{u}(0, x_{\text{data},i}) - u_0(x_{\text{data},i}) \|^2 .
\]

The physics-informed loss

\[
L_{\text{eq}} = \frac{1}{N_{\text{eq}}} \sum_{i=1}^{N_{\text{eq}}} \| \partial_t \hat{u}(t_{\text{eq},i}, x_{\text{eq},i}) - A\hat{u}(t_{\text{eq},i}, x_{\text{eq},i}) \|^2
\]
evaluates the residual of the PDE for a set of \( N_{\text{eq}} \) collocation points \( (t_{\text{eq},i}, x_{\text{eq},i}) = 1, \ldots, N_{\text{eq}} \). In particular, this contribution to the loss term aims at minimizing \( R \), which is used in the error bounds derived in Section III.

The third contribution \( L_{\text{space}} \) accounts for space constraints such as the divergence-freeness of solutions to the Navier–Stokes equations (NSEs) or soft boundary constraints. The parameter \( \rho \) scales this loss relative to the other two contributions.

We use NNs with \( n_i \in \mathbb{N} \) input neurons, \( n_o \in \mathbb{N} \) output neurons, and \( n_{\text{hd}} \in \mathbb{N} \) hidden layers with \( n_{\text{ln}} \in \mathbb{N} \) neurons each as basis for networks used in the numerical experiments. In Sections IV-A and IV-B, we discuss the modifications in network topology or loss functions which are necessary to realize hard and soft boundary constraints.

Remark 5: In case no further control parameters are given, \( n_i \) corresponds to the dimension of \( \mathbb{T} \times \Omega \), i.e., \( n_i = d + 1 \). Similarly, assuming that no hidden variables are included in the PDE which are not described explicitly in the PDE (like the pressure in the NSEs, cf., Section V-C), \( n_o \) corresponds to the dimension of \( u \), i.e., \( n_o = n \).

A. Hard Boundary Constraints

The implementation of hard boundary constraints requires a modification of the network topology. For the following numerical experiments, we need topologies to account for the Dirichlet or periodic boundary conditions. For the Dirichlet boundary conditions, we extend the purely sequential core NN by a second path as depicted in Fig. 1. In the homogeneous case, we use a mask (as presented for arbitrary boundary conditions in [29] and [30] and therein called length factor) for a d-dimensional rectangular domain of width \( w = (w_0, \ldots, w_d) \) with center \( x_0 = (x_1,0, \ldots, x_d,0) \) of the type

\[
\lambda(x) = \prod_{i=1}^{d} \left( \frac{\left( \frac{w_i}{2} \right)}{2} - \left( x_i - x_{i,0} \right) \right) .
\]

For inhomogeneous problems, we add a suitable offset

\[
\chi(t, x) = u_b(t, x) \quad \text{for} \quad t \in \mathbb{T}, \quad x \in \partial \Omega .
\]

We can summarize the model to be

\[
\hat{u}(t, x) = \lambda(x) \cdot \hat{u}(t, x) + \chi(t, x)
\]

where \( \hat{u} \) is the output of the sequential core NN.

For periodic boundary condition, we consider again a d-dimensional rectangular domain of width \( w = (w_0, \ldots, w_d) \) and follow [31] to construct a periodic base layer for one (the \( i \)th) space dimension with \( n_p \) units as

\[
\sigma(A \cdot \cos(v x + \phi) + B) \quad (16)
\]
wherein \( \sigma \) is the activation function. Furthermore, \( A, B, v, \phi \in \mathbb{R}^{n_p} \). The frequency \( v \) may not be trained but needs to be fixed by the period \( w_i \) in the respective space direction as \( v = 2\pi / w_i \cdot I_{n_p} \). Here, \( I_{n_p} \) denotes the vector of all ones in \( \mathbb{R}^{n_p} \).

The nonperiodic input variables are processed by a dense layer. Exemplarily, a nonsequential network realizing this is shown in Fig. 2. To enhance training performance compared with [31], we introduce dense layers before the periodic base layers to model the dependence of \( A^i, B^i, \phi^i \) on \( t \) as an extension.

B. Soft Boundary Constraints and Other Restrictions

In case, the solution shall meet additional constraints which are not satisfied automatically by the NN, e.g.,
Proposition 1 for notation simplicity. Furthermore, if we apply \( \varepsilon \) the integrated error from the misrepresentation of the temporal needs to be taken into account according to Theorem 2. If soft boundary constraints are used, the boundary error contributions by splitting the predicted error in a part reflecting the propagation of the error in the initial condition \( \varepsilon \) according to (13) initially using training collocation points only and

\[
\mu = 0.1 \frac{1}{N_{\text{eq}}} \sum_{i=1}^{N_{\text{eq}}} | R \left( t_{\text{eq},i}, x_{\text{eq},i} \right) |.
\]

This is the same approach as taken in [10]. By approximating \( K \) by a pointwise upper limit

\[
K \approx \sup_{i \in [1,...,N_{\text{eq}}]} \left| \frac{d^2}{dt^2} \left( e^{-\omega t} \xi \left( t \right) \right) \right|_{t=t_{\text{eq},i}}
\]

the error estimator is not rigorous anymore. However, by introducing increasing numbers of support points, the estimation of \( K \) can be made arbitrarily accurate.

We setup NNs based on TensorFlow [32] using the hyperbolic tangent as the activation function. Problem-specific parameters such as network size are mentioned in the relevant sections. A set of common parameters and settings are listed here: We train using the Limited-Memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) optimizer [33] with learning rate 0.1 unless stated otherwise. The NNs rely on the hyperbolic tangent as activation function, and the data-driven and physics-informed contributions are equally weighted with \( \kappa = 1 \) unless explicitly mentioned. Collocation points are selected using Latin hypercube sampling [34].

For testing the NN and our error estimator, we generate a grid of equally spaced data points in the complete spatial domain and evaluate the prediction of the NN and the error estimator at various points in time.

### A. Heat Equation

We consider the scalar BIVP

\[
\begin{align*}
\partial_t u (t, x) &= \frac{1}{5} \partial_{xx} u (t, x) \\
u (0, x) &= \sin \left( 2\pi x \right) \quad \text{for} \quad x \in \Omega \\
u (t, 0) &= 0 = u (t, 1) \quad \text{for} \quad t \in \mathbb{T}
\end{align*}
\]

for the spatio-temporal domain \( \mathbb{T} \times \Omega = [0, 0.5] \times (0, 1) \) with analytical solution

\[
u(t, x) = \sin(2\pi x)e^{-(1/5)(2\pi)^2t}; \text{ see [35].}
\]

1) Semigroup Properties: The linear operator under consideration is defined by

\[
A: D \left( A \right) = H^2 \left( \Omega \right) \rightarrow L^2 \left( \Omega \right), \quad u \mapsto \frac{1}{5} \partial_{xx} u.
\]

The operator \( A \) generates a 0-contraction semigroup on \( L^2(\Omega) \) [22, Ch. 7.4.3, Th. 5], which is also exponentially stable on \( L^2(\Omega) \) using the results from [36].

Proposition 2: The operator \( A \equiv (1/5) \partial_{xx} \) generates an exponentially stable semigroup on \( L^2(\Omega) \) for \( \Omega = (0, 1) \) with decay parameter \( \mu = (1/5)^{\frac{1}{2}} \).

Proof: This is an application of [36, Th. 11.2.1] to the present problem.
with hard boundary constraints. The evaluation domain is larger than the training domain which is given by \( T = [0, 0.5] \). The a posteriori error estimator \( \varepsilon_{eq} + \varepsilon_{\text{init}} \) is depicted with the reference error \( \varepsilon_{\text{ref}} \). The error estimator for the semigroup of the heat equation as exponentially decaying semigroup is shown in dark blue denoted by superscript exp, while the results for the contractive semigroup are shown in red marked with the superscript con.

2) Setup: For this problem, we use an NN with four hidden layers of ten neurons each and train it for 3000 epochs. The PDE is enforced by 1000 collocation points in the domain \( T \times \Omega \) and the initial condition is trained with 200 data points.

We perform two experiments, one with hard boundary constraints using Theorem 1 and one with soft boundary constraints using Theorem 2. In the former case, we design the network with a 1-D mask to enforce the Dirichlet boundary condition parameterized by \( w_1 = 1 \) and \( x_1 = (1/2) \) (cf. Section IV). In the latter case, we train the network to match the boundary condition using the loss contribution

\[
L_{\text{space}} = \frac{1}{N_{\text{space}}} \sum_{i=1}^{N_{\text{space}}} (|u(t_i, x = 0)|^2 + |u(t_i, x = 1)|^2)
\]

with fixed weighting \( \rho = 10 \).

3) Numerical Results for Hard Boundary Constraints: The trained network agrees well with the known reference solution, with fixed weighting \( \rho = 10 \).

4) Numerical Results for Soft Boundary Constraints: Instead of training the heat equation with hard boundary constraints, we now use the extended theory with ISS as formulated in Theorem 2 to demonstrate that the error bound including ISS statements is valid for the heat equation. The error prediction is shown in Fig. 4. It is visible that the rise in the true error \( \varepsilon_{\text{ref}} \) toward the end of the time interval \([0, 0.5]\) is related to an increased error in the boundary condition. This is properly reflected by the contribution \( \varepsilon_{\text{bc}} \), which is computed using ISS parameters as analyzed in [25].

B. Homogeneous Advection Equation

We consider the homogeneous advection equation in two space dimensions with one output value

\[
\partial_t u(t, x) = \left( \frac{1}{5}, \frac{1}{2} \right) \cdot \nabla u(t, x) \quad (t, x) \in T \times \Omega
\]

Fig. 3. Posteriori error estimator for the heat equation (18) on \( \tilde{T} = [0, 1] \) with hard boundary constraints. The evaluation domain is larger than the training domain which is given by \( T = [0, 0.5] \). The a posteriori error estimator \( \varepsilon_{eq} + \varepsilon_{\text{init}} \) is depicted with the reference error \( \varepsilon_{\text{ref}} \).

Fig. 4. Heat equation with soft boundary constraints: prediction error estimation including additional error introduced by the boundary error via ISS properties.

\[
u(0, x) = \begin{cases} 
\frac{1}{2} - \|x\|_1 & , \text{for } \|x\|_1 \leq \frac{1}{2} \\
0 & , \text{else}
\end{cases}
\]

\[u(t, -2, x_2) = u(t, 2, x_2) \quad t \in T, \quad x_2 \in [-2, 2]
\]

\[u(t, x_1, -2) = u(t, x_1, 2) \quad t \in T, \quad x_1 \in [-2, 2]
\]

for \( \Omega := (-2, 2)^2 \) and \( \tilde{T} := [0, 8] \).

1) Semigroup Properties: To account for the periodic boundary constraints, we introduce the space

\[
X := \left\{ f \in L^2_{\text{loc}}(\mathbb{R}^2) \mid f(x) = f \left( x + \sum_{i=1}^2 k_i e_i \right) \right\}
\]

for \( k_i, e_i \in \mathbb{Z} \), i.e., \( x \in \mathbb{R}^2 \). The operator

\[
A : D(A) \to L^2(\Omega), \quad u \mapsto \left( \frac{1}{5}, \frac{1}{2} \right) \cdot \nabla u.
\]

is bounded on \( D(A) = H^1(\Omega) \cap X \) and generates a strongly continuous semigroup on \( X \). The solution to the advection equation is defined by the shift-semigroup \( S_{\text{shift}, C}(s)f(x) = f(x+sC) \) with \( C = ((1/5), (1/2))^T \), which generates an isometry on \( D(A) \subseteq L^2_{\text{loc}}(\mathbb{R}^2) \) [37, Ch. 1, Th. 4.15]. In particular, \( S_{\text{shift}, C} \) is a 0-contraction semigroup.

2) Setup: We set up an NN with eight hidden layers of 40 neurons each and train it for 10000 epochs with \( \kappa = 0.3 \). For the physics-informed contribution, 10000 collocation points in the domain \( \tilde{T} \times \Omega \), with \( \tilde{T} := [0, 4] \), are used. The initial condition is learned on training points located on a grid of \( 201 \times 201 \) in \( \Omega \) for \( t = 0 \). The periodic boundary condition are enforced according to Section IV.

3) Numerical Results: In Fig. 5, it is clearly visible that the prediction of the error lies in the same magnitude as the actual error and is mostly dominated by the error on the initial condition for the trained time range \( \tilde{T} = [0, 4] \). Even for an extended time domain \( T = [0, 8] \), for which the network has not been trained, the error estimator is a true upper bound on the prediction error. Since the estimator does not depend on NNs as a method or the quality of the approximation, this is an expected result.

The true error is overestimated increasingly over time but remains at the same magnitude of \( 10^3 \). The lower part of Fig. 5 illustrates that leaving the training domain increases both the error and the overestimation of the error.
C. Navier–Stokes Equations

The NSEs are one prominent example of numerical challenges and their solution via PINNs has been investigated previously in [38], [39], and [40]. We consider velocity–pressure formulation of the NSEs

\[ \begin{aligned}
    \partial_t u + \nabla p + \frac{1}{\text{Re}} \nabla^2 u &= 0 \quad \text{in } T \times \Omega \\
    \nabla \cdot u &= 0 \quad \text{in } T \times \Omega
\end{aligned} \]

to address a problem for which the analytical solution is known, the Taylor flow [41]. For the Taylor flow, we consider a 2-D domain \( \Omega := (0, \pi)^2 \) with associated velocities \( u = (u, v)^T \), pressure \( p \), \( \text{Re} = 1 \), and \( \varrho = 1 \) on the time domain \( T := [0, 1] \). The analytical solution is then given by [41]

\[ \begin{aligned}
    u(t, x_1, x_2) &= -\cos(x_1) \sin(x_2) e^{-2t} \\
    v(t, x_1, x_2) &= \sin(x_1) \cos(x_2) e^{-2t} \\
    p(t, x_1, x_2) &= -\frac{1}{4} (\cos(2x_1) + \cos(2x_2)) e^{-4t}.
\end{aligned} \]

1) Semigroup Properties: The semigroup properties for the NSEs can be derived by applying the Helmholtz–Leray projection [42] on (19) and investigating the resulting abstract Cauchy problem. The Helmholtz–Leray projection \( \mathcal{P} \) is the projection from \( L^2(\Omega) \) into the space of divergence-free vector fields \( L^2_0(\Omega) := \mathcal{P} L^2(\Omega) \). Using the Stokes operator

\[ \mathcal{A} = \mathcal{P} \Delta, \quad D(\mathcal{A}) = H^2(\Omega; \mathbb{R}^2) \cap L^2_0(\Omega; \mathbb{R}^2) \]

the projected NSEs are

\[ \hat{u}(t) = \mathcal{A} u(t) - \mathcal{P} (u \cdot \nabla) u \]

wherein the rightmost term is considered an inhomogeneity. The Stokes operator generates a bounded analytic semigroup and hence a strongly continuous semigroup in \( L^2(\Omega, \mathbb{R}^2) \) with \( \Omega \subset \mathbb{R}^2 \) bounded with a smooth boundary and Dirichlet boundary condition [43]. In consequence of [27, Th. 7.7] and [43], the operator \( \mathcal{P} \Delta \) generates a 0-contraction semigroup on \( L^2_0(\Omega) \).

2) Setup: We train a PINN with the loss function (14), \( \kappa = 0.7 \), \( \rho = 100 \), and

\[ L_{\text{space}} = \sum_{i=1}^{N_{\text{eq}}} \| \nabla \cdot u \|^2. \]

Since the PINN does not automatically satisfy the restriction that the target function must lie in the space of divergence-free functions, we hereby achieve that the violations made by our NN approximation are negligible in comparison to the other approximation errors. Due to this modification in the strategy, our estimator is no longer guaranteed to be a rigorous upper bound. We use hard boundary conditions as discussed in Section IV-A with centers \( x_{1,0} = x_{2,0} = \pi/2 \) and widths \( w_1 = w_2 = \pi \). The contribution \( \chi(t, x) \) contains the target boundary condition. Its components are given by

\[ \chi_{1,2}(t, x) = \left( -\frac{x_{1,2}^2}{\pi^2} + 1 \right) \left( \mp \sin(x_{2,1}) e^{-2t} \right) \]

\[ + \left( -\frac{(x_{1,2} - \pi)^2}{\pi^2} + 1 \right) \left( \pm \sin(x_{2,1}) e^{-2t} \right). \]

Here, we want to emphasize that the pressure \( p \) (or, to be more precise, its spatial derivative) and the nonlinear terms are considered only as an additional contribution to the residual (cf., Remark 1). Especially the pressure contribution to the residual is justified since it is not governed by a distinct evolution equation. For the sake of completeness, it is included in the initial condition, but the pressure is not restricted by any boundary condition.

Remark 6: Using the methodology as proposed in [44], one could enforce divergence-free fields in NNs by construction. Since this method relies on Fourier transformation and, therefore, would introduce an extensive bias to a sin/cos-based solution as desired, we decide to approximate the divergence-freeness as a soft constraint.

We train a PINN with ten hidden layers with 80 neurons in each layer over 20000 epochs with 2000 collocation points. The initial condition is trained on a grid of 31 \( \times \) 31 points over the spatial domain.

3) Numerical Results: The trained network agrees well with the initial condition deviating by \( \xi_0 = 1.2 \cdot 10^{-3} \), and the temporal evolution is also reflected well such that \( \xi = 5 \cdot 10^{-3} \). As visible in Fig. 6, the actual error decays slightly over time, which is not reflected by the error estimator. Here, the error estimator overestimates the actual error by approximately two magnitudes.

D. Klein–Gordon Equation With Homogeneous Neumann Boundary

As a final example, we consider the Klein–Gordon equation with Neumann boundary condition, which is relevant to problems in quantum field theory [45]. The 1-D problem can be formulated as

\[ \partial_{tt} u - \partial_{xx} u - \frac{1}{4} u \]

in the domain \( T \times \Omega = [0, 0.2] \times (0, 1) \). The initial values are given as \( u(0, x) = \cos(2\pi x) \), \( \partial_t u(0, x) = (1/2) \cos(4\pi x) \),
C0-cosine operator function on \( L^2 \) [47, Lemma 4.48, Remark 4.49], the operator equation can be written as a first-order system of the form for a bounded linear operator 
\( \partial_t u(t, x) = \frac{1}{4} \Delta_x - a_4 \cos(4\pi x) \sin(a_4 t) \)
with \( a_4 = (n\pi)^2 + (1/4)^{1/2} \); cf. [45].

1) Semigroup Properties: The BIVP for the Klein–Gordon equation can be written as a first-order system of the form
\[
\partial_t \begin{pmatrix} u \\ u_t \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ T \lambda^2 - \frac{1}{4} & 0 \end{pmatrix} \begin{pmatrix} u \\ u_t \end{pmatrix}.
\]

According to [46], the Neumann Laplacian generates a C0-cosine operator function on \( L^2(\Omega) \) and applying results [47, Lemma 4.48, Remark 4.49], the operator
\[
\hat{A} = \begin{pmatrix} 0 & T \\ A + B & 0 \end{pmatrix}
\]
for a bounded linear operator \( B: D(A) \rightarrow L^2(\Omega) \) generates a strongly continuous semigroup.

Since \( B = -(1/4)T \) with \( T \) denoting the identity operator, \( B \) suffices the conditions for [47, Remark 4.49]. Hence, the operator \( \hat{A} \) generates a strongly continuous semigroup on \( X = L^2(\Omega) \times L^2(\Omega) \) for the domain \( D(\hat{A}) = H^2(\Omega) \times L^2(\Omega) \).

We use the results [47, Remark 4.49] to derive the necessary parameters for error estimation. According to these results, the resolvent set of the new operator \( \rho(\hat{A}) \) is given by all \( \lambda \in \mathbb{C} \) whenever \( \lambda^2 \in \rho(A + B) \). This then transfers to the spectrum, such that \( \lambda \in \sigma(\hat{A}) \) whenever \( \lambda^2 \in \sigma(A + B) \).

The spectrum of the Neumann Laplacian consists of the point spectrum only and is given by \( \sigma(\Delta^N_{[0,1]}) = \{-k\pi \mid k \in \mathbb{N}_0\} \); so that the spectrum of \( \hat{A} \) can be derived to be \( \sigma(\hat{A}) = \{\{-k\pi - 0.25\}^{1/2} \mid k \in \mathbb{N}_0\} \). Hence, the largest real part of the eigenvalues is given by \( \lambda_{\text{max}} = 0 \) for \( k = 0 \), which we use to apply Corollary 1 using the spectral mapping theorem [27, Th. 2.4].

If we consider the Klein–Gordon equation on the more restrictive space \( X = H^2(\Omega) \times L^2(\Omega) \) with norm
\[
\| (u, v) \| = \left( \| \nabla u \|^2_{L^2(\Omega)} + 0.25 \| u \|^2_{L^2(\Omega)} + \| v \|^2_{L^2(\Omega)} \right)^{1/2}
\]
then one can show that the generated semigroup is a 0-contraction semigroup analogous to the proof of [22, Ch. 7.4, Th. 5] with minor modifications.

2) Numerical Determination of \( M \): This Klein–Gordon example on \( X = L^2(\Omega) \times L^2(\Omega) \) shows nicely the technical difficulties arising when the parameter \( M \) is required for a complete understanding of the growth behavior of the weak solution. To retrieve a reasonable estimate for \( M \), we use the inequality \( M \geq (\|S(t)v\|/\|e^0\| \|v\|) \). Here, we use input data \( v \), which is Gaussian noise with means in the range \([0, 20] \times [0, 20]\) and variances in the range \([0.05, 0.5] \times [0.05, 0.5]\), and compute \( \|S(t)v\| \) with the python framework FiPy [48]. Thereby we approximate \( M \geq 164.43 \).

3) Setup: We use an NN with six hidden layers with ten neurons each. The initial data are given by 201 equally spaced data points in the range \([0, 1]\) and 2500 collocation points are selected equally distributed from \( \mathbb{T} \times [0, 1] \). The Neumann boundary condition is implemented as soft boundary constraint via
\[
L_{\text{space}} = \frac{1}{N_{\text{eq}}} \sum_{i=1}^{N_{\text{eq}}} \left( |u_x(t_{\text{eq},i}, 0)|^2 + |u_x(t_{\text{eq},i}, 1)|^2 \right)
\]
with \( \rho = 50 \). The data-driven loss is more heavily considered by choosing \( \kappa = 2.5 \). We use multiple optimization algorithms to train this NN properly. First, we run 30,000 epochs of the Adam optimizer with learning rate \( 10^{-3} \) to find a suitable starting point for further 50,000 epochs of the second-order L-BFGS algorithm with learning rate \( 10^{-2} \).

4) Numerical Results: The error estimator for the Klein–Gordon equation (19) on \( X = L^2(\Omega) \times L^2(\Omega) \) overestimates the true error from the beginning significantly by two orders of magnitude (cf., Fig. 7, top). This improves slightly over time to only one order of magnitude, but is most clearly due to the high value of \( M \). The fact that \( M \neq 1 \) is reflected nicely in Fig. 7 (top), in which the error contribution \( e_{\text{init}} \) is significantly larger than the true error \( e_{\text{ref}} \) even for \( t = 0 \).

When evaluating the NN on \( X = H^2(\Omega) \times L^2(\Omega) \) with the norm (20) we are considering a 0-contraction semigroup, which is properly reflected in Fig. 7 (bottom). In consequence, the error estimator lies in the same order of magnitude as the true error and overestimates the error by less than 10%.

E. Efficiency of the Error Estimator

The previous numerical examples illustrate some aspects influencing the efficiency of the error estimator as discussed in Section III-C. In particular, the transport equation (cf., Section V-B) demonstrates a close agreement of the error estimator with the reference error.

The effect of the growth bound of the semigroup of operators can be observed particularly well for the Klein–Gordon equation (cf., Section V-D). Therein, it becomes evident that choosing spaces and norms for which \( M \gg 1 \) deteriorates the efficiency of the error estimator significantly.

For the heat equation with soft boundary constraints (cf., Section V-A), the contribution \( \epsilon_{\text{hbc}} \) dominates the error estimator, causing it to overshoot the reference error by a magnitude. This indicates that the ISS gain \( \gamma \) worsens the efficiency of the error estimator.
We derived computable rigorous upper bounds for the prediction error of PINNs and other surrogate modeling techniques that approximate solutions to linear PDEs. We illustrated the applicability of the error estimator in several examples. We discuss in the following the most important of the thereby illustrated properties, possibilities, and shortcomings.

We demonstrated with the example of the NSEs that the error estimator can also be applied to nonlinear PDEs. Two points are worth discussing concerning this example: First, the chosen fluid-dynamics problem is a very simple case and there is a gap to real-world fluid-dynamics challenges. Second, as the taken approach presumes that the dominant contribution to the time evolution is still linear and leads to a weakening of the guaranteed upper bound to an approximate error estimation. In conclusion, the previously presented methodology is not directly applicable to nonlinear PDEs, which is subject to further investigation.

Apart from the question of how to thoroughly extend the presented theory to nonlinear PDEs, finding the correct governing semigroup parameters remains a challenge. Even though most real-world problems (such as the NSEs, Maxwell equations) probably have been discussed extensively as part of many mathematical research areas, the investigations of the parameters necessary for the a posteriori error estimator have been more academic in nature. More precisely, while the existence of these parameters has been proven in various constellations, their explicit calculation is either not directly included in the publications or hidden in lengthy proofs. We have presented for the Klein–Gordon equation how to determine numerically one of the key parameters. This approach could be extended to compute both the growth bound $\omega$ and the scaling factor $M$ by discretizing the problem using finite-element methods and finding the eigenvalues. This problem in general and the latter suggestion need further study.

We have shown for the heat equation that the two different estimators resulting once from the analysis that it is a 0-contractive semigroup and once from the exponential stability analysis, overestimate the true error in different magnitudes. This improvement of the estimator through a more thorough analysis of the semigroup is consistent with the fundamental reason why PINNs are more powerful than purely data-driven NNs: the introduction of a priori knowledge.

Mostly, except for the heat equation and the Klein–Gordon equation, we used hard boundary constraints. This is rarely possible, and when it is, it can be laborious to find a suitable representation. Hence, the extension of the theory using input-to-state stability (ISS) is highly relevant for the practical use of this estimator. Since ISS is restricted to exponentially stable semigroups, a similar analysis for, e.g., the Klein–Gordon equation remains open.

The methodology has not yet been extended by strategies to leverage the error to judge training success or even to design adaptive training methods. Beyond the previously mentioned technical limitations and challenges, it does not provide information about a lower limit to the true error, as it would be required for judging the efficiency of the error estimator [49].

VI. CONCLUSION

We derived computable rigorous upper bounds for the prediction error of PINNs and other surrogate modeling techniques that approximate solutions to linear PDEs. We illustrated the applicability of the error estimator in several examples. We discuss in the following the most important of the thereby illustrated properties, possibilities, and shortcomings.

We demonstrated with the example of the NSEs that the error estimator can also be applied to nonlinear PDEs. Two points are worth discussing concerning this example: First, the chosen fluid-dynamics problem is a very simple case and there is a gap to real-world fluid-dynamics challenges. Second, as the taken approach presumes that the dominant contribution to the time evolution is still linear and leads to a weakening of the guaranteed upper bound to an approximate error estimation. In conclusion, the previously presented methodology is not directly applicable to nonlinear PDEs, which is subject to further investigation.

Apart from the question of how to thoroughly extend the presented theory to nonlinear PDEs, finding the correct governing semigroup parameters remains a challenge. Even though most real-world problems (such as the NSEs, Maxwell equations) probably have been discussed extensively as part of many mathematical research areas, the investigations of the parameters necessary for the a posteriori error estimator have been more academic in nature. More precisely, while the existence of these parameters has been proven in various constellations, their explicit calculation is either not directly included in the publications or hidden in lengthy proofs. We have presented for the Klein–Gordon equation how to determine numerically one of the key parameters. This approach could be extended to compute both the growth bound $\omega$ and the scaling factor $M$ by discretizing the problem using finite-element methods and finding the eigenvalues. This problem in general and the latter suggestion need further study.

We have shown for the heat equation that the two different estimators resulting once from the analysis that it is a 0-contractive semigroup and once from the exponential stability analysis, overestimate the true error in different magnitudes. This improvement of the estimator through a more thorough analysis of the semigroup is consistent with the fundamental reason why PINNs are more powerful than purely data-driven NNs: the introduction of a priori knowledge.

Mostly, except for the heat equation and the Klein–Gordon equation, we used hard boundary constraints. This is rarely possible, and when it is, it can be laborious to find a suitable representation. Hence, the extension of the theory using input-to-state stability (ISS) is highly relevant for the practical use of this estimator. Since ISS is restricted to exponentially stable semigroups, a similar analysis for, e.g., the Klein–Gordon equation remains open.

The methodology has not yet been extended by strategies to leverage the error to judge training success or even to design adaptive training methods. Beyond the previously mentioned technical limitations and challenges, it does not provide information about a lower limit to the true error, as it would be required for judging the efficiency of the error estimator [49].
Q. Hong, J. W. Siegel, and J. Xu, “A priori analysis of stable neural network solutions to numerical PDEs,” 2021, arXiv:2104.02903.

T. D. Ryck and S. Mishra, “Generic bounds on the approximation error for physics-informed (and) operator learning,” in Proc. Adv. Neural Inf. Process. Syst., 2022, pp. 1–14.

G. Singh, T. Gehr, M. Mirman, M. Pischel, and M. Vechev, “Fast and effective robustness certification,” in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018.

P. Minakowski and T. Richter, “Error estimates for neural network solutions of partial differential equations,” 2021, arXiv:2107.11035.

S. Berrone, C. Canuto, and M. Pintore, “Solving PDEs by variational physics-informed neural networks: An a posteriori error analysis,” Annali Dell’Università di Ferrara, vol. 68, no. 2, pp. 575–595, Nov. 2022.

E. D. Sontag, “Smooth stabilization implies co-prime factorization,” IEEE Trans. Autom. Control, vol. 34, no. 4, pp. 435–443, Apr. 1989.

Z. Ding, “A proof of the trace theorem of Sobolev spaces on Lipschitz domains,” Proc. Amer. Math. Soc., vol. 124, no. 2, pp. 591–600, 1996.

L. C. Evans, Partial Differential Equations, vol. 19. Providence, RI, USA: American Mathematical Society, 2010.

F. L. Schwenger, “Input-to-state stability for parabolic boundary control: Linear and semilinear systems,” in Control Theory of Infinite-Dimensional Systems. New York, NY, USA: Springer-Verlag, 2020, pp. 83–116.

M. Tucsnak and G. Weiss, Observation and Control for Operator Semigroups. New York, NY, USA: Springer-Verlag, 2009.

B. Jacob, R. Nabiullin, J. R. Partington, and F. L. Schwenger, “Infinite-dimensional input-to-state stability and Orlicz spaces,” SIAM J. Control Optim., vol. 56, no. 2, pp. 868–889, Jan. 2018.

J. Yu, L. Lu, X. Meng, and G. E. Karniadakis, “Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems,” Comput. Methods Appl. Mech. Eng., vol. 393, Apr. 2022, Art. no. 114823.

A. Pazy, Semigroups of Linear Operators and Applications to Partial Differential Equations, vol. 44. New York, NY, USA: Springer-Verlag, 1989.

K.-I. Mitani, K.-S. Saito, M. Kato, and T. Tamura, “On sharp triangle inequalities in Banach spaces,” J. Math. Anal. Appl., vol. 336, no. 2, pp. 1178–1186, Dec. 2007.

K. S. McFall, “An artificial neural network method for solving boundary value problems with arbitrary irregular boundaries,” M.S. thesis, Georgia Inst. Technol., Atlanta, Georgia, 2006.

K. S. McFall and J. R. Mahan, “Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions,” IEEE Trans. Neural Netw., vol. 20, no. 8, pp. 1221–1233, Aug. 2009.

S. Dong and N. Ni, “A method for representing periodic functions and enforcing exactly periodic boundary conditions with deep neural networks,” J. Comput. Phys., vol. 435, Jun. 2021, Art. no. 110242.

M. Abadi et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. [Online]. Available: http://download.tensorflow.org/paper/whitepaper2015.pdf

D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” Math. Program., vol. 45, nos. 1–3, pp. 503–528, Aug. 1989.

M. D. McKay, R. J. Beckman, and W. J. Conover, “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” Technometrics, vol. 21, no. 2, p. 239, May 1979.

J. Fourier, The Analytical Theory of Heat. Cambridge, U.K.: Cambridge Univ. Press, 1878.

C. Seifer, T. Turoff, and M. Waurick, “Exponential stability of evolutionary equations,” in Evolutionary Equations. New York, NY, USA: Springer-Verlag, 2022.

K.-J. Engel, R. Nagel, and S. Brendle, One-Parameter Semigroups for Linear Evolution Equations, vol. 194. New York, NY, USA: Springer-Verlag, 2000.

X. Jin, S. Cai, H. Li, and G. E. Karniadakis, “NSFinets (Navier–Stokes flow nets): Physics-informed neural networks for the incompressible Navier–Stokes equations,” J. Comput. Phys., vol. 426, Feb. 2021, Art. no. 109951.

C. Amalainadi, P. S. Palar, R. Stevenson, and L. Zuluah, “On physics-informed deep learning for solving Navier–Stokes equations,” in Proc. AIAA SCITECH Forum, Jan. 2022, p. 1436.

Birgit Hillebrecht received the B.Sc. degree in mathematics and the B.Sc. degree in physics from the University of Aachen, Aachen, Germany, in 2015, the M.Sc. degree in physics from Technical University Munich, Munich, Germany, in 2018, and the M.Sc. degree in mathematics from the University of Hagen, Hagen, Germany, in 2021. She is currently pursuing the Ph.D. degree in mathematics and simulation science with the University of Stuttgart, Stuttgart, Germany.

She joined Vector Informatik GmbH as a Software Development Engineer in 2018 and is currently a Senior Solution Manager with the Embedded Systems Department. Her research interests include dynamical systems, partial differential equations and neural networks, specifically the intersection between the previous mentioned topics.

Benjamin Unger received the Ph.D. degree in mathematics from the Technische Universität Berlin, Berlin, Germany, in 2020.

He is an Independent Junior Research Group Leader for dynamical systems with the University of Stuttgart, Stuttgart, Germany, within the cluster of excellence “Data-integrated simulation science (SimTech)” and a Faculty Member of the International Max Planck Research School for Intelligent Systems. His research interests include surrogate modeling, port-Hamiltonian systems, and delay differential-algebraic equations.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.