ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Error analysis for physics informed neural networks (PINNs) approximating Kolmogorov PDEs

T. De Ryck and S. Mishra

Research Report No. 2021-17
June 2021

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

ERROR ANALYSIS FOR PHYSICS INFORMED NEURAL NETWORKS (PINNS) APPROXIMATING KOLMOGOROV PDES

T. DE RYCK AND S. MISHRA

Abstract. Physics informed neural networks approximate solutions of PDEs by minimizing pointwise residuals. We derive rigorous bounds on the error, incurred by PINNs in approximating the solutions of a large class of linear parabolic PDEs, namely Kolmogorov equations that include the heat equation and Black-Scholes equation of option pricing, as examples. We construct neural networks, whose PINN residual (generalization error) can be made as small as desired. We also prove that the total L^2-error can be bounded by the generalization error, which in turn is bounded in terms of the training error, provided that a sufficient number of randomly chosen training (collocation) points is used. Moreover, we prove that the size of the PINNs and the number of training samples only grow polynomially with the underlying dimension, enabling PINNs to overcome the curse of dimensionality in this context. These results enable us to provide a comprehensive error analysis for PINNs in approximating Kolmogorov PDEs.

1. Introduction

Background and context. Partial differential equations (PDEs) are ubiquitous as mathematical models in the sciences and engineering. Explicit solution formulas for PDEs are not available except in very rare cases. Hence, numerical methods, such as finite difference, finite element and finite volume methods, are key tools in approximating solutions of PDEs. In spite of their well-documented successes, it is clear that these methods are inadequate for a variety of problems involving PDEs. In particular, these methods are not suitable for efficiently approximating PDEs with *high-dimensional* state or parameter spaces. Such problems arise in different contexts ranging from PDEs such as the Boltzmann, Radiative transfer, Schrödinger and Black-Scholes type equations with very high number of spatial dimensions, to *many-query* problems, as in uncertainty quantification (UQ), optimal design and inverse problems, which are modelled by PDEs with very high parametric dimensions.

Given this pressing need for efficient algorithms to approximate the afore-mentioned problems, machine learning methods are being increasingly deployed in the context of scientific computing. In particular, deep neural networks (DNNs) i.e., multiple compositions of affine functions and scalar nonlinearities, are being widely used. Given the *universality* of DNNs in being able to approximate any continuous (measurable) function to desired accuracy, they can serve as ansatz spaces for solutions of PDEs, as for high-dimensional semi-linear parabolic PDEs [7], linear elliptic PDEs [35, 15] and nonlinear hyperbolic PDEs [23, 24] and references therein. More recently, DNN-inspired architectures such as DeepOnets [4, 21, 18] and Fourier Neural operators [20] have been shown to even learn infinite-dimensional *operators*, associated with underlying PDEs, efficiently.

A large part of the literature on the use of deep learning for approximating PDEs relies on the *supervised learning* paradigm, where the DNN has to be *trained* on possibly large amounts of labelled data. However in practice, such data is acquired from either measurements or computer simulations. Such simulations might be very computationally expensive [23] or even infeasible in many contexts, impeding the efficiency of the supervised learning algorithms. Hence, it would be very desirable to find a class of machine learning algorithms that can approximate PDEs, either without any explicit need for data or with very small amounts of data. Physics informed neural networks (PINNs) provide exactly such a framework.

Physics Informed Neural Networks (PINNs). PINNs were first proposed in the 90s [6, 17, 16] as a machine learning framework for approximating solutions of differential equations. However, they were resurrected recently in [32, 33] as a practical and computationally efficient paradigm for solving both forward and inverse problems for PDEs. Since then, there has been an explosive growth in designing

(T. De Ryck and S. Mishra) Seminar for Applied Mathematics, ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland

E-mail addresses: tim.deryck@sam.math.ethz.ch, siddhartha.mishra@sam.math.ethz.ch.

and applying PINNs for a variety of applications involving PDEs. A very incomplete list of references includes [34, 22, 25, 31, 39, 12, 13, 26, 27, 28, 1] and references therein.

We briefly illustrate the idea behind PINNs by considering the following general form of a PDE:

$$\mathcal{D}[u](x,t) = 0, \quad \mathcal{B}u(y,t) = \psi(y,t), \quad u(x,0) = \varphi(x), \quad \text{for } x \in D, y \in \partial D, t \in [0,T], \tag{1.1}$$

Here, $D \subset \mathbb{R}^d$ is compact and $\mathcal{D}, \mathcal{B}$ are the differential and boundary operators, $u : D \times [0,T] \to \mathbb{R}^m$ is the solution of the PDE, $\psi : \partial D \times [0,T] \to \mathbb{R}^m$ specifies the (spatial) boundary condition and $\varphi : D \to \mathbb{R}^m$ is the initial condition.

We seek deep neural networks $u_\theta : D \times [0,T] \to \mathbb{R}^m$ (see (2.6) for a definition), parameterized by $\theta \in \Theta$, constituting the weights and biases, that approximate the solution u of (1.1). To this end, the key idea behind PINNs is to consider pointwise *residuals*, defined for any sufficiently smooth function $f : D \times [0,T] \to \mathbb{R}^m$ as,

$$\mathcal{R}_i[f](x,t) = \mathcal{D}[f](x,t), \quad \mathcal{R}_s[f](t,y) = \mathcal{B}f(t,y) - \psi(t,y), \quad \mathcal{R}_t[f](x) = f(0,x) - \varphi(x) \tag{1.2}$$

for $x \in D$, $y \in \partial D$, $t \in [0,T]$. Using these residuals, one measures how well a function f satisfies resp. the PDE, the boundary condition and the initial condition of (1.1). Note that for the exact solution $\mathcal{R}_i[u] = \mathcal{R}_s[u] = \mathcal{R}_t[u] = 0$.

Hence, within the PINNs algorithm, one seeks to find a neural network u_θ, for which all residuals are simultaneously minimized, e.g. by minimizing the quantity,

$$\mathcal{E}_G(\theta)^2 = \int_{D\times[0,T]} \left|\mathcal{R}_i[u_\theta](x,t)\right|^2 dxdt + \int_{\partial D\times[0,T]} \left|\mathcal{R}_s[u_\theta](x,t)\right|^2 ds(x)dt + \int_D \left|\mathcal{R}_t[u_\theta](x)\right|^2 dx. \tag{1.3}$$

However, the quantity $\mathcal{E}_G(\theta)$, often referred to as the *population risk* or *generalization error* [26] of the neural network u_θ involves integrals and can therefore not be directly minimized in practice. Instead, the integrals in (1.3) are approximated by a numerical quadrature, resulting in,

$$\mathcal{E}_T^i(\theta, \mathcal{S}_i)^2 = \sum_{n=1}^{N_i} w_i^n \left|\mathcal{R}_i[u_\theta](t_i^n, x_i^n)\right|^2, \quad \mathcal{E}_T^s(\theta, \mathcal{S}_s)^2 = \sum_{n=1}^{N_s} w_s^n \left|\mathcal{R}_s[u_\theta](t_s^n, x_s^n)\right|^2, \quad \mathcal{E}_T^t(\theta, \mathcal{S}_t)^2 = \sum_{n=1}^{N_t} w_t^n \left|\mathcal{R}_t[u_\theta](x_i^t)\right|^2. \tag{1.4}$$

Here, one samples quadrature points in space-time to construct data sets $\mathcal{S}_i = \{(t_i^n, x_i^n)\}_n^{N_i}$, $\mathcal{S}_s = \{(t_s^n, x_s^n)\}_n^{N_s}$ and $\mathcal{S}_t = \{x_t^n\}_n^{N_t}$, and w_q^n are suitable quadrature weights for $q = i, t, s$. Thus, the *generalization error* $\mathcal{E}_G(\theta)$ is approximated by the so-called *training loss* or *training error* [26],

$$\mathcal{E}_T(\theta, \mathcal{S})^2 = \mathcal{E}_T^i(\theta, \mathcal{S}_i)^2 + \mathcal{E}_T^s(\theta, \mathcal{S}_s)^2 + \mathcal{E}_T^t(\theta, \mathcal{S}_t)^2, \tag{1.5}$$

where $\mathcal{S} = (\mathcal{S}_i, \mathcal{S}_s, \mathcal{S}_t)$, and a stochastic gradient descent algorithm is to used to approximate the non-convex optimization problem,

$$\theta^* = \arg\min_{\theta\in\Theta} \mathcal{E}_T(\theta, \mathcal{S})^2, \tag{1.6}$$

and $u^* = u_{\theta^*}$ is the trained PINN that approximates the solution u of the PDE (1.1).

Theory for PINNs. Given this succinct description of the PINNs algorithm, the following fundamental theoretical questions arise immediately,

Q1. Given a tolerance $\varepsilon > 0$, does there exist a PINN $\hat{u} = u_{\hat{\theta}}$, parametrized by a $\hat{\theta} \in \Theta$ such that the corresponding generalization error (population risk) $\mathcal{E}_G(\hat{\theta})$ (1.3) is small i.e., $\mathcal{E}_G(\hat{\theta}) < \varepsilon$?

Q2. Given a PINN $\hat{u}$ with small generalization error, is the corresponding *total error* $\|u - \hat{u}\|$ small, i.e., is $\|u - \hat{u}\| < \delta(\varepsilon)$, for some $\delta(\varepsilon) \sim \mathcal{O}(\varepsilon)$, for some suitable norm $\|.\|$, and with u being the solution of the PDE (1.1)?

The above questions are of fundamental importance as affirmative answers to them certify that, *in principle*, there exists a PINN, corresponding to the parameter $\hat{\theta}$, such that the resulting PDE residual (1.2) is small, and consequently also the overall error in approximating the solution of the PDE (1.1).

Moreover, the smallness of the generalization error $\mathcal{E}_G(\hat{\theta})$ can imply that the training error $\mathcal{E}_T(\hat{\theta})$ (1.5), which is an approximation of the generalization error, is also small. Hence, *in principle*, the (global) minimization of the optimization problem (1.6) should result in a proportionately small training error.

However, the optimization problem (1.6) involves the minimization of a *non-convex*, very-high dimensional objective function. Hence, it is unclear if a global minimum is attained by a gradient-descent algorithm. *In practice*, one can evaluate the training error $\mathcal{E}_T(\theta^*)$ for the (local) minimizer θ^* of (1.6). Thus, it is natural to ask if, Q3. Given a small training error $\mathcal{E}_T(\theta^*)$ and a sufficiently large training set $\mathcal{S}$, is the corresponding generalization error $\mathcal{E}_G(\theta^*)$ also proportionately small?

An affirmative answer to question Q3, together with question Q2, will imply that the trained PINN u_{θ^*} is an accurate approximation of the solution u of the underlying PDE (1.1). Thus, answering the above three questions affirmatively will constitute a comprehensive theoretical investigation of PINNs and provide a rationale for their very successful empirical performance.

Given the very large number of papers exploring PINNs empirically, the rigorous theoretical study of PINNs is in a relative state of infancy. In [36], the authors prove a consistency result for PINNs, for linear elliptic and parabolic PDEs, where they show that if $\mathcal{E}_T(\theta_m) \to 0$ for a sequence of neural networks $\{u_{\theta_m}\}_{m\in\mathbb{N}}$, then $\|u_{\theta_m} - u\|_{L^\infty} \to 0$, under the assumption that one adds a specific $C^{k,\alpha}$-regularization term to the loss function, thus partially addressing question Q3 for these PDEs. However, this result does not provide quantitative estimates on the underlying errors. A similar result, with more quantitative estimates for advection equations is provided in [37].

In [26, 27], the authors provide a strategy for answering questions Q2 and Q3 above. They leverage the *stability* of solutions of the underlying PDE (1.1) to bound the total error in terms of the generalization error (question Q2). Similarly, they use accuracy of quadrature rules to bound the generalization error in terms of the training error (question Q3). This approach is implemented for Forward problem corresponding to a variety of PDEs such as the semi-linear and quasi-linear parabolic equations and the incompressible Euler (Navier-Stokes) equations [26], radiative transfer equations [28], nonlinear dispersive PDEs such as the KdV equations [1] and for the unique continuation (data assimilation) inverse problem for many linear elliptic, parabolic and hyperbolic PDEs [27]. However, these works suffer from two essential limitations: first, question Q1 on the smallness of generalization error is not addressed and second, the assumptions on the quadrature rules in [26, 27] are rather stringent and in particular, the analysis does not include the common choice of using random sampling points in $\mathcal{S}$, unless an additional validation set is chosen. Thus, the theoretical analysis presented in [26, 27] is incomplete and this sets the stage for the current paper.

Aims and scope of this paper. Given the above discussion, our main aims in this paper are to address the fundamental questions Q1, Q2 and Q3 and to establish a solid foundation and rigorous rationale for PINNs in approximating PDEs.

To this end, we choose to focus on a specific class of PDEs, the so-called Kolmogorov equations [30] in this paper. These equations are a class of *linear, parabolic* PDEs which describe the space-time evolution of the density for a large set of stochastic processes. Prototypical examples include the heat (diffusion) equation and Black-Scholes type PDEs that arise in option pricing. A key feature of Kolmogorov PDEs is the fact that the equations are set in very high dimensions. For instance, the spatial dimension in a Black-Scholes PDE is given by the number of underlying assets (stocks), upon which the basket option is contingent, and can range up to hundreds of dimensions.

Our motivation for illustrating our analysis on Kolmogorov PDEs is two-fold. First, they offer a large class of PDEs with many applications, while still being linear. Second, it has already been shown empirically in [26, 38, 29] that PINNs can approximate very high-dimensional Kolmogorov PDEs efficiently.

Thus in this paper,

- We show that there exist PINNs, approximating a class of Kolmogorov PDEs, such that the resulting generalization error (1.3), and the total error, can be made as small as possible. Moreover under suitable hypothesis on the initial data and the underlying exact solutions, we will show that the size of these PINNs does not grow exponentially with respect to the spatial dimension of the underlying PDE. This is done by explicitly constructing PINNs using a representation formula, the so-called Dynkin's formula, that relates the solutions of the Kolmogorov PDE to the generator and sample paths for the underlying stochastic process.
- We leverage the stability of Kolmogorov PDEs to bound the error, incurred by PINNs in L^2-norm in approximating solutions of Kolmogorov PDEs, by the underlying generalization error.
- We provide rigorous bounds for the generalization error of the PINN approximating Kolmogorov PDEs in terms of the underlying training error (1.5), provided that the number of *randomly* chosen training points is sufficiently large. Furthermore, the number of random training points does not grow exponentially with the dimension of the underlying PDE. We use a novel error decomposition and standard Hoeffding's inequality type covering number estimates to derive these bounds.

Thus, we provide affirmative answers to questions Q1, Q2 and Q3 for this large class of PDEs. Moreover, we also show that PINNs can *overcome the curse of dimensionality* in approximating these PDEs. Hence, our results will place PINNs for these PDEs on solid theoretical foundations.

The rest of the paper is organized as follows: In section 2, we present preliminary material on linear Kolmogorov equations and describe the PINNs algorithm to approximate them. The generalization error and total error (questions Q1 and Q2) are considered in section 3 and the generalization error is bounded in terms of training error (question Q3) in section 4.

2. PINNs for Linear Kolmogorov Equations

2.1. **Linear Kolmogorov PDEs.** In this paper, we will consider the following general form of linear time-dependent partial differential equations,

$$\begin{cases} u_t(t,x) = \frac{1}{2}\mathrm{Trace}(\sigma(x)\sigma(x)^T H_x[u](t,x)) + \mu(x)^T \cdot \nabla_x[u](t,x) & \text{for all } (t,x) \in [0,T] \times D, \\ u(0,x) = \varphi(x) & \text{for all } x \in D, \\ u(t,x) = \psi(x,t) & \text{for all } (t,x) \in [0,T] \times \partial D. \end{cases} \tag{2.1}$$

where $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ and $\mu : \mathbb{R}^d \to \mathbb{R}^d$ are affine functions, ∇_x denotes the gradient and H_x the Hessian (both with respect to the space coordinates). For definiteness, we set $D = (0,1)^d$. PDEs of the form (2.1) are referred to as Kolmogorov equations and arise in a large number of models in science and engineering. Prototypical examples of Kolmogorov PDEs include,

1. **Heat Equation**: Let $\mu = 0$ and $\sigma = \sqrt{\kappa} I_d$, where $\kappa > 0$ is the thermal diffusivity of the medium and I_d is the d-dimensional identity matrix. This results in the following PDE for the temperature u,

$$u_t(t,x) = \kappa \sum_{j=1}^{d} u_{x_j x_j}(t,x), \qquad u(0,x) = \varphi(x). \tag{2.2}$$

   Here, φ describes the initial heat distribution. Dirichlet or Neumann boundary data complete the problem.

2. **Black-Scholes equation**: If both μ and σ in (2.1) are linear functions, we obtain the Black-Scholes equation, which models the evolution in time t of the price of an option u that is based on d underlying stocks x_i. Up to a straightforward change of variables, the corresponding PDE is given by (see e.g. [30]),

$$u_t(t,x) = \sum_{i,j=1}^{d} \beta_i \beta_j \rho_{ij} x_i x_j u_{x_i x_j}(t,x) + \sum_{j=1}^{d} \mu x_j u_{x_j}(t,x), \qquad u(0,x) = \varphi(x). \tag{2.3}$$

   Here, the β_i are stock volatilities, the coefficients ρ_{ij} model the correlation between the different stock prices, μ is an interest rate and the initial condition φ is interpreted as a payoff function. Prototypical examples of such payoff functions are $\varphi(x) = \max\{\sum_i a_i x_i - K, 0\}$ (basket call option), $\varphi(x) = \max\{\max_i a_i x_i - K, 0\}$ (call on max) and analogously for put options.

Our goal in this paper is to approximate the classical solution u of Kolmogorov equations with PINNs. We start with a brief recapitulation of neural networks below.

2.2. **Neural Networks.** We denote by $\sigma : \mathbb{R} \to \mathbb{R}$ be an (at least) twice continuously differentiable activation function, like tanh or sigmoid. For any $n \in \mathbb{N}$, we write for $z \in \mathbb{R}^n$ that $\sigma(z) := (\sigma(z_1), \ldots, \sigma(z_n))$. We formally define a neural network below,

Definition 2.1. *Let $R \in (0,\infty]$, $L, W \in \mathbb{N}$ and $l_0, \ldots, l_L \in \mathbb{N}$. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a twice differentiable function and define*

$$\Theta = \Theta_{L,W,R} := \bigcup_{L' \in \mathbb{N}, L' \leq L} \; \bigcup_{l_0,\ldots,l_L \in \{1,\ldots,W\}} \; \bigtimes_{k=1}^{L'} \left([-R,R]^{l_k \times l_{k-1}} \times [-R,R]^{l_k} \right). \tag{2.4}$$

For $\theta \in \Theta_{L,W,R}$, we define $(W_k, b_k) := \theta_k$ and $\mathcal{A}_k^\theta : \mathbb{R}^{l_{k-1}} \to \mathbb{R}^{l_k} : z \mapsto W_k z + b_k$ for $1 \leq k \leq L$ and we define $f_k^\theta : \mathbb{R}^{l_{k-1}} \to \mathbb{R}^{l_k}$ by

$$f_k^\theta(z) = \begin{cases} \mathcal{A}_L^\theta(z) & k = L, \\ (\sigma \circ \mathcal{A}_k^\theta)(z) & 1 \leq k < L. \end{cases} \tag{2.5}$$

We denote by $u_\theta : \mathbb{R}^{l_0} \to \mathbb{R}^{l_L}$ the function that satisfies for all $z \in \mathbb{R}^{l_0}$ that

$$u_\theta(z) = \left(f_L^\theta \circ f_{L-1}^\theta \circ \cdots \circ f_1^\theta\right)(z), \tag{2.6}$$

where in the setting of approximating Kolmogorov PDEs (2.1) *we set $l_0 = d+1$ and $z = (x,t)$.*

We refer to u_θ as the realization of the neural network associated to the parameter θ with L layers with widths $(l_0, l_1, \dots, l_L)$, of which the middle $L-1$ layers are called hidden layers. For $1 \le k \le L$, we say that layer k has width l_k and we refer to W_k and b_k as the weights and biases corresponding to layer k. If $L \ge 3$, we say that u_θ is a deep neural network (DNN).

2.3. **PINNs.** As already mentioned in the introduction, the key idea behind PINNs is to minimize pointwise residuals associated with the Kolmogorov PDE (2.1). To this end, we define the differential operator associated with (2.1),

$$\mathcal{L}\left[v\right](x) = \sum_{i=1}^{d} \mu_i(x)(\partial_i v)(x,t) + \frac{1}{2}\sum_{i,j,k=1}^{d} \sigma_{ik}(x)\sigma_{kj}(x)(\partial_{ij}^2 v)(x), \tag{2.7}$$

for any $v \in C^2(\mathbb{R}^d)$. Next, we define the following residuals associated with (2.1),

$$\begin{aligned} \mathcal{R}_i[v](x,t) &= \partial_t v(x,t) - \mathcal{L}\left[v\right](x,t), \quad (x,t) \in D \times [0,T], \\ \mathcal{R}_s[v](y,t) &= v(y,t) - \psi(y,t), \quad (y,t) \in \partial D \times [0,T], \\ \mathcal{R}_t[v](x) &= v(0,x) - \varphi(x), \quad \forall x \in D. \end{aligned} \tag{2.8}$$

The *generalization error* for a neural network of the form (2.6), approximating the Kolmogorov PDE is then given by the formula (1.3), but with the residuals defined in (2.8).

Given the possibly very high-dimensional domain D of (2.1), it is natural to use random sampling points to define the loss function for PINNs $\theta \mapsto \mathcal{E}_T(\theta, \mathcal{S})^2$ as follows,

$$\begin{aligned} &\mathcal{E}_T^i(\theta, \mathcal{S}_i)^2 = \frac{1}{N_i}\sum_{n=1}^{N_i} \left|\mathcal{R}_i[u_\theta](t_i^n, x_i^n)\right|^2, \\ &\mathcal{E}_T^s(\theta, \mathcal{S}_s)^2 = \frac{1}{N_s}\sum_{n=1}^{N_s} \left|\mathcal{R}_s[u_\theta](t_s^n, x_s^n)\right|^2, \quad \mathcal{E}_T^t(\theta, \mathcal{S}_t)^2 = \frac{1}{N_t}\sum_{n=1}^{N_t} \left|\mathcal{R}_t[u_\theta](x_i^t)\right|^2, \\ &\mathcal{E}_T(\theta, \mathcal{S})^2 = \mathcal{E}_T^i(\theta, \mathcal{S}_i)^2 + \mathcal{E}_T^s(\theta, \mathcal{S}_s)^2 + \mathcal{E}_T^t(\theta, \mathcal{S}_t)^2, \end{aligned} \tag{2.9}$$

where the training data sets, $\mathcal{S}_i = \{(t_i^n, x_i^n)\}_n^{N_i}$, $\mathcal{S}_s = \{(t_s^n, x_s^n)\}_n^{N_s}$ and $\mathcal{S}_t = \{x_t^n\}_n^{N_t}$, are chosen randomly, independently with respect to the corresponding Lebesgue measures and the residuals $\mathcal{R}_{i,s,t}$ are defined in (2.8).

A *trained PINN* $u^* = u_{\theta^*}$ is then defined as a (local) minimum of the optimization problem (1.6), with loss function (2.9) (possibly with additional data and weight regularization terms), found by a (stochastic) gradient descent algorithm such as ADAM or L-BFGS.

3. Bounds on the approximation error for PINNs

In this section, we will first answer the question Q1 for the PINNs approximating linear Kolmogorov equations (2.1) i.e., our aim will be to construct a deep neural network (2.6) for approximating (2.1), such that the corresponding generalization error $\mathcal{E}_G$ (1.3) is as small as desired.

Recalling that the Kolmogorov PDE is a linear parabolic equation with smooth coefficients, one can use standard parabolic theory to conclude that there exists a unique classical solution u of (2.1) and it is sufficiently regular, for instance $u \in W^{s,\infty}((0,T) \times D)$ for some $s > 2$. As u is a classical solution, the residuals (2.8), evaluated at u, vanish i.e.,

$$\mathcal{R}_i[u](x,t) = 0, \quad \mathcal{R}_s[u](y,t) = 0, \quad \mathcal{R}_t[u](x,0) = 0, \tag{3.1}$$

for all $x \in D, y \in \partial D$.

Moreover, one can use recent results in approximation theory, such as those presented in [9, 10, 5] and references therein, to infer that one can find a deep neural network (2.6) that approximates the solution u in the $W^{2,\infty}$-norm, and therefore yields an approximation for which the PINN residual is small. For instance, one appeals to the following theorem (more details, including exact constants and bounds on the network weights, can be derived from the results in [5]).

Theorem 3.1. *Let $T > 0$, $\gamma, d, s \in \mathbb{N}$ with $s \ge 2 + \gamma$ and let $u \in W^{s,\infty}([0,T] \times [0,1]^d)$ be the solution of a linear Kolmogorov PDE* (2.1)*. Then for every $\varepsilon > 0$ there exists a tanh neural network $\widehat{u}^\varepsilon = u_{\widehat{\theta}^\varepsilon}$ with two hidden layers of width at most $\mathcal{O}(\varepsilon^{-d/(s-2-\gamma)})$ such that $\mathcal{E}_G(\widehat{\theta}^\varepsilon) \le \varepsilon$.*

Proof. It follows from [5, Theorem 5.1] that there exists a tanh neural network $\widehat{u}^{\varepsilon}$ with two hidden layers of width at most $\mathcal{O}(\varepsilon^{-d/(s-2-\gamma)})$ such that

$$\|u-\widehat{u}^{\varepsilon}\|_{W^{2,\infty}([0,T]\times[0,1]^d)} \leq \varepsilon. \tag{3.2}$$

By virtue of the nature of linear Kolmogorov PDEs (2.1) it follows immediately that $\left\|\mathcal{R}_i[u]\right\|_{L^2([0,T]\times[0,1]^d)} \leq \varepsilon$. Using a standard trace inequality, one finds similar bounds for the $\mathcal{R}_s[u]$ and $\mathcal{R}_t[u]$. From this, it follows directly that $\mathcal{E}_G(\widehat{\theta}^{\varepsilon}) \leq \varepsilon$. □

Hence, $\widehat{u}^{\varepsilon}$ is a neural network for which the generalization error (1.3) can be made arbitrarily small, providing an affirmative answer to Q1. However from Theorem 3.1, we observe that the size (width) of the resulting deep neural network $\widehat{u}^{\varepsilon}$, grows *exponentially* with spatial dimension d for (2.1). Thus, this neural network construction clearly suffers from the *curse of dimensionality.* Hence, this construction cannot explain the robust empirical performance of PINNs in approximating Kolmogorov equations (2.1) in very high spatial dimensions [26, 38, 29]. Therefore, we need a different approach for obtaining bounds on the generalization error that overcome this curse of dimensionality. To this end, we rely on the specific structure of the Kolmogorov equations (2.1). In particular, we will use the Dynkin's formula, which relates Kolmogorov PDEs to Itô diffusion SDEs.

In order to state Dynkin's formula, we first need to introduce some notation. Let $(\Omega,\mathcal{F},P,(\mathbb{F}_t)_{t\in[0,T]})$ be a stochastic basis, $D\subseteq\mathbb{R}^d$ a compact set and, for every $x\in D$, let $X^x:\Omega\times[0,T]\to\mathbb{R}^d$ be the solution, in the Itô sense, of the following stochastic differential equation,

$$dX_t^x=\mu(X_t^x)dt+\sigma(X_t^x)dB_t, \quad X_0^x=x, \quad x\in D, t\in[0,T], \tag{3.3}$$

where B_t is a standard d-dimensional Brownian motion on $(\Omega,\mathcal{F},P,(\mathbb{F}_t)_{t\in[0,T]})$. The existence of X^x is guaranteed by Lemma A.5. Dynkin's formula relates the generator $\mathcal{F}$ of X_t^x, given in e.g. [30],

$$\left(\mathcal{F}\varphi\right)(X_t^x)=\sum_{i=1}^d \mu_i(X_t^x)(\partial_i\varphi)(X_t^x)+\frac{1}{2}\sum_{i,j,k=1}^d \sigma_{ik}(X_t^x)\sigma_{kj}(X_t^x)(\partial_{ij}^2\varphi)(X_t^x), \tag{3.4}$$

with the initial condition $\varphi\in C^2(D)$ and differential operator $\mathcal{L}$ (2.7) of the corresponding Kolmogorov PDE (2.1). Equipped with this notation, we state the Dynkin's formula below,

Lemma 3.2 (Dynkin's formula)**.** *For every $x\in D$, let X^x be the solution to a linear Kolmogorov SDE (3.3) with affine $\mu:\mathbb{R}^d\to\mathbb{R}^d$ and $\sigma:\mathbb{R}^d\to\mathbb{R}^{d\times d}$. If $\varphi\in C^2(\mathbb{R}^d)$ with bounded first partial derivatives, then it holds that $(\partial_t u)(x,t)=\mathcal{L}[u](x,t)$ where u is defined as*

$$u(x,t)=\varphi(x)+\mathbb{E}\left[\int_0^t \left(\mathcal{F}\varphi\right)(X_\tau^x)\,d\tau\right], \qquad \text{for } x\in D, t\in[0,T]. \tag{3.5}$$

Proof. See Corollary 6.5 and Section 6.10 in [14]. □

Our construction of a neural network with small residual (2.8) relies on emulating the right hand side of Dynkin's formula (3.5) with neural networks. In particular, the initial data φ and the generator $\mathcal{F}\varphi$ will be approximated by suitable tanh neural networks. On the other hand, the expectation in (3.5) will be replaced by an accurate Monte Carlo sampling. Our construction is summarized in the following theorem,

Theorem 3.3. *Let $\alpha,\beta,\varpi,\zeta,T>0$ and let $p>2$. For every $d\in\mathbb{N}$, let $D_d=[0,1]^d$, $\varphi_d\in C^5(\mathbb{R}^d)$ with bounded first partial derivatives, let $(D_d\times[0,T],\mathcal{F},\mu)$ be a probability space and let $u_d\in C^{2,1}(D_d\times[0,T])$ be a function that satisfies*

$$(\partial_t u_d)(x,t)=\mathcal{L}\left[u_d\right](x,t), \quad u_d(x,0)=\varphi_d(x) \quad \text{for all } (x,t)\in D_d\times[0,T]. \tag{3.6}$$

Moreover, assume that for every $\xi,\delta,c>0$, there exist tanh neural networks $\widehat{\varphi}_{\xi,d}:\mathbb{R}^d\to\mathbb{R}$ and $(\widehat{\mathcal{F}\varphi})_{\delta,d}:\mathbb{R}^d\to\mathbb{R}$ with respectively $\mathcal{O}(d^\alpha\xi^{-\beta})$ and $\mathcal{O}(d^\alpha\delta^{-\beta})$ neurons and weights that grow as $\mathcal{O}(d^\varpi\xi^{-\zeta})$ and $\mathcal{O}(d^\varpi\delta^{-\zeta})$ such that

$$\left\|\varphi_d-\widehat{\varphi}_{\xi,d}\right\|_{C^2(D_d)}\leq\xi \quad \text{and} \quad \left\|\mathcal{F}\varphi-(\widehat{\mathcal{F}\varphi})_{\delta,d}\right\|_{C^2([-c,c]^d)}\leq\delta. \tag{3.7}$$

Then there exist constants $C,\lambda>0$ such that for every $\varepsilon>0$ and $d\in\mathbb{N}$, there exist a constant $\rho_d>0$ and a tanh neural network $\Psi_{\varepsilon,d}$ with at most $C(d\rho_d)^\lambda\varepsilon^{-\max\{5p+3,2+p+\beta\}}$ neurons and weights that grow at most as $C(d\rho_d)^\lambda\varepsilon^{-\max\{\zeta,8p+6\}}$ for $\varepsilon\to0$ such that

$$\left\|\partial_t\Psi_{\varepsilon,d}-\mathcal{L}\left[\Psi_{\varepsilon,d}\right]\right\|_{L^2(D_d\times[0,T])}+\left\|\Psi_{\varepsilon,d}-u_d\right\|_{H^1(D_d\times[0,T])}+\left\|\Psi_{\varepsilon,d}-u_d\right\|_{L^2(\partial(D_d\times[0,T]))}\leq\varepsilon. \tag{3.8}$$

Moreover, ρ_d is defined as

$$\rho_d := \max_{x\in D_d} \sup_{\substack{s,t\in[0,T],\\ s<t}} \frac{\|X^x_s - X^x_t\|_{\mathcal{L}^q(P,\|\cdot\|_{\mathbb{R}^d})}}{|s-t|^{\frac{1}{p}}} < \infty, \tag{3.9}$$

where X^x is the solution, in the Itô sense, of the SDE (3.3) *and $q > 2$ is independent of d.*

Proof. Based on the Dynkin's formula of Lemma 3.2, we will construct a tanh neural network, denoted by $\widehat{u}^{M,N}$ for some $M, N \in \mathbb{N}$, and we will prove that the PINN residual (2.8) of $\widehat{u}^{M,N}$ is small. To do so, we need to define intermediate approximations $\bar{u}^N$ and $\tilde{u}^{M,N}$. In this proof, $C > 0$ will denote a constant that will be updated throughout and can only depend on d, D, μ, T, φ and $\mathcal{L}$, i.e., not on M nor N. In particular, the dependence of C on the input dimension d will be of interest. We will argue that the final value of C will depend polynomially on d and ρ_d (3.9). Because of the third point of Lemma A.5, the quantity within the maximum in the definition of ρ_d (3.9) is finite for every individual $x \in D$ and hence the maximum of this quantity over $x \in \{0, e_1, \dots, e_d\}$ will be finite as well. As a result of the fourth point of Lemma A.5 it then follows that $\rho_d < \infty$. Moreover, if ρ_d depends polynomially on d, then so will C. For notational simplicity, we will not explicitly keep track of the dependence of C on d and ρ_d.

Next, we observe that

(3.10)

$$\max_{x\in D} \sup_{t\in[0,T]} \|X^x_t\|_{\mathcal{L}^q(P,\|\cdot\|_{\mathbb{R}^d})} \leq \max_{x\in D} \sup_{t\in[0,T]} \left(\|x\|_{\mathbb{R}^d} + t^{\frac{1}{p}} \frac{\|X^x_t - x\|_{\mathcal{L}^q(P,\|\cdot\|_{\mathbb{R}^d})}}{t^{\frac{1}{p}}} \right) \leq \max_{x\in D} \|x\|_{\mathbb{R}^d} + (1 + T^{\frac{1}{p}})\rho_d,$$

such that the left-hand side also grows at most polynomially in d and ρ_d.

Finally, we will denote by $\|\cdot\|_2$ the norm $\|\cdot\|_{L^2(D\times[0,T])}$ and to simplify notation we will write $u := u_d$ and $D := D_d$.

Step 1: from u to $\bar{u}^N$. In the first step we approximate the temporal integral in (3.5) by a Riemann sum, that can be readily approximated by neural networks. To this end, let $h : \mathbb{R} \to \mathbb{R}$ be defined by $h(x) = \max\{0, \min\{x, 1\}\}$. Then we define for $N \in \mathbb{N}$,

$$\bar{u}^N(x,t) = \varphi(x) + \frac{T}{N} \sum_{n=1}^{N} \mathbb{E}\left[h\left(\frac{Nt}{T} - n\right) \cdot (\mathcal{F}\varphi)\left(X^x_{\frac{nT}{N}}\right) \right]. \tag{3.11}$$

We first define $n_0(t) = \lfloor Nt/T \rfloor$ and calculate for $t \in \left(\frac{n_0(t)T}{N}, \frac{(n_0(t)+1)T}{N}\right)$,

$$\partial_t(\bar{u}^N - u) = \mathbb{E}\left[(\mathcal{F}\varphi)\left(X^x_{\frac{n_0(t)T}{N}}\right) - (\mathcal{F}\varphi)\left(X^x_t\right) \right]. \tag{3.12}$$

Next, we make the observation that there exist constants a_i, b_i, c_{ij} (that only depend on the coefficients of μ and σ) and functions Λ_i, Ψ_i and Φ_{ij} (that linearly depend on φ and its derivatives) such that

$$(\mathcal{F}\varphi)(Z^x) = \sum_{i=1}^{d} a_i \Lambda_i(Z^x) + \sum_{i=1}^{d} b_i Z^x_i \Psi_i(Z^x) + \sum_{i,j=1}^{d} c_{ij} Z^x_i Z^x_j \Phi_{ij}(Z^x) \tag{3.13}$$

for any d-dimensional stochastic process Z^x. If we define x to be random variable that is uniformly distributed on D, we can use the Lipschitz continuity of Λ_i and the temporal regularity of X^x (property (3) of Lemma A.5 with $\lambda \leftarrow x$) to see that

$$\sup_{t\in[0,T]} \int_D \mathbb{E}\left[\left| \Lambda_i(X^x_{\frac{n_0(t)T}{N}}) - \Lambda_i(X^x_t) \right|^2 \right] dx \leq C \sup_{t\in[0,T]} \int_D \mathbb{E}\left[\left\| X^x_{\frac{n_0(t)T}{N}} - X^x_t \right\|^2 \right] dx \leq \frac{C}{N^{\frac{2}{p}}}. \tag{3.14}$$

Similarly, we find using Lemma A.5 and the generalized Hölder inequality with $q > 0$ such that $\frac{1}{p}+\frac{1}{q}=\frac{1}{2}$,

$$\begin{aligned}
&\sup_{t\in[0,T]}\left(\int_D \mathbb{E}\left[\left|(X^x_{\frac{n_0(t)T}{N}})_i\Psi_i(X^x_{\frac{n_0(t)T}{N}})-(X^x_t)_i\Psi_i(X^x_t)\right|^2\right]dx\right)^{1/2}\\
&\quad\leq \sup_{t\in[0,T]}\left(\int_D \mathbb{E}\left[\left|(X^x_{\frac{n_0(t)T}{N}})_i-(X^x_t)_i\right|^p\right]dx\right)^{1/p}\left(\int_D \mathbb{E}\left[\left|\Psi_i(X^x_{\frac{n_0(t)T}{N}})\right|^q\right]dx\right)^{1/q}\\
&\qquad+\sup_{t\in[0,T]}\left(\int_D \mathbb{E}\left[\left|(X^x_t)_i\right|^q\right]dx\right)^{1/q}\left(\int_D \mathbb{E}\left[\left|\Psi_i(X^x_{\frac{n_0(t)T}{N}})-\Psi_i(X^x_t)\right|^p\right]dx\right)^{1/p}\\
&\quad\leq \sup_{t\in[0,T]} C\left(\int_D \mathbb{E}\left[\left\|X^x_{\frac{n_0(t)T}{N}}-X^x_t\right\|^p\right]dx\right)^{1/p}\leq \frac{C}{N^{1/p}}.
\end{aligned}\tag{3.15}$$

Using also the fact that

$$\sup_{t\in[0,T]}\left(\int_D \mathbb{E}\left[\left|Z^x_iZ^x_j\right|^q\right]dx\right)^{1/q}\leq \sup_{t\in[0,T]}\left(\int_D \mathbb{E}\left[|Z^x_i|^{2q}\right]dx\right)^{1/2q}\sup_{t\in[0,T]}\left(\int_D \mathbb{E}\left[\left|Z^x_j\right|^{2q}\right]dx\right)^{1/2q},\tag{3.16}$$

we can find that

$$\sup_{t\in[0,T]}\left(\int_D \mathbb{E}\left[\left|(X^x_{\frac{n_0(t)T}{N}})_i(X^x_{\frac{n_0(t)T}{N}})_j\Phi_{ij}(X^x_{\frac{n_0(t)T}{N}})-(X^x_t)_i(X^x_t)_j\Phi_{ij}(X^x_t)\right|^2\right]dx\right)^{1/2}\leq\frac{C}{N^{1/p}}.\tag{3.17}$$

As a result, we find that

$$\left\|\partial_t(\bar{u}^N-u)\right\|_2\leq\frac{C}{N^{1/p}}.\tag{3.18}$$

In a similar fashion, one can also find that

$$\left\|\mathcal{L}\left[u-\bar{u}^N\right]\right\|_2\leq\frac{C}{N^{1/p}}.\tag{3.19}$$

To obtain this result, one can use that for all $x\in\mathbb{R}^d$ and $t\in[0,T]$ it holds that

$$X^x_t=\sum_{i=1}^d(X^{e_i}_t-X^0_t)x_i+X^0_t,\tag{3.20}$$

see Lemma A.5. Using this, and writing $X^\cdot_t : D\to\mathbb{R} : x\mapsto X^x_t$, one can calculate that $\mathcal{L}\left[(\mathcal{F}\varphi)\left(X^\cdot_t\right)\right](x)$ is a linear combination of terms of the form $(X^{y_1}_t)_{k_1}\cdots(X^{y_r}_t)_{k_r}F(X^x_t)G(x)$ for $y_1,\dots,y_r\in\{0,e_1,\dots e_d\}$, $1\leq k_1,\dots,k_r\leq d$ (with r independent of d) and where F is a linear combination of φ and its partial derivatives and G is a product of μ and σ and their derivatives. Using these observations and the fact that $\rho_d<\infty$, one can obtain (3.19). Moreover, very similar yet tedious computations yield,

$$\left\|u-\bar{u}^N\right\|_{H^1(D\times[0,T])}\leq\frac{C}{N^{1/p}}.\tag{3.21}$$

Step 2: from $\bar{u}^N$ to $\tilde{u}^{M,N}$. We continue the proof by constructing a Monte Carlo approximation of $\bar{u}^N$. For this purpose, we randomly draw $\omega_m\in\Omega$ for all $m\in\mathbb{N}$ and define for every $M,N\in\mathbb{N}$ the random variable

$$U^{M,N}(x,t)=\varphi(x)+\frac{T}{MN}\sum_{n=1}^N\sum_{m=1}^M h\left(\frac{Nt}{T}-n\right)\cdot(\mathcal{F}\varphi)\left(X^x_{\frac{nT}{N}}(\omega_m)\right).\tag{3.22}$$

Using the same arguments as in the proofs of (3.18) and (3.19), we find for all $(x,t)\in D\times[0,T]$ and $q\in\{t,x_1,\dots x_d\}$ that,

$$\mathbb{E}\left[\left(\partial_qU^{1,N}(x,t)-\mathbb{E}\left[\partial_qU^{1,N}(x,t)\right]\right)^2\right]\leq C\quad\text{and}\quad\partial_q\bar{u}^N(x,t)=\mathbb{E}\left[\partial_qU^{1,N}(x,t)\right].\tag{3.23}$$

Invoking Lemma A.2, we find that

$$\mathbb{E}\left[\left\|\partial_q(U^{M,N}-u)\right\|_2\right]\leq\frac{C}{\sqrt{M}}.\tag{3.24}$$

Similarly, one can prove that

$$\mathbb{E}\left[\left(\mathcal{L}\left[U^{1,N}\right](x,t)-\mathbb{E}\left[\mathcal{L}\left[U^{1,N}\right](x,t)\right]\right)^2\right]\leq C \quad \text{and} \quad \mathcal{L}[u](x,t)=\mathbb{E}\left[\mathcal{L}\left[U^{1,N}\right](x,t)\right]. \tag{3.25}$$

This can be proven using the same arguments as in the proof of (3.19). Using again Lemma A.2 and Lemma A.5, and in combination with our previous result, we find that there is a constant $C_0>0$ independent of M (and with the same properties of C in terms of dependence on d) such that

$$\mathbb{E}\left[\max_{0\leq n\leq N}\max_{y\in\{0,e_1,\dots e_d\}}\left\|X^y_{\frac{nT}{N}}\right\|_{\mathbb{R}^d}+\sqrt{M}\left\|U^M-u\right\|_{H^1(D\times[0,T])}+\sqrt{M}\left\|\mathcal{L}\left[U^M-u\right]\right\|_2\right]\leq C_0 \tag{3.26}$$

and therefore by Lemma A.3 that

$$\mathbb{P}\left(\max_{0\leq n\leq N}\max_{y\in\{0,e_1,\dots e_d\}}\left\|X^y_{\frac{nT}{N}}\right\|_{\mathbb{R}^d}+\sqrt{M}\left\|U^M-u\right\|_{H^1(D\times[0,T])}+\sqrt{M}\left\|\mathcal{L}\left[U^M-u\right]\right\|_2\leq C_0\right)>0. \tag{3.27}$$

The fact that this event has a non-zero probability implies the existence of some *fixed* $\omega_m\in\Omega$, $1\leq m\leq M$, such that for the function

$$\tilde{u}^{M,N}(x,t)=\varphi(x)+\frac{T}{MN}\sum_{n=1}^N\sum_{m=1}^M h\left(\frac{Nt}{T}-n\right)\cdot(\mathcal{F}\varphi)\left(X^x_{\frac{nT}{N}}(\omega_m)\right) \tag{3.28}$$

it holds for all $1\leq m\leq M$ that

$$\left\|\tilde{u}^{M,N}-u\right\|_{H^1(D\times[0,T])}+\left\|\mathcal{L}\left[\tilde{u}^{M,N}-u\right]\right\|_2\leq\frac{C_0}{\sqrt{M}} \quad \text{and} \quad \max_{0\leq n\leq N}\max_{y\in\{0,e_1,\dots e_d\}}\left\|X^y_{\frac{nT}{N}}(\omega_m)\right\|_{\mathbb{R}^d}\leq C_0. \tag{3.29}$$

Step 3: from $\tilde{u}^{M,N}$ to $\widehat{u}^{M,N}$. For every $\epsilon>0$ and $N=N(\epsilon)\in\mathbb{N}$, let h_ϵ be a tanh neural network such that

$$\|h_\epsilon-h\|_{L^\infty(\mathbb{R})}\leq\epsilon,\quad \left\|h'_\epsilon-\chi_{[0,1]}\right\|_{L^2([-N,N])}\leq\epsilon \quad \text{and} \quad \left\|h'_\epsilon\right\|_{L^\infty(\mathbb{R})}\leq 2, \tag{3.30}$$

where $\chi_{[0,1]}$ denotes the indicator function on $[0,1]$. The existence of this neural network is guaranteed by Lemma A.6. Moreover, for $C_1=\max_{x\in[-C_0,C_0]^d}\widehat{(\mathcal{F}\varphi)}_\delta(x)$, we denote the multiplication operator $\times:[-2,2]\times[-2C_1,2C_1]\to\mathbb{R}:(x,y)\mapsto xy$ and every $\eta>0$, we define $\widehat{\times}_\eta:[-2,2]\times[-2C_1,2C_1]\to\mathbb{R}$ to be a tanh neural network such that

$$\left\|\times-\widehat{\times}_\eta\right\|_{C^2([-2,2]\times[-2C_1,2C_1])}\leq\eta. \tag{3.31}$$

If we now in (3.28) replace φ and $\mathcal{F}\varphi$ by $\widehat{\varphi}_\xi$ and $\widehat{(\mathcal{F}\varphi)}_\delta$ as from (3.7), h by h_ϵ and $\times$ by $\widehat{\times}_\eta$, then we end up with the tanh neural network

$$\widehat{u}^{M,N}(x,t)=\widehat{\varphi}_\xi(x)+\frac{T}{MN}\sum_{n=1}^N\sum_{m=1}^M\widehat{\times}_\eta\left(h_\epsilon\left(\frac{Nt}{T}-n\right),\widehat{(\mathcal{F}\varphi)}_\delta\left(X^{x,m}_{\frac{nT}{N}}\right)\right). \tag{3.32}$$

A sketch of this network can be found in Figure 1. In what follows, we will write ∂_1 for the partial derivative to the first component and we will write

$$y_1=h_\epsilon\left(\frac{Nt}{T}-n_0(t)\right),\quad y_2=\widehat{(\mathcal{F}\varphi)}_\delta\left(X^x_{\frac{n_0(t)T}{N}}(\omega_m)\right)\quad y_3=\frac{Nt}{T}-n_0(t),\quad \text{and}\quad y_4=X^x_{\frac{n_0(t)T}{N}}(\omega_m). \tag{3.33}$$

It holds that

$$\begin{aligned}\left\|\partial_t(\widehat{u}^{M,N}-\tilde{u}^{M,N})\right\|_2&\leq\frac{1}{M}\sum_{m=1}^M\left\|\sum_{n\neq n_0(t)}\partial_1\widehat{\times}_\eta(y_1,y_2)\,h'_\epsilon\left(\frac{Nt}{T}-n\right)\right\|_2\\&\quad+\frac{1}{M}\sum_{m=1}^M\left\|\partial_1\widehat{\times}_\eta(y_1,y_2)\,h'_\epsilon(y_3)-(\mathcal{F}\varphi)(y_4)\right\|_2.\end{aligned} \tag{3.34}$$

Using (3.31), we find that

$$\frac{1}{M}\sum_{m=1}^M\left\|\sum_{n\neq n_0(t)}\partial_1\widehat{\times}_\eta(y_1,y_2)\,h'_\epsilon\left(\frac{Nt}{T}-n\right)\right\|_2\leq CN\left\|\widehat{\times}_\eta\right\|_{C^2}\epsilon\leq CN\epsilon. \tag{3.35}$$

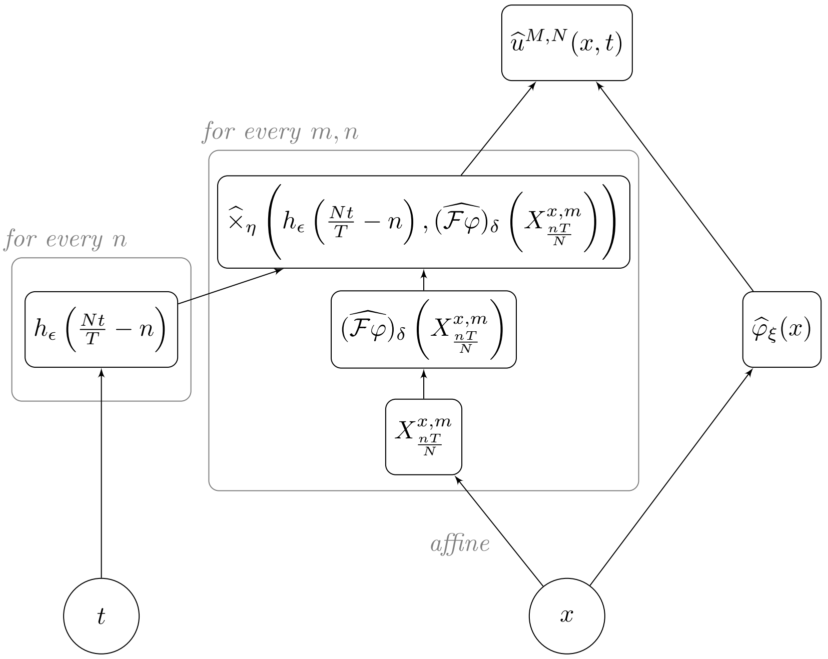
$\widehat{u}^{M,N}(x,t)$
for every m,n
$\widehat{\times}_\eta\left(h_\epsilon\left(\frac{Nt}{T}-n\right),\widehat{(\mathcal{F}\varphi)}_\delta\left(X^{x,m}_{\frac{nT}{N}}\right)\right)$
for every n
$h_\epsilon\left(\frac{Nt}{T}-n\right)$
$\widehat{(\mathcal{F}\varphi)}_\delta\left(X^{x,m}_{\frac{nT}{N}}\right)$
$\widehat{\varphi}_\xi(x)$
$X^{x,m}_{\frac{nT}{N}}$
affine
t
x

FIGURE 1. Flowchart to visualize the construction of the neural network $\widehat{u}^{M,N}(x,t) = \widehat{\varphi}_\xi(x) + \frac{T}{MN}\sum_{n=1}^{N}\sum_{m=1}^{M}\widehat{\times}_\eta\left(h_\epsilon\left(\frac{Nt}{T}-n\right),\widehat{(\mathcal{F}\varphi)}_\delta\left(X^{x,m}_{\frac{nT}{N}}\right)\right)$.

For the other term, we calculate using (3.7), (3.30) and (3.31) that

$$\begin{aligned}
&\left\|\partial_1\widehat{\times}_\eta\left(y_1,y_2\right)h_\epsilon'\left(y_3\right)-\left(\mathcal{F}\varphi\right)\left(y_4\right)\right\|_2\\
&\le\left\|h_\epsilon'\left(y_3\right)\left(\partial_1\widehat{\times}_\eta\left(y_1,y_2\right)-y_2\right)+h_\epsilon'\left(y_3\right)\left(\widehat{(\mathcal{F}\varphi)}_\delta\left(y_4\right)-\left(\mathcal{F}\varphi\right)\left(y_4\right)\right)+\left(\mathcal{F}\varphi\right)\left(y_4\right)\left(h_\epsilon'\left(y_3\right)-\chi_{[0,1]}(y_3)\right)\right\|_2\\
&\le C\left\|h_\epsilon'\right\|_\infty\left\|\times-\widehat{\times}_\eta\right\|_{C^2}+C\left\|h_\epsilon'\right\|_\infty\left\|\widehat{(\mathcal{F}\varphi)}_\delta-\mathcal{F}\varphi\right\|_{C^2}+\left\|\mathcal{F}\varphi\right\|_\infty\left\|h_\epsilon'-\chi_{[0,1]}\right\|_{L^2}\\
&\le C(\eta+\delta+\epsilon).
\end{aligned}\tag{3.36}$$

Thus, we find that

$$\left\|\partial_t(\widehat{u}^{M,N}-\tilde{u}^{M,N})\right\|_2\le C\left(N\epsilon+\eta+\delta\right)\tag{3.37}$$

Finally, we obtain a bound on $\left\|\mathcal{L}\left[\tilde{u}^{M,N}-\widehat{u}^{M,N}\right]\right\|_2$. We simplify notation again by setting

$$z_1=h_\epsilon\left(\frac{Nt}{T}-n\right),\quad z_2=\widehat{(\mathcal{F}\varphi)}_\delta\left(X^x_{\frac{nT}{N}}(\omega_m)\right)\quad z_3=\frac{Nt}{T}-n,\quad\text{and}\quad z_4=X^x_{\frac{nT}{N}}(\omega_m).\tag{3.38}$$

We start off by calculating

$$\begin{aligned}
\mathcal{L}\left[\tilde{u}^{M,N}-\widehat{u}^{M,N}\right]=&\mathcal{L}\left[\varphi-\widehat{\varphi}_\xi\right]+\frac{T}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}h\left(z_3\right)\cdot\mathcal{L}\left[\left(\mathcal{F}\varphi\right)\left(X^{\cdot}_{\frac{nT}{N}}(\omega_m)\right)\right](x)\\
&-\frac{T}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\mathcal{L}\left[\widehat{\times}_\eta\left(z_1,\widehat{(\mathcal{F}\varphi)}_\delta\left(X^{\cdot}_{\frac{nT}{N}}(\omega_m)\right)\right)\right](x).
\end{aligned}\tag{3.39}$$

Explicitly working out the above formula is straightforward, but tedious, and we omit the calculations for the sake of brevity. From this, together with a repeated use of the triangle inequality and (3.29), we find that

$$\begin{aligned}
\left\|\mathcal{L}\left[\tilde{u}^{M,N}-\widehat{u}^{M,N}\right]\right\|_2&\le C\left(\left\|\varphi-\widehat{\varphi}_\xi\right\|_{C^2}+\left\|\times-\widehat{\times}_\eta\right\|_{C^2}+\left\|\mathcal{F}\varphi-\widehat{(\mathcal{F}\varphi)}_\delta\right\|_{C^2}+\left\|h_\epsilon-h\right\|_{L^\infty(\mathbb{R})}\right)\\
&\le C(\xi+\eta+\delta+\epsilon).
\end{aligned}\tag{3.40}$$

Moreover, using similar tools as above we also find that

$$\left\| \tilde{u}^{M,N} - \widehat{u}^{M,N} \right\|_{H^1(D\times[0,T])} \leq C\left(N\epsilon + \xi + \eta + \delta\right). \tag{3.41}$$

Step 4: Total error bound. From the triangle inequality and inequalities (3.18), (3.29), (3.37), (3.40) and (3.19), we get that

$$\begin{aligned}\left\|\partial_t \widehat{u}^{M,N} - L\widehat{u}^{M,N}\right\|_2 &\leq \left\|\partial_t(\widehat{u}^{M,N} - \tilde{u}^{M,N})\right\|_2 + \left\|\partial_t(\tilde{u}^{M,N} - \bar{u}^{N})\right\|_2 + \left\|\partial_t(\bar{u}^{N} - u)\right\|_2 \\ &\quad + \left\|\mathcal{L}\left[u - \bar{u}^N\right]\right\|_2 + \left\|\mathcal{L}\left[\bar{u}^N - \tilde{u}^{M,N}\right]\right\|_2 + \left\|\mathcal{L}\left[\tilde{u}^{M,N} - \widehat{u}^{M,N}\right]\right\|_2 \\ &\leq C\left(\frac{1}{N^{1/p}} + \frac{1}{\sqrt{M}} + (N\epsilon + \eta + \delta) + (\xi + \eta + \delta + \epsilon) + \frac{1}{\sqrt{M}} + \frac{1}{N^{1/p}}\right) \\ &\leq C\left(\frac{1}{N^{1/p}} + \frac{1}{\sqrt{M}} + N\epsilon + \eta + \delta + \xi\right).\end{aligned} \tag{3.42}$$

Similarly, the triangle inequality together with inequalities (3.21), (3.29) and (3.41) gives us,

$$\left\|\widehat{u}^{M,N} - u\right\|_{H^1(D\times[0,T])} \leq C\left(\frac{1}{N^{1/p}} + \frac{1}{\sqrt{M}} + N\epsilon + \eta + \delta + \xi\right). \tag{3.43}$$

Combining this result with a multiplicative trace inequality (e.g. [11, Theorem 3.10.1]) provides us with the result

$$\left\|\widehat{u}^{M,N} - u\right\|_{L^2(\partial(D\times[0,T]))} \leq C\left(\frac{1}{N^{1/p}} + \frac{1}{\sqrt{M}} + N\epsilon + \eta + \delta + \xi\right). \tag{3.44}$$

Step 5: network size. Recall that we need a tanh neural network with $\mathcal{O}(d^\alpha\delta^{-\beta})$ neurons to approximate $\mathcal{F}\varphi$ to an accuracy of $\delta > 0$. Similary for approximating φ, we need a tanh neural network with $\mathcal{O}(d^\alpha\xi^{-\beta})$ neurons.

We first determine the complexity of the network sizes in terms of ε. The network will consist of multiple sub-networks, as illustrated in Figure 1. The first part constructs $M \cdot N$ copies of $(\widehat{\mathcal{F}\varphi})_\delta$, leading to a subnetwork with $\mathcal{O}\left(MN\delta^{-\beta}\right) = \mathcal{O}\left(\varepsilon^{-2-p-\beta}\right)$ neurons. Next, we need N copies of h_ϵ. From Lemma A.6 it follows that for each copy, one needs a subnetwork with two hidden layers of width $\mathcal{O}\left(N^{\frac{1}{2(1-\gamma)}}\epsilon^{\frac{-3}{1-\gamma}}\right)$ for any $\gamma > 0$. One can calculate that N copies of this lead to a width of $\mathcal{O}\left(N^{1+\frac{1}{2(1-\gamma)}}\epsilon^{\frac{-3}{1-\gamma}}\right) = \mathcal{O}\left(\varepsilon^{-5p-3}\right)$. The subnetwork approximating φ consists of $\mathcal{O}(\xi^{-\beta}) = \mathcal{O}(\varepsilon^{-\beta})$ neurons. We assume that the subnetworks to approximate the identity function have a size that is negligible compared to the network sizes of the other parts [5]. Combining these observations with the fact that C depends polynomially on d and ρ_d, we find that there exists a constant $\lambda > 0$ such that the number of neurons of the network is bounded by $\mathcal{O}((d\rho_d)^\lambda\varepsilon^{-\max\{5p+3,2+p+\beta\}})$.

By assumption, the weights of $(\widehat{\mathcal{F}\varphi})_\delta$ and $\widehat{\varphi}_\xi$ scale as $\mathcal{O}(\varepsilon^{-\zeta})$. From [5, Corollary 3.7], it follows that the weights of $\widehat{\times}_\eta$ scale as $\mathcal{O}(\varepsilon^{-1/2})$. Finally, from Lemma A.6, the weights of $\hat{h}_\epsilon$ scale as $\mathcal{O}\left(N^{\frac{1}{(1-\gamma)}}\epsilon^{\frac{-6}{1-\gamma}}\right) = \mathcal{O}\left(\varepsilon^{-8p-6}\right)$. Hence, the weights of the total network $\widehat{u}^{M,N}$ grow as $\mathcal{O}\left((d\rho_d)^\lambda\varepsilon^{-\max\{\zeta,8p+6\}}\right)$, where we possibly adapted the size of λ. □

Remark 3.4. *For the Black-Scholes equation* (2.3)*, the initial condition is to be interpreted as a payoff function. Note that any mollified version of the payoff functions mentioned in Section 2.1 satisfies the regularity requirements of Theorem 3.3. Moreover, because of their compositional structure, these payoff functions and their derivatives can be approximated without the curse of dimensionality. Hence, the assumption* (3.7) *is satisfied as well.*

Theorem 3.3 reveals that the size of the constructed tanh neural network, approximating the underlying solution u of the linear Kolmogorov equation (2.1), and whose PINN residual is as small as desired (3.8), grows with increasing accuracy, but at a rate that is *independent of the underlying dimension d*. Thus, it appears that this neural network overcomes the curse of dimensionality in this sense.

However, Theorem 3.3 reveals that the overall network size grows polynomially in ρ_d. It could be that this constant grows exponentially with dimension. Consequently, the overall network size will be subject to the curse of dimensionality. Given this issue, we will prove that at least for a subclass of Kolmogorov PDEs (2.1), ρ_d only grows polynomially on d. This is for example the case when the coefficients μ and σ are both constant functions.

Theorem 3.5. *Assume the setting of Theorem 3.3 and assume that μ and σ are both constant. Then there exists a constant $\lambda > 0$ such that for every $\varepsilon > 0$ and $d \in \mathbb{N}$, there exists a tanh neural network $\Psi_{\varepsilon,d}$ with $\mathcal{O}(d^{\lambda}\varepsilon^{-\max\{5p+3,2+p+\beta\}})$ neurons and weights that grow as $\mathcal{O}(d^{\lambda}\varepsilon^{-\max\{\zeta,8p+6\}})$ for small ε and large d such that*

$$\left\|\partial_t \Psi_{\varepsilon,d} - \mathcal{L}\left[\Psi_{\varepsilon,d}\right]\right\|_{L^2(D\times[0,T])} + \left\|\Psi_{\varepsilon,d} - u_d\right\|_{H^1(D\times[0,T])} + \left\|\Psi_{\varepsilon,d} - u_d\right\|_{L^2(\partial(D\times[0,T]))} \leq \varepsilon. \tag{3.45}$$

Proof. We show that when μ and σ are both constant functions, the constant ρ_d, as defined in (3.9), grows only polynomially in d. It is well-known that in this setting the solution process to the SDE (3.3) is given by $X_t^x = x + \mu t + \sigma B_t$, where $(B_t)_{t\in[0,T]}$ is a d-dimensional Brownian motion. The fact that ρ_d only grows polynomially in d then follows directly from the Lévy's modulus of continuity (Lemma A.4). The corollary then is a direct consequence of Theorem 3.3 □

Thus, we have been able to answer question Q1 by showing that there exists a neural network, for which the PINN residual (generalization error) (1.3) is as small as desired. In this process, we have also answered Q2 for this particular tanh neural network as the bound (3.43) clearly shows that the overall error (in the L^2-norm and even H^1-norm) of the tanh neural network $\Psi_{\varepsilon,d}$ is arbitrarily small.

Although in this particular case, an affirmative answer to question Q2 was a by-product of the proof of question Q1, it turns out that one can follow the recent paper [26] and leverage the stability of Kolmogorov PDEs to answer question Q2 in much more generality, by showing that as long as the generalization error is the small, the overall error is proportionately small. We have the following precise statement about this fact,

Theorem 3.6. *Let u be a (classical) solution to a linear Kolmogorov equation (2.1) with $\mu \in C^1(D;\mathbb{R}^d)$ and $\sigma \in C^2(D;\mathbb{R}^{d\times d})$, u_θ a PINN and let the residuals be defined by (2.8). Then*

$$\begin{aligned}\|u - u_\theta\|^2_{L^2(D\times[0,T])} \leq C_1 \Bigg[&\big\|\mathcal{R}_i[u_\theta]\big\|^2_{L^2(D\times[0,T])} + \big\|\mathcal{R}_t[u_\theta]\big\|^2_{L^2(D)} \\ &+ C_2\big\|\mathcal{R}_s[u_\theta]\big\|_{L^2(\partial D\times[0,T])} + C_3\big\|\mathcal{R}_s[u_\theta]\big\|^2_{L^2(\partial D\times[0,T])}\Bigg],\end{aligned} \tag{3.46}$$

where $C_0 = \sum_{i,j=1}^d \left\|\partial_{ij}(\sigma\sigma^T)_{ij}\right\|_{L^\infty(D\times[0,T])}$, $C_1 = Te^{(C_0+\|\mathrm{div}\mu\|_\infty+1)T}$, $C_2 = \sum_{i=1}^d \left\|(\sigma\sigma^T J_x[u-u_\theta]^T)_i\right\|_{L^2(\partial D\times[0,T])}$ *and* $C_3 = \|\mu\|_\infty + \sum_{i,j,k=1}^d \left\|\partial_i(\sigma_{ik}\sigma_{jk})\right\|_{L^\infty(\partial D\times[0,T])}$.

Proof. Let $\hat{u} = u_\theta - u$. Integrating $\mathcal{R}_i[\hat{u}](t,x)$ over D and rearranging terms gives

$$\frac{1}{2}\frac{d}{dt}\int_D |\hat{u}|^2 = \frac{1}{2}\int_D \mathrm{Trace}(\sigma\sigma^T H_x[\hat{u}])\hat{u} + \int_D \mu J_x[\hat{u}]\hat{u} + \int_D \mathcal{R}_i[\hat{u}]\hat{u} \tag{3.47}$$

where all integrals are to be interpreted as integrals with respect to the Lebesgue measure on D, resp. ∂D. For the first term of (3.47), we observe that $\mathrm{Trace}(\sigma\sigma^T H_x[\hat{u}]) = \sum_{i,j,k=1}^d \sigma_{ik}\sigma_{jk}\partial_{ij}\hat{u}$ and also that

$$\int_D \partial_i(\sigma_{ik}\sigma_{jk})\hat{u}\partial_j\hat{u} = \int_{\partial D} \partial_i(\sigma_{ik}\sigma_{jk})\hat{u}^2(\hat{e}_j\cdot\hat{n}) - \int_D \partial_i(\sigma_{ik}\sigma_{jk})\hat{u}\partial_j\hat{u} - \int_D \partial_{ij}(\sigma_{ik}\sigma_{jk})\hat{u}^2 \tag{3.48}$$

for any $1 \leq i,j,k \leq d$. Next, we define

$$c_1 = 2\sum_{i=1}^d \left\|(\sigma\sigma^T J_x[\hat{u}]^T)_i\right\|_{L^2(\partial D\times[0,T])}, c_2 = \sum_{i,j,k=1}^d \left\|\partial_i(\sigma_{ik}\sigma_{jk})\right\|_{L^\infty(\partial D\times[0,T])}, c_3 = \sum_{i,j=1}^d \left\|\partial_{ij}(\sigma\sigma^T)_{ij}\right\|_{L^\infty(D\times[0,T])}. \tag{3.49}$$

From this, using integration by parts and letting $\hat{n}$ denote the unit normal on ∂D, we find that

$$
\begin{aligned}
&\int_D \mathrm{Trace}(\sigma\sigma^T H_x[\hat{u}])\hat{u} \\
&= \sum_{i,j,k=1}^{d} \left[\int_{\partial D} \sigma_{ik}\sigma_{jk}\hat{u}\partial_j\hat{u}(\hat{e}_i\cdot\hat{n}) - \int_D \sigma_{ik}\sigma_{jk}\partial_i\hat{u}\partial_j\hat{u} - \int_D \partial_i(\sigma_{ik}\sigma_{jk})\hat{u}\partial_j\hat{u}\right] \\
&= \sum_{i,j,k=1}^{d} \left[\int_{\partial D} \sigma_{ik}\sigma_{jk}\hat{u}\partial_j\hat{u}(\hat{e}_i\cdot\hat{n}) - \int_D \sigma_{ik}\sigma_{jk}\partial_i\hat{u}\partial_j\hat{u} - \frac{1}{2}\int_{\partial D} \partial_i(\sigma_{ik}\sigma_{jk})\hat{u}^2(\hat{e}_j\cdot\hat{n}) + \frac{1}{2}\int_D \partial_{ij}(\sigma_{ik}\sigma_{jk})\hat{u}^2\right] \\
&\leq \sum_{i=1}^{d} \int_{\partial D} \left|(\sigma\sigma^T J_x(\hat{u})^T)_i\hat{u}(\hat{e}_i\cdot\hat{n})\right| - \underbrace{\int_D J_x[\hat{u}]\sigma(J_x[\hat{u}]\sigma)^T}_{\geq 0} + \frac{c_2}{2}\int_{\partial D} \left|\mathcal{R}_s[u_\theta]\right|^2 + \frac{c_3}{2}\int_D \hat{u}^2.
\end{aligned}
\tag{3.50}
$$

For the second term of (3.47), we find that

$$
\begin{aligned}
\int_D \mu J_x[\hat{u}]\hat{u} &= \frac{1}{2}\int_D \mu J_x[\hat{u}^2] = -\frac{1}{2}\int_D \hat{u}^2\mathrm{div}\mu + \frac{1}{2}\int_{\partial D} \hat{u}^2\mu^T\cdot\hat{n} \\
&\leq \frac{1}{2}\|\mathrm{div}\mu\|_\infty \int_D \hat{u}^2 + \frac{1}{2}\|\mu\|_\infty \int_{\partial D} \left|\mathcal{R}_s[u_\theta]\right|^2
\end{aligned}
\tag{3.51}
$$

Finally, we find for the third term of the right-hand side of (3.47) that

$$
\int_D \mathcal{R}_i[\hat{u}]\hat{u} \leq \frac{1}{2}\int_D \mathcal{R}_i[\hat{u}]^2 + \frac{1}{2}\int_D \hat{u}^2
\tag{3.52}
$$

Integrating (3.47) over the interval $[0,\tau]\subset[0,T]$, using all the previous inequalities together with Hölder's inequality, we find that

$$
\begin{aligned}
\int_D \left|\hat{u}(x,\tau)\right|^2 dx \leq \int_D \left|\mathcal{R}_t[u_\theta]\right|^2 + c_1\left(\int_{\partial D\times[0,T]} \left|\mathcal{R}_s[u_\theta]\right|^2\right)^{1/2} + \int_{D\times[0,T]} \left|\mathcal{R}_i[\hat{u}]\right|^2 \\
+ (c_2 + \|\mu\|_\infty)\int_{\partial D\times[0,T]} \left|\mathcal{R}_s[u_\theta]\right|^2 + (c_3 + \|\mathrm{div}\mu\|_\infty + 1)\int_{[0,\tau]}\int_D \left|\hat{u}(x,s)\right|^2 dxds.
\end{aligned}
\tag{3.53}
$$

Using Grönwall's inequality and integrating over $[0,T]$ then gives

$$
\begin{aligned}
\int_{D\times[0,T]} |\hat{u}|^2 \leq Te^{(c_3+\|\mathrm{div}\mu\|_\infty+1)T}\Bigg[\int_D \left|\mathcal{R}_t[u_\theta]\right|^2 + c_1\left(\int_{\partial D\times[0,T]} \left|\mathcal{R}_s[u_\theta]\right|^2\right)^{1/2} \\
+ \int_{D\times[0,T]} \left|\mathcal{R}_i[\hat{u}]\right|^2 + (c_2 + \|\mu\|_\infty)\int_{\partial D\times[0,T]} \left|\mathcal{R}_s[u_\theta]\right|^2\Bigg].
\end{aligned}
\tag{3.54}
$$

Renaming the constants yields the statement of the theorem. □

Thus the bound (3.46) clearly shows that controlling the generalization error (1.3) suffices to control the L^2-error for the PINN, approximating the Kolmogorov equations (2.1). In particular, combining Theorem 3.6 with Theorem 3.3 then proves that it is possible to approximate solutions to linear Kolmogorov equations in L^2-norm at a rate that is independent of the spatial dimension d.

4. Generalization error of PINNs

Having answered the questions Q1 and Q2 on the smallness of the PINN residual (generalization error (1.3)) and the total error for PINNs approximating the Kolmogorov PDEs (2.1), we turn our attention to question Q3 i.e., given small training error (2.9) and for sufficiently many training samples $\mathcal{S}_{i,s,t}$, can one show that the generalization error (1.3) (and consequently the total error by Theorem 3.6) is proportionately small?

To this end, we start with the observation that the PINN residual as well training error (2.9) has three parts, two *data terms* corresponding to the mismatches with the initial and boundary data and a *residual term* that measures the amplitude of the PDE residual. Thus, we can embed these two types of terms in the following very general set-up: let $D\subset\mathbb{R}^d$ be compact and let $f: D\to\mathbb{R}$, $f_\theta: D\to\mathbb{R}$ be functions for all $\theta\in\Theta$. We can think of f as the ground truth for the initial or boundary data for the PDE (2.1) and f_θ be the corresponding restriction of approximating PINNs to the spatial or temporal boundaries. Similarly, we can think of $f \equiv 0$ as the PDE residual, corresponding to the exact solution of (2.1) and f_θ is the *interior* PINN residual (first term in (2.8)), for a neural network with weights θ. Let $M \in \mathbb{N}$ be the training set size and let $\mathcal{S} = \{z_1, \dots, z_M\} \subset D^M$ be the training set, where each z_i is independently drawn according to some probability measure μ on D. We define the (squared) training error, generalization error and empirical risk minimizer as

(4.1)
$$\mathcal{E}_T(\theta, \mathcal{S})^2 = \frac{1}{M} \sum_{i=1}^{M} \left| f_\theta(z_i) - f(z_i) \right|^2, \quad \mathcal{E}_G(\theta)^2 = \int_D \left| f_\theta(z) - f(z) \right|^2 d\mu(z), \quad \theta^*(\mathcal{S}) \in \arg\min_{\theta\in\Theta} \mathcal{E}_T(\theta, \mathcal{S})^2,$$

where we restrict ourselves to the (squared) L^2-norm only for definiteness, while claiming that all the subsequent results readily extend to general L^p-norms for $1 \le p < \infty$. It is easy to see that the above set-up encompasses all the terms in the definitions of the generalization error (1.3) and training error (2.9) for PINNs.

Our first aim is to decompose this very general form of generalization error in (4.1) as,

Lemma 4.1. *Let $k \in \mathbb{N}$ and $\Theta \subset \mathbb{R}^k$ compact. Then it holds that*

$$\begin{aligned} \mathcal{E}_G(\theta^*(\mathcal{S}))^2 \le & \sup_{\substack{\theta,\vartheta\in\Theta:\\ \|\theta-\vartheta\|\le\delta}} \left| \mathcal{E}_G(\vartheta)^2 - \mathcal{E}_G(\theta)^2 \right| + \sup_{\theta\in\Theta} \left| \mathcal{E}_G(\theta)^2 - \mathcal{E}_T(\theta)^2 \right| \\ & + \sup_{\substack{\theta,\vartheta\in\Theta:\\ \|\theta-\vartheta\|\le\delta}} \left| \mathcal{E}_T(\theta, \mathcal{S})^2 - \mathcal{E}_T(\vartheta, \mathcal{S})^2 \right| + \mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})^2. \end{aligned} \tag{4.2}$$

Proof. Since Θ is compact, there exist for every $\delta > 0$ a natural number $N = N(\delta) \in \mathbb{N}$ and parameters $\theta_1, \dots \theta_N \in \Theta$ such that for all $\theta \in \Theta$ there exists $1 \le i \le N$ such that $\|\theta - \theta_i\|_\infty \le \delta$. For every $1 \le i \le N$ it holds that

$$\mathcal{E}_G(\theta^*(\mathcal{S}))^2 \le \left| \mathcal{E}_G(\theta^*(\mathcal{S}))^2 - \mathcal{E}_G(\theta_i)^2 \right| + \left| \mathcal{E}_G(\theta_i)^2 - \mathcal{E}_T(\theta_i)^2 \right| + \left| \mathcal{E}_T(\theta_i)^2 - \mathcal{E}_T(\theta^*)^2 \right| + \mathcal{E}_T(\theta^*(\mathcal{S}))^2. \tag{4.3}$$

This error decomposition holds in particular for $i^* = i^*(\theta^*) \in \arg\min_i \|\theta^* - \theta_i\|_\infty$. Using that $\|\theta^* - \theta_{i^*}\|_\infty \le \delta$ and then majorizing gives the bound from the statement. □

Note that we have leveraged the compactness of the parameter space Θ in (4.2) to decompose the generalization error in terms of the training error $\mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})$, the so-called *generalization gap* i.e., $\sup_{\theta\in\Theta} \left| \mathcal{E}_G(\theta)^2 - \mathcal{E}_T(\theta)^2 \right|$ and error terms that measure the modulus of continuity of the generalization and training errors. From this decomposition, we can intuitively see that these error terms can be made suitably small by requiring that the generalization and training errors are, for instance, Lipschitz continuous. Then, we can use standard concentration inequalities to obtain the following *very general* bound on the generalization error in terms of the training error,

Theorem 4.2. *Let $a, c, \mathfrak{L} > 0$, $k, d, M \in \mathbb{N}$, $D \subset \mathbb{R}^d$ compact, $(\Omega, \mathcal{A}, \mathbb{P})$ a probability space, $\Theta = [-a, a]^k$ and let $f : D \to \mathbb{R}$ and $f_\theta : D \to \mathbb{R}$ be functions for all $\theta \in \Theta$. Let $X_i : \Omega \to D$, $1 \le i \le M$ be iid random variables, $\mathcal{S} = \{X_1, \dots X_M\}$ and let $\theta^*(\mathcal{S})$ be a minimizer of $\theta \mapsto \mathcal{E}_T(\theta, \mathcal{S})^2$. Let $\mathcal{E}_T(\theta)^2, \mathcal{E}_G(\theta, \mathcal{S})^2 \in [0, c]$ for all $\theta \in \Theta$ and $\mathcal{S} \subset D^M$ and let $\theta \mapsto \mathcal{E}_G(\theta)^2$ and $\theta \mapsto \mathcal{E}_T(\theta, \mathcal{S})^2$ be Lipschitz continuous with Lipschitz constant $\mathfrak{L}$. For every $\epsilon, \eta > 0$, it holds that*

$$\mathbb{P}\left(\mathcal{E}_G(\theta^*(\mathcal{S})) \le \epsilon + \mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})\right) \ge 1 - \eta \quad \textit{if} \quad M \ge \frac{c^2}{2\epsilon^4}\left(k \ln\left(\frac{2a\mathfrak{L}}{\epsilon^2}\right) + \ln\left(\frac{1}{\eta}\right)\right). \tag{4.4}$$

Proof. For arbitrary $\epsilon > 0$, set $\delta = \frac{\epsilon^2}{2\mathfrak{L}}$ and let $\{\theta_i\}_{i=1}^N$ be a δ-covering of Θ with respect to the supremum norm. Then it holds that N can be bounded by $(2a\mathfrak{L}/\epsilon^2)^k$ and moreover

$$\sup_{\theta,\vartheta\in\Theta:\|\theta-\vartheta\|\le\delta} \left| \mathcal{E}_G(\vartheta)^2 - \mathcal{E}_G(\theta)^2 \right| + \sup_{\theta,\vartheta\in\Theta:\|\theta-\vartheta\|\le\delta} \left| \mathcal{E}_T(\theta, \mathcal{S})^2 - \mathcal{E}_T(\vartheta, \mathcal{S})^2 \right| \le \epsilon. \tag{4.5}$$

Then it holds for every $1 \le i \le N$ that

(4.6)
$$\mathcal{E}_G(\theta^*(\mathcal{S}))^2 \le \left| \mathcal{E}_G(\theta^*(\mathcal{S}))^2 - \mathcal{E}_G(\theta_i)^2 \right| + \left| \mathcal{E}_G(\theta_i)^2 - \mathcal{E}_T(\theta_i, \mathcal{S})^2 \right| + \left| \mathcal{E}_T(\theta_i, \mathcal{S})^2 - \mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})^2 \right| + \mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})^2.$$

Next, we define a projection $\mathcal{P} : \Theta \to \Theta$ that maps θ to a unique θ_{i^*} with $i^* \in \arg\min_i \|\theta - \theta_i\|_\infty$ and we define the following events for $1 \leq i \leq N$,

$$\begin{aligned}
&\mathcal{A} = \left\{\mathcal{E}_G(\theta^*(\mathcal{S}))^2 \leq \epsilon^2 + \mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})^2\right\}, \quad \mathcal{B}_i = \left\{\mathcal{E}_G(\theta_i)^2 \leq \epsilon^2 + \mathcal{E}_T(\theta_i, \mathcal{S})^2\right\}, \quad \mathcal{C}_i = \left\{\mathcal{P}(\theta^*(\mathcal{S})) = \theta_i\right\}, \\
&\mathcal{D} = \left\{\exists i \in \{1, \dots, N\} : \left(\mathcal{E}_G(\theta_i)^2 \leq \epsilon^2 + \mathcal{E}_T(\theta_i, \mathcal{S})^2\right) \text{ and } (\mathcal{P}(\theta^*(\mathcal{S})) = \theta_i)\right\}.
\end{aligned} \tag{4.7}$$

Note that (4.5) and (4.6) imply that $\mathcal{D} \subseteq \mathcal{A}$ and thus $\mathbb{P}(\mathcal{D}) \leq \mathbb{P}(\mathcal{A})$. Next, by the definition of $\mathcal{P}$ it holds that $\mathcal{P}$ induces a partition on Θ and thus $\sum_i \mathcal{P}(\mathcal{C}_i) = 1$. As $\mathcal{E}_T(\theta, \{X_i\})^2 : \Omega \to [0, c]$ and $\mathbb{E}\left[\mathcal{E}_T(\theta, \{X_i\})^2\right] = \mathcal{E}_G(\theta)^2$ for all i, Hoeffding's inequality (Lemma C.1) proves that $\mathbb{P}(\mathcal{B}_i) \geq 1 - \exp\left(-2\epsilon^4 M/c^2\right)$. Combining this with the observation that $\mathcal{D} = \bigsqcup_{i=1}^N (\mathcal{B}_i \cap \mathcal{C}_i)$ then proves that

$$\begin{aligned}
\mathbb{P}(\mathcal{A}) \geq \mathbb{P}(\mathcal{D}) &= \sum_{i=1}^N \mathbb{P}(\mathcal{B}_i \cap \mathcal{C}_i) \geq \sum_{i=1}^N (\mathbb{P}(\mathcal{B}_i) + \mathbb{P}(\mathcal{C}_i) - \mathbb{P}(\mathcal{B}_i \cup \mathcal{C}_i)) \\
&\geq 1 + \sum_{i=1}^N (\mathbb{P}(\mathcal{B}_i) - 1) \geq 1 - N \exp\left(\frac{-2\epsilon^4 M}{c^2}\right) \geq 1 - \left(\frac{2a\mathfrak{L}}{\epsilon^2}\right)^k \exp\left(\frac{-2\epsilon^4 M}{c^2}\right).
\end{aligned} \tag{4.8}$$

As a consequence, it holds that

$$\begin{aligned}
M \geq \frac{c^2}{2\epsilon^4}\left(k \ln\left(\frac{2a\mathfrak{L}}{\epsilon^2}\right) + \ln\left(\frac{1}{\eta}\right)\right) \quad &\Longrightarrow \quad \mathbb{P}\left(\mathcal{E}_G(\theta^*(\mathcal{S}))^2 \leq \epsilon^2 + \mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})^2\right) \geq 1 - \eta \\
&\Longrightarrow \quad \mathbb{P}\left(\mathcal{E}_G(\theta^*(\mathcal{S})) \leq \epsilon + \mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})\right) \geq 1 - \eta.
\end{aligned} \tag{4.9}$$

□

The bound on the generalization error in terms of the training error (4.4) is a probabilistic statement. It can readily be recast in terms of *averages* by defining the so-called *cumulative* generalization and training errors of the form,

$$\overline{\mathcal{E}}_G^2 = \int_{D^M} \mathcal{E}_G(\theta^*(\mathcal{S}))^2 d\mu^M(\mathcal{S}), \quad \overline{\mathcal{E}}_T^2 = \int_{D^M} \mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})^2 d\mu^M(\mathcal{S}). \tag{4.10}$$

Here $\mu^M = \mu \otimes \mu \ldots \otimes \mu$ is the induced product measure on the training set $\mathcal{S}$. We have the following *ensemble* version of Theorem 4.2;

Corollary 4.3. *Assume the setting of Theorem 4.2. It holds that*

$$\overline{\mathcal{E}}_G \leq \epsilon + \overline{\mathcal{E}}_T \quad \textit{if} \quad M \geq \frac{2c^2}{\epsilon^4}\left(k \ln\left(\frac{4a\mathfrak{L}}{\epsilon^2}\right) + \ln\left(\frac{2c}{\epsilon^2}\right)\right). \tag{4.11}$$

Proof. Let $X = \mathcal{E}_G(\theta^*(\mathcal{S}))^2 - \mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})^2$. Using (the last step of the proof of) Theorem 4.2 with $\eta = \frac{\epsilon^2}{2c}$ then gives that

$$\mathbb{E}[X] = \mathbb{E}[X \mathbb{1}_{X \leq \frac{\epsilon^2}{2}}] + \mathbb{E}[X \mathbb{1}_{X > \frac{\epsilon^2}{2}}] \leq \frac{\epsilon^2}{2} + c\mathbb{P}\left(X > \frac{\epsilon^2}{2}\right) \leq \epsilon^2, \tag{4.12}$$

provided that $M \geq \frac{2c^2}{\epsilon^4}\left(k \ln\left(\frac{4a\mathfrak{L}}{\epsilon^2}\right) + \ln\left(\frac{2c}{\epsilon^2}\right)\right)$. □

As a first example for illustrating the bounds of Theorem 4.2 (and Corollary 4.3), we apply it to the estimation of the generalization errors, corresponding to the spatial and temporal boundaries, in terms of the corresponding training errors (2.9). These bounds readily follow from the following general bound.

Corollary 4.4. *Let $L, W \in \mathbb{N}$, $R \geq 1$, $L \geq 2$ and let $f_\theta : D \to \mathbb{R}$, $\theta \in \Theta$, be tanh neural networks with at most $L - 1$ hidden layers, width at most W and weights and biases bounded by R. For every $0 < \epsilon < 1$, it holds that for the generalization and training error* (4.1) *that,*

$$\mathbb{P}\left(\mathcal{E}_G(\theta^*(\mathcal{S})) \leq \epsilon + \mathcal{E}_T(\theta^*(\mathcal{S}), \mathcal{S})\right) \geq 1 - \eta \quad \textit{if} \quad M \geq \frac{16d(L+3)^2 W^6 R^4}{\epsilon^4} \ln\left(\frac{4\sqrt[5]{d+4}RW}{\epsilon}\right). \tag{4.13}$$

Proof. Using the inverse triangle inequality and the fact that $a^2 - b^2 = (a+b)(a-b)$ for $a, b \in \mathbb{R}$, we find for $\theta, \vartheta \in \Theta$ that

$$\begin{aligned} \left| \int_D \left|f_\theta(x) - f(x)\right|^2 - \left|f_\vartheta(x) - f(x)\right|^2 d\mu(x) \right| &\leq 4R \int_D \Big| \left|f_\theta(x) - f(x)\right| - \left|f_\vartheta(x) - f(x)\right| \Big| d\mu(x) \\ &\leq 4R \int_D \left|f_\theta(x) - f_\vartheta(x)\right| d\mu(x). \end{aligned} \tag{4.14}$$

Combining this with Lemma B.3 and Lemma C.2 proves that the Lipschitz constant of the map $\theta \mapsto f_\theta$ is at most $4(d+4)R^L W^{L-1}$. We can then use Corollary 4.3 with $a \leftarrow R$, $\mathfrak{L} \leftarrow 4(d+4)R^L W^{L-1}$ and $c \leftarrow 4W^2R^2$ (from (4.1)). Moreover, one can calculate that every f_θ has at most $(d + (L-2)W + 1)W$ weights and $(L-1)W + 1$ biases, such that $k \leftarrow 2dLW^2$. Next, we make the estimate

$$\frac{c^2}{2\epsilon^4} \left(k \ln\left(\frac{2a\mathfrak{L}}{\epsilon^2} \right) + \ln\left(\frac{2c}{\epsilon^2} \right) \right) \leq \frac{8W^4R^4}{\epsilon^4} \cdot 2dLW^2 \ln\left(\frac{2^6(d+4)R^{L+3}W^{L-1}}{\epsilon^4} \right) \leq \frac{16d(L+3)^2W^6R^4}{\epsilon^4} \ln\left(\frac{4\sqrt[5]{d+4}RW}{\epsilon} \right). \tag{4.15}$$

□

Next, we will apply the above general results to PINNs for the Kolmogorov equation (2.1). The following corollary provides an estimate on the (cumulative) PINN generalization error and can be seen as the counterpart of Corollary 4.4. It is based on the fact that neural networks and their derivatives are Lipschitz continuous in the parameter vector, the proof of which can be found in Appendix B. Consequently, the PINN generalization error is Lipschitz as well (cf. Lemma C.3).

Corollary 4.5. *Let $L, W \in \mathbb{N}$, $R \geq 1$, $a, b \in \mathbb{R}$ with $a < b$ and let $u_\theta : [a,b]^d \to \mathbb{R}$, $\theta \in \Theta$, be tanh neural networks with smooth activation function σ, at most $L-1$ hidden layers, width at most W and weights and biases bounded by R. For $q = i, t, s$ let the PINN generalization $\mathcal{E}_G^q$ and training $\mathcal{E}_T^q$ errors for linear Kolmogorov PDEs (cf. Section 2.1) and let $c_q > 0$ be such that $\mathcal{E}_T^q(\theta)^2, \mathcal{E}_G^q(\theta, \mathcal{S})^2 \in [0, c_q]$ for all $\theta \in \Theta$ and $\mathcal{S} \subset D^M$. Assume that $\max\{\|\varphi\|_\infty, \|\psi\|_\infty\} \leq \max_{\theta\in\Theta} \|u_\theta\|_\infty$ and define the constants*

$$\begin{aligned} \alpha &= \max\{1, |a|, |b|, \|\sigma\|_\infty\}, \qquad \beta = \max\{1, \|\sigma'\|_\infty, \|\sigma''\|_\infty, \|\sigma'''\|_\infty\}, \\ C &= \max_{x\in D} \left(1 + \sum_{i=1}^d |\mu(x)_i| + \sum_{i,j=1}^d |(\sigma(x)\sigma(x)^*)_{ij}| \right). \end{aligned} \tag{4.16}$$

Then for any $\epsilon > 0$ it holds that

$$\overline{\mathcal{E}}_G^q \leq \epsilon + \overline{\mathcal{E}}_T^q \quad \text{if } M_q \geq \frac{24dL^2W^2c_q^2}{\epsilon^4} \ln\left(4c_q RW\beta \sqrt[6]{\frac{C(d+7)}{\epsilon^2}} \right). \tag{4.17}$$

Proof. Setting $C = \max_{x\in D} \left(1 + \sum_{i=1}^d |\mu(x)_i| + \sum_{i,j=1}^d |(\sigma(x)\sigma(x)^*)_{ij}| \right)$, we can use Corollary 4.3 with $a \leftarrow R$, $c \leftarrow c_q$, $\mathfrak{L} \leftarrow 2^5C^2(d+7)^2L^4R^{6L-1}W^{6L-6}\beta^{2L}$ (cf. Lemma C.3) and $k \leftarrow 2dLW^2$ (cf. proof of Corollary 4.4). We then calculate

$$k \ln\left(\frac{4a\mathfrak{L}}{\epsilon^2} \right) + \ln\left(\frac{2c_q}{\epsilon^2} \right) \leq 6kL \ln\left(4c_q RW\beta \sqrt[6]{\frac{C(d+7)}{\epsilon^2}} \right) = 12dL^2W^2 \ln\left(4c_q RW\beta \sqrt[6]{\frac{C(d+7)}{\epsilon^2}} \right). \tag{4.18}$$

□

Remark 4.6. *Corollary 4.5 requires bounds c_q on the training errors $\mathcal{E}_T^q$ and the generalization errors $\mathcal{E}_G^q$ of the PINN. Lemma C.3 provides such bounds, given by $c_i = 4\alpha C(d+7)L^2R^{3L}W^{3L-3}\beta^L$ and $c_t = c_s = 2WR$. Although the values for c_t and c_s are of reasonable size, the value for c_i is likely to be a large overestimate. It might makes sense to consider the approximation*

$$c_i \approx \max_{n,m} \mathcal{E}_T^i(\theta_n, \{x_m\}) \tag{4.19}$$

for some randomly sampled $\theta_n \in \Theta$ and $x_m \in D$.

Combining Corollary 4.5 with Theorem 3.6 allows us to bound the L^2-error of the PINN in terms of the (cumulative) training error and the training set size. The following corollary proves that a well-trained PINN on average has a low L^2-error provided that the training set is large enough. It is also possible to prove a similar probabilistic statement instead of a statement that holds on average.

Corollary 4.7. *Let u be a (classical) solution to a linear Kolmogorov equation* (2.1) *with $\mu \in C^1(D;\mathbb{R}^d)$ and $\sigma \in C^1(D;\mathbb{R}^{d\times d})$, $u^* = u_{\theta^*(\mathcal{S})}$ a trained PINN, let $\overline{\mathcal{E}}_T^i, \overline{\mathcal{E}}_T^s$ and $\overline{\mathcal{E}}_T^t$ denote the interior, spatial and temporal cumulative training error, cf.* (1.3) *and let C_1, C_2 and C_3 be the constants as defined in Theorem 3.6. If the training set sizes where chosen as in* (4.17) *of Corollary 4.5 for some $\epsilon > 0$, then*

(4.20)

$$\int_{(D\times[0,T])^M}\int_{D\times[0,T]} \left|u(x,t) - u_{\theta^*(\mathcal{S})}(x,t)\right|^2 dxdtd\mu^M(\mathcal{S}) \le C_1\left[(\overline{\mathcal{E}}_T^i)^2 + (\overline{\mathcal{E}}_T^t)^2 + C_2(\overline{\mathcal{E}}_T^s + \sqrt{\epsilon}) + C_3(\overline{\mathcal{E}}_T^s)^2 + (C_3+2)\epsilon\right].$$

Proof. This is a direct consequence of Corollary 4.5 and the proof of Theorem 3.6 (in particular, one needs to take the expectation of all training sets $\mathcal{S}$ before applying Hölder's inequality in the proof of Theorem 3.6). □

Thus, in Corollaries 4.5 and 4.7, we have answered the question Q3 by proving that a small training error and a sufficiently large number of samples, as chosen in (4.17), suffice to ensure a small generalization error (and total error). Moreover, the number of samples only depends polynomially on the dimension. Therefore, it overcomes the *curse of dimensionality.*

5. Discussion

Physics informed neural networks (PINNs) are widely used in approximating both forward as well as inverse problems for PDEs. However, there is a paucity of rigorous theoretical results on PINNs that can explain their excellent empirical performance. In particular, one wishes to answer the questions Q1 (on the smallness of PINN residuals), Q2 (smallness of the total error) and Q3 (smallness of the generalization error if the training error is small) in order to provides rigorous guarantees for PINNs.

In this article, we aimed to address these theoretical questions rigorously. We do so within the context of the Kolmogorov equations, which are linear parabolic PDEs of the general form (2.1). The heat equation as well as the Black-Scholes equation of option pricing are prototypical examples of these PDEs. Moreover, these PDEs can be set in very high-dimensional spatial domains. Thus, in addition to providing rigorous bounds on the PINN generalization error and total error, we also aimed to investigate whether PINNs can overcome the curse of dimensionality in this context.

To this end, we answered question Q1 in Theorem 3.3, where we constructed a PINN (see Figure 1) for which the PINN residual (generalization error) can be made as small as possible. Our constuction relied on emulating Dynkin's formula (3.5). Under suitable assumptions on the initial data as well as on the underlying stochastic process (cf. (3.9) and Theorem 3.5), we are also able to prove that the size of the constructed only grew polynomially, in input spatial dimension. Thus, we were able to show that this PINN was able to overcome the curse of dimensionality in attaining as small a residual as desired.

Next, we answered question Q2 in Theorem 3.6 by leveraging the stability of Kolmogorov PDEs to bound the total error (in L^2) for PINNs in terms of the underlying generalization error.

Finally, question Q3 that required one to bound the generalization error in terms of the training error was answered by using an error decomposition, Lipschitz continuity of the underlying generalization and training error maps and concentration inequalities in Corollary 4.5, where we derived a bound on the generalization error in terms of the training error and for sufficiently many randomly chosen training samples (4.17). Moreover, the number of training samples only grew polynomially in the dimension, alleviating the curse of dimensionaly in this regard.

Although we do not present numerical experiments in this paper, we point the readers to [38] and the forthcoming paper [29], where a large number of numerical experiments for PINNs in approximating both forward and inverse problems for Kolmogorov type and related equations, are presented. In particular, these experiments reveal that PINNs overcome the curse of dimensionality in this context. These findings are consistent with our theoretical results.

At this stage, it is instructive to contrast our results with related works. As mentioned in the introduction, there are very few papers where PINNs are rigorously analyzed. When comparing to [36], we highlight that the fact that the authors of [36] used a special bespoke Hölder-type regularization term that penalized the gradients in their loss function. In practice, one trains PINNs in the L^2 (or L^1) setting and it is unclear how relevant the assumptions of [36] are in this context. On the other hand, we use the natural training paradigm for PINNs and prove rigorously that overall errors can be made small. Comparing with [26], we observe that the authors of [26] only address questions Q2 and (partially) Q3, but in a very general setting. It is not proved in [26] that the total error can be made small. We do so here. Moreover, we also provide the first bounds for PINNs, where the curse of dimensionality is alleviated.

It is an appropriate juncture to compare our results with a large number of articles demonstrating the alleviation of the curse of dimensionality for neural networks approximating Kolmogorov type PDEs, see [8, 3] and references therein. We would like to point out that these articles consider the *supervised learning* paradigm, where (possibly large amounts of) data needs to be provided to train the neural network for approximating solutions of PDEs. This data has to be generated by either expensive numerical simulations or the use of representation formulas such as the Feynman-Kac formulas, which requires solutions of underlying SDEs. In contrast, we recall that PINNs do not require *any data* in the interior of the domain and thus are very diferent in design and conception to supervised learning frameworks.

We would also like to highlight some limitations of our analysis. We showed in Theorem 3.3 that network size in approximating solutions of general Kolmogorov equations (2.1) depended on the rate of growth the quantity ρ_d, defined in (3.9). We were also able to prove in Theorem 3.5 that ρ_d only grew polynomially (in dimension) for a subclass of Kolmogorov PDEs. Extending these results to general Kolmogorov PDEs is an open question. Moreover, it is worth repeating (see Remark 4.6) that the constants in our estimates are clearly not optimal and might be significant overestimates, see [26] for a discussion on this issue.

Finally, we point out that although we focussed our results on the large and important class of Kolmogorov PDEs in this paper, the methods that we developed will be very useful in the analysis of PINNs for approximating PDEs. In particular, the use of smoothness of the underlying PDEs solutions and their approximation by Tanh neural networks (as in [5]), to build PINNs with small PDE residuals can be applied to a variety of linear and non-linear PDEs. Similarly, the error decomposition (4.2) and Theorem 4.2 (Corollary 4.3) are very general and can be used in many different contexts, to bound PINN generalization error by training error, for sufficiently many random training points. We plan to apply these techniques for the comprehensive error analysis of PINNs for approximating forward as well as inverse problems for PDEs in forthcoming papers.

References

[1] G. Bai, U. Koley, S. Mishra, and R. Molinaro. Physics informed neural networks (PINNs) for approximating nonlinear dispersive PDEs. *arXiv preprint arXiv:2104.05584*, 2021.

[2] A. Barth, A. Jentzen, A. Lang, and C. Schwab. *Numerical Analysis of Stochastic Ordinary Differential Equations.* ETH Zürich, 2018.

[3] J. Berner, P. Grohs, and A. Jentzen. Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of black–scholes partial differential equations. *SIAM Journal on Mathematics of Data Science*, 2(3):631–657, 2020.

[4] T. Chen and H. Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

[5] T. De Ryck, S. Lanthaler, and S. Mishra. On the approximation of functions by tanh neural networks, 2021.

[6] M. Dissanayake and N. Phan-Thien. Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, 1994.

[7] W. E, J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017.

[8] P. Grohs, F. Hornung, A. Jentzen, and P. Von Wurstemberger. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations. *arXiv preprint arXiv:1809.02362*, 2018.

[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(05):803–859, 2020.

[10] I. Gühring and M. Raslan. Approximation rates for neural networks with encodable weights in smoothness spaces. *Neural Networks*, 134:107–130, 2021.

[11] R. Hiptmair and C. Schwab. *Numerical Methods for Elliptic and Parabolic Boundary Value Problems.* ETH Zürich, 2008.

[12] A. D. Jagtap and G. E. Karniadakis. Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5):2002–2041, 2020.

[13] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365:113028, 2020.

[14] F. C. Klebaner. *Introduction to stochastic calculus with applications.* World Scientific Publishing Company, 2012.

[15] G. Kutyniok, P. Petersen, M. Raslan, and R. Schneider. A theoretical analysis of deep neural networks and parametric pdes. *Constructive Approximation*, pages 1–53, 2021.

[16] I. E. Lagaris, A. Likas, and P. G. D. Neural-network methods for bound- ary value problems with irregular boundaries. *IEEE Transactions on Neural Networks*, 11:1041–1049, 2000.

[17] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 2000.

[18] S. Lanthaler, S. Mishra, and G. E. Karniadakis. Error estimates for DeepOnets: A deep learning framework in infinite dimensions, 2021.

[19] P. Lévy and P. Lévy. *Théorie de l'addition des variables aléatoires*. Gauthier-Villars, 1954.

[20] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations, 2020.

[21] L. Lu, P. Jin, and G. E. Karniadakis. DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

[22] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021.

[23] K. O. Lye, S. Mishra, and D. Ray. Deep learning observables in computational fluid dynamics. *Journal of Computational Physics*, page 109339, 2020.

[24] K. O. Lye, S. Mishra, D. Ray, and P. Chandrashekar. Iterative surrogate model optimization (ISMO): An active learning algorithm for pde constrained optimization with deep neural networks. *Computer Methods in Applied Mechanics and Engineering*, 374:113575, 2021.

[25] Z. Mao, A. D. Jagtap, and G. E. Karniadakis. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, 2020.

[26] S. Mishra and R. Molinaro. Estimates on the generalization error of physics informed neural networks (PINNs) for approximating PDEs. *arXiv preprint arXiv:2006.16144*, 2020.

[27] S. Mishra and R. Molinaro. Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for pdes. *IMA Journal of Numerical Analysis*, 2021.

[28] S. Mishra and R. Molinaro. Physics informed neural networks for simulating radiative transfer. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 270:107705, 2021.

[29] S. Mishra, R. Molinaro, and R. Tanios. Physics informed neural networks for option pricing. *In preparation*, 2021.

[30] B. Øksendal. *Stochastic differential equations*. Springer, 2003.

[31] G. Pang, L. Lu, and G. E. Karniadakis. fPINNs: Fractional physics-informed neural networks. *SIAM journal of Scientific computing*, 41:A2603–A2626, 2019.

[32] M. Raissi and G. E. Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.

[33] 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*, 378:686–707, 2019.

[34] M. Raissi, A. Yazdani, and G. E. Karniadakis. Hidden fluid mechanics: A Navier-Stokes informed deep learning framework for assimilating flow visualization data. *arXiv preprint arXiv:1808.04327*, 2018.

[35] C. Schwab and J. Zech. Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in uq. *Analysis and Applications*, 17(01):19–55, 2019.

[36] Y. Shin, J. Darbon, and G. E. Karniadakis. On the convergence and generalization of physics informed neural networks. *arXiv preprint arXiv:2004.01806*, 2020.

[37] Y. Shin, Z. Zhang, and G. E. Karniadakis. Error estimates of residual minimization using neural networks for linear equations. *arXiv preprint arXiv:2010.08019*, 2020.

[38] R. Tanios. Physics informed neural networks in computational finance: high-dimensional forward and inverse option pricing. Master's thesis, ETH Zürich, 2021.

[39] L. Yang, X. Meng, and G. E. Karniadakis. B-PINNs: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

Appendix A. Additional material for Section 3

A.1. Auxiliary results.

Lemma A.1. *Let $p \in [2, \infty)$, $d, m \in \mathbb{N}$, let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space, and let $X_i : \Omega \to \mathbb{R}^d, i \in \{1, \dots, m\}$, be i.i.d. random variables with $\mathbb{E}\left[\|X_1\|\right] < \infty$. Then it holds that*

$$\left(\mathbb{E}\left[\left\| \mathbb{E}\left[X_1\right] - \frac{1}{m}\sum_{i=1}^{m} X_i \right\|^p \right] \right)^{1/p} \leq 2\sqrt{\frac{p-1}{m}} \left(\mathbb{E}\left[\left\| \mathbb{E}\left[X_1\right] - X_1 \right\|^p \right] \right)^{1/p}. \tag{A.1}$$

Proof. This result is [8, Corollary 2.5]. □

Lemma A.2. *Let $p \in [2, \infty)$, $q, m \in \mathbb{N}$, let $(\Omega, \mathcal{F}, \mathcal{P})$ and $(\mathcal{D}, \mathcal{A}, \mu)$ be probability spaces, and let for every $q \in \mathcal{D}$ the maps $X_i^q : \Omega \to \mathbb{R}, i \in \{1, \dots, m\}$, be i.i.d. random variables with $\mathbb{E}\left[\left|X_1^q\right|\right] < \infty$. Then it holds that*

$$\mathbb{E}\left[\left(\int_{\mathcal{D}} \left| \mathbb{E}\left[X_1^q\right] - \frac{1}{m}\sum_{i=1}^{m} X_i^q \right|^p \mu(dq) \right)^{1/p} \right] \leq 2\sqrt{\frac{p-1}{m}} \left(\int_{\mathcal{D}} \mathbb{E}\left[\left| \mathbb{E}\left[X_1^q\right] - X_1^q \right|^p \right] \mu(dq) \right)^{1/p}. \tag{A.2}$$

Proof. The proof involves Hölder's inequality, Fubini's theorem and Lemma A.1. The calculation is as in [8, eq. (226)]. □

Lemma A.3. *Let $\epsilon > 0$, let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space, and let $X : \Omega \to \mathbb{R}$ be a random variable that satisfies $\mathbb{E}\left[|X|\right] \leq \epsilon$. Then it holds that $\mathbb{P}(|X| \leq \epsilon) > 0$.*

Proof. This result is [8, Proposition 3.3]. □

Lemma A.4 (Lévy's modulus of continuity)**.** *For $(B_t)_{t\in[0,1]}$ a Brownian motion, it holds almost surely that*

$$\limsup_{h\downarrow 0} \sup_{0\leq t\leq 1-h} \frac{|B_{t+h} - B_t|}{\sqrt{2h\log\bigl(1/h\bigr)}} = 1. \tag{A.3}$$

Proof. This result is due to [19] and can be found in most probability theory textbooks. □

Lemma A.5. *Let $T > 0$, $p \geq 2$, $d, m \in \mathbb{N}$, let $(\Omega, \mathcal{F}, P, (\mathbb{F}_t)_{t\in[0,T]})$ be a stochastic basis and let $W : [0,T] \times \Omega \to \mathbb{R}^m$ be a standard m-dimensional Brownian motion on $(\Omega, \mathcal{F}, P, (\mathbb{F}_t)_{t\in[0,T]})$. Let $\lambda \in \mathcal{L}^p(P|_{\mathbb{F}_0}, \|\cdot\|_{\mathbb{R}^d})$ and let $\mu : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times m}$ be affine functions. Then there exists an up to indistinguishability unique $(\mathbb{F}_t)_{t\in[0,T]}$-adapted stochastic process $X^\lambda : [0,T] \times \Omega \to \mathbb{R}^d$, which satisfies*

(1) *that for all $t \in [0,T]$ it holds P-a.s. that*

$$X_t^\lambda = \lambda + \int_0^t \mu(X_s^\lambda)ds + \int_0^t \sigma(X_s^\lambda)dW_s \tag{A.4}$$

(2) *it holds that $\sup_{t\in[0,T]} \left\|X_t^\lambda\right\|_{\mathcal{L}^p(P,\|\cdot\|_{\mathbb{R}^d})} < \infty$,*

(3) *it holds that for all $\alpha \in (0, \frac{1}{2}]$ that*

$$\sup_{\substack{s,t\in[0,T],\\ s<t}} \frac{\left\|X_s^\lambda - X_t^\lambda\right\|_{\mathcal{L}^p(P,\|\cdot\|_{\mathbb{R}^d})}}{|s-t|^\alpha} < \infty, \tag{A.5}$$

(4) *for all $x \in \mathbb{R}^d$, $t \in [0,T]$ and $\omega \in \Omega$ it holds that*

$$X_t^x(\omega) = \sum_{i=1}^d \Bigl(X_t^{e_i}(\omega) - X_t^0(\omega)\Bigr) x_i + X_t^0(\omega). \tag{A.6}$$

Proof. Properties (1)-(3) are proven in [2, Theorem 4.5.1]. Property (4) follows from Lemma 2.20 in [8] and Lemma 3.3 in [3]. □

Lemma A.6. *Let $h : \mathbb{R} \to \mathbb{R} : x \mapsto \min\{1, \max\{0, x\}\}$. For every $N \geq 2$ and $\epsilon, \gamma > 0$ there exists a tanh neural network $\hat{h}$ with two hidden layers, $\mathcal{O}\left(N^{\frac{1}{2(1-\gamma)}} \epsilon^{\frac{-3}{1-\gamma}}\right)$ neurons and weights growing as $\mathcal{O}\left(N^{\frac{1}{(1-\gamma)}} \epsilon^{\frac{-6}{1-\gamma}}\right)$ such that*

$$\left\|h - \hat{h}\right\|_{L^\infty(\mathbb{R})} \leq \epsilon, \quad \left\|h' - \hat{h}'\right\|_{L^2([-N,N])} \leq \epsilon \quad \text{and} \quad \left\|\hat{h}'\right\|_{L^\infty(\mathbb{R})} \leq 2. \tag{A.7}$$

Proof. We first approximate h with a function $\tilde{h}$ that is twice continuously differentiable,

$$\tilde{h}(x) = \begin{cases} 0 & x \leq -\frac{\pi\epsilon^2}{2}, \\ \frac{1}{2}\left(\frac{\pi\epsilon^2}{2} + x - \epsilon^2\cos\bigl(\frac{x}{\epsilon^2}\bigr)\right) & -\frac{\pi\epsilon^2}{2} < x \leq \frac{\pi\epsilon^2}{2}, \\ x & \frac{\pi\epsilon^2}{2} < x \leq 1 - \frac{\pi\epsilon^2}{2}, \\ \frac{1}{2}\left(1 - \frac{\pi\epsilon^2}{2} + x + \epsilon^2\cos\bigl(\frac{1-x}{\epsilon^2}\bigr)\right) & 1 - \frac{\pi\epsilon^2}{2} < x \leq 1 + \frac{\pi\epsilon^2}{2}, \\ 1 & 1 + \frac{\pi\epsilon^2}{2} < x. \end{cases} \tag{A.8}$$

It is easy to prove that $\left\|h - \tilde{h}\right\|_{L^\infty(\mathbb{R})} = \mathcal{O}(\epsilon^2)$. Next, we calculate the derivative of $\tilde{h}$,

$$\tilde{h}'(x) = \begin{cases} 0 & x \leq -\frac{\pi\epsilon^2}{2}, \\ \frac{1}{2}\left(1 + \sin\bigl(\frac{x}{\epsilon^2}\bigr)\right) & -\frac{\pi\epsilon^2}{2} < x \leq \frac{\pi\epsilon^2}{2}, \\ 1 & \frac{\pi\epsilon^2}{2} < x \leq 1 - \frac{\pi\epsilon^2}{2}, \\ \frac{1}{2}\left(1 + \sin\bigl(\frac{1-x}{\epsilon^2}\bigr)\right) & 1 - \frac{\pi\epsilon^2}{2} < x \leq 1 + \frac{\pi\epsilon^2}{2}, \\ 0 & 1 + \frac{\pi\epsilon^2}{2} < x. \end{cases} \tag{A.9}$$

A straightforward calculation leads to the bound $\left\|h' - \tilde{h}'\right\|_{L^2(\mathbb{R})} = \mathcal{O}(\epsilon)$. Finally, one can easily check that $\tilde{h}''$ is continuous and that $\left\|\tilde{h}''\right\|_{L^\infty(\mathbb{R})} = \mathcal{O}(\epsilon^{-2})$. An application of [5, Theorem 5.1] on $\tilde{h}$ gives us for every $\gamma > 0$ and N large enough the existence of a tanh neural network $\hat{h}^{\mathcal{N}}$ with two hidden layers and $\mathcal{O}(\mathcal{N})$ neurons for which it holds that $\left\|\tilde{h} - \hat{h}^{\mathcal{N}}\right\|_{W^{1,\infty}([-1,2])} = \mathcal{O}(N^{-1+\gamma}\epsilon^{-2})$. Because of the nature of the construction of $\hat{h}^{\mathcal{N}}$, the monotonous behaviour of the hyperbolic tangent towards infinity and the fact that $\tilde{h}$ is constant outside $[-1, 2]$, the stronger result that $\left\|\tilde{h} - \hat{h}^{\mathcal{N}}\right\|_{W^{1,\infty}(\mathbb{R})} = \mathcal{O}(\mathcal{N}^{-1+\gamma}\epsilon^{-2})$ holds automatically as well. As a result we find that $\left\|(\hat{h}^{\mathcal{N}})'\right\|_{L^\infty(\mathbb{R})} \leq 2$, $\left\|\tilde{h} - \hat{h}^{\mathcal{N}}\right\|_{L^\infty(\mathbb{R})} = \mathcal{O}(\mathcal{N}^{-1+\gamma}\epsilon^{-2})$ and $\left\|\tilde{h} - \hat{h}^{\mathcal{N}}\right\|_{L^2([-N,N])} = \mathcal{O}(\sqrt{N}\mathcal{N}^{-1+\gamma}\epsilon^{-2})$. If we choose $\mathcal{N} \sim N^{\frac{1}{2(1-\gamma)}}\epsilon^{\frac{-3}{1-\gamma}}$ then we find that

$$\left\|\tilde{h} - \hat{h}^{\mathcal{N}}\right\|_{L^\infty(\mathbb{R})} \leq \epsilon \quad \text{and} \quad \left\|\tilde{h}' - (\hat{h}^{\mathcal{N}})'\right\|_{L^2([-N,N])} \leq \epsilon. \tag{A.10}$$

Moreover, [5, Theorem 5.1] tells us that the weights of $\hat{h}^{\mathcal{N}}$ grow as $\mathcal{O}(\mathcal{N}^2) = \mathcal{O}\left(N^{\frac{1}{(1-\gamma)}}\epsilon^{\frac{-6}{1-\gamma}}\right)$. The statement then follows from applying the triangle inequality. □

Appendix B. Lipschitz continuity in the parameter vector of a neural network and its derivatives

In this section we will prove that for any $x \in D$, a neural network and its corresponding Jacobian and Hessian matrix are Lipschitz continuous in the parameter vector. This property is of crucial importance to find bounds on the generalization error of physics informed neural networks, cf. Section 4. We first introduce some notation and then state or results. The main results of this section are Lemma B.3 and Lemma B.5.

We denote by $\sigma : \mathbb{R} \to \mathbb{R}$ be an (at least) twice continuously differentiable activation function, like tanh or sigmoid. For any $n \in \mathbb{N}$, we write for $x \in \mathbb{R}^n$ that $\sigma(x) := (\sigma(x_1), \ldots, \sigma(x_n))$. We use the definition of a neural network as in Definition 2.1.

Recall that for a differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$ the Jacobian matrix $J[f]$ is defined by

$$J[f]_{ij} = \frac{\partial f_i}{\partial x_j} \in \mathbb{R}^{m\times n}. \tag{B.1}$$

For our purpose, we make the following the following convention. For any $1 \leq k \leq L$, we define

$$J_k^\theta(x) := J[f_k^\theta]\left((f_{k-1}^\theta \circ \cdots \circ f_1^\theta)(x)\right) \in \mathbb{R}^{l_k \times l_{k-1}}. \tag{B.2}$$

Similarly, for a twice differentiable function $g : \mathbb{R}^n \to \mathbb{R}$ the Hessian matrix is defined by

$$H[g]_{ij} = \frac{\partial^2 g}{\partial x_i \partial x_j}. \tag{B.3}$$

Slightly abusing notation, we generalize this to vector-valued functions $g : \mathbb{R}^n \to \mathbb{R}^m$. We write

$$H[g]_{kij} = \frac{\partial^2 g_k}{\partial x_i \partial x_j}, \tag{B.4}$$

where we identify $\mathbb{R}^{1\times n\times n}$ with $\mathbb{R}^{n\times n}$ to make the definitions consistent. Similarly, if $v \in \mathbb{R}^{1\times m}$, then $v \cdot H[g]$ should be interpreted as

$$v \cdot H[g](x) := \sum_{k=1}^m v_k H[g_k](x) \in \mathbb{R}^{n\times n}. \tag{B.5}$$

For any $1 \leq k < L$, we write

$$H_k^\theta(x) := H[f_k^\theta]\left((f_{k-1}^\theta \circ \cdots \circ f_1^\theta)(x)\right) \in \mathbb{R}^{l_k \times l_{k-1} \times l_{k-1}}. \tag{B.6}$$

Finally, we will use the notation $J^\theta := J[\Psi^\theta]$ and $H^\theta := H[\Psi^\theta]$. The following lemma presents a generalized version of the chain rule.

Lemma B.1. *Let $f : \mathbb{R}^n \to \mathbb{R}^m$ and $g : \mathbb{R}^m \to \mathbb{R}$. Then it holds that*

$$H[g \circ f](x) := J[f](x)^T \cdot H[g](f(x)) \cdot J[f](x) + J[g](f(x)) \cdot H[f](x). \tag{B.7}$$

We now apply this formula to find an expression for H^θ in terms of J_k^θ and H_k^θ.

Lemma B.2. *It holds that*

$$J[\Psi^\theta] = \prod_{k=0}^{L-1} J^\theta_{L-k} \quad \text{and} \quad H[\Psi^\theta] = \sum_{k=1}^{L} (J^\theta_1)^T \cdots (J^\theta_{k-1})^T \cdot \left(J^\theta_L \cdots J^\theta_{k+1} \cdot H^\theta_k\right) \cdot J^\theta_{k-1} \cdots J^\theta_1. \tag{B.8}$$

Proof. The first statement is just the chain rule for calculating the derivative of a composite function. We prove the second statement using induction. For the base step, let $L = 1$. Then $\Psi^\theta = f^\theta_L$ and we have $H[\Psi^\theta] = H^\theta_L$. For the induction step, take $K \in \mathbb{N}, K \geq 2$ and assume that the statement holds for $L = K - 1$. Now let $\Phi^\theta = f^\theta_K \circ \cdots \circ f^\theta_2$ and $\Psi^\theta = \Phi^\theta \circ f^\theta_1$. Applying the generalized chain rule to calculate $H[\Phi^\theta \circ f^\theta_1]$ and using the induction hypothesis on $H[\Phi^\theta]$ gives the wanted result. □

Next, we formally introduce the element-wise supremum norm $|\cdot|_\infty$. Let $N \in \mathbb{N}$, $n_0, \ldots n_N \in \mathbb{N}$ and $A \in \mathbb{R}^{n_1 \times \cdots \times n_N}$. Then we define

$$|A|_\infty := \max_{1 \leq i_1 \leq n_1} \cdots \max_{1 \leq i_N \leq n_N} |A_{i_1 \cdots i_N}|. \tag{B.9}$$

Let $R > 0$ and suppose that $A_i \in \mathbb{R}^{n_{i-1} \times n_i}$. Then it holds that

$$\left|\prod_{i=1}^{N} A_i\right|_\infty \leq |A_N|_\infty \prod_{i=1}^{N-1} n_i |A_i|_\infty. \tag{B.10}$$

Moreover, for $v \in \mathbb{R}^{1 \times a}$ and $A \in \mathbb{R}^{a \times b \times c}$ it holds that $|v \cdot A|_\infty \leq a|v|_\infty |A|_\infty$.

The following lemma states that the output of each layer of a neural network is Lipschitz continuous in the parameter vector for any input $x \in [a, b]^d$. The lemma is stated for neural networks with a differentiable activation function, but can be easily adapted for e.g. ReLU neural networks.

Lemma B.3. *Let $d, L, W \in \mathbb{N}$ with $L, W \geq 2$, $a, b \in \mathbb{R}$ with $a < b$ and $R \geq 1$. Moreover, let $\theta, \vartheta \in \Theta_{L,W,R}$, $\alpha = \max\{1, |a|, |b|, \|\sigma\|_\infty\}$ and $\beta = \max\{1, \|\sigma'\|_\infty\}$. Then it holds for $1 \leq K \leq L$ that*

$$\left\| f^\theta_K \circ \cdots \circ f^\theta_1 - f^\vartheta_K \circ \cdots \circ f^\vartheta_1 \right\|_{L^\infty([a,b]^d)} \leq \alpha(d+4) W^{K-1} R^{K-1} \beta^K |\theta - \vartheta|_\infty. \tag{B.11}$$

Proof. Let $l_0, \ldots, l_L$ denote the widths of the neural network, where $l_0 = d$. Let $x \in [a, b]^d$ be arbitrary. First of all, it holds that

$$\begin{aligned} \left|f^\theta_1(x) - f^\vartheta_1(x)\right|_\infty &= \left|\sigma(W^\theta_1 x + b^\theta_1) - \sigma(W^\vartheta_1 x + b^\vartheta_1)\right|_\infty \\ &\leq \|\sigma'\|_\infty \left|(W^\theta_1 - W^\vartheta_1)x + (b^\theta_1 - b^\vartheta_1)\right|_\infty \\ &\leq \beta(d\alpha + 1)|\theta - \vartheta|_\infty. \end{aligned} \tag{B.12}$$

Now let $2 \leq k \leq L$ and define $y = (f^\theta_{k-1} \circ \cdots \circ f^\theta_1)(x)$ and $\tilde{y} = (f^\vartheta_{k-1} \circ \cdots \circ f^\vartheta_1)(x)$. We find that

$$\begin{aligned} \left|f^\theta_k(y) - f^\vartheta_k(\tilde{y})\right|_\infty &\leq \max\{1, \|\sigma'\|_\infty\} \left|(W^\theta_k - W^\vartheta_k)y + b^\theta_k - b^\vartheta_k + W^\vartheta_k(y - \tilde{y})\right|_\infty \\ &\leq \beta((l_{k-1}\alpha + 1)|\theta - \vartheta|_\infty + l_{k-1}R|y - \tilde{y}|_\infty). \end{aligned} \tag{B.13}$$

A recursive application of this inequality then gives us for $1 \leq K \leq L$ that

$$\begin{aligned} &\left\| f^\theta_K \circ f^\theta_{K-1} \circ \cdots \circ f^\theta_1 - f^\vartheta_K \circ f^\vartheta_{K-1} \circ \cdots \circ f^\vartheta_1 \right\|_\infty \\ &\leq \sum_{k=1}^{K} l_{K-1} \cdots l_k (l_{k-1}\alpha + 1) R^{K-k} \beta^{K-k+1} |\theta - \vartheta|_\infty \\ &\leq W^{K-1}(d\alpha + 1) R^{K-1} \beta^K |\theta - \vartheta|_\infty + \beta(W\alpha + 1)|\theta - \vartheta|_\infty \sum_{k=2}^{K} W^{K-k} R^{K-k} \beta^{K-k} \\ &\leq W^{K-1}(d\alpha + 1) R^{K-1} \beta^K |\theta - \vartheta|_\infty + \frac{\beta(W\alpha + 1) W^{K-1} R^{K-1} \beta^{K-1}}{WR\beta - 1} |\theta - \vartheta|_\infty \\ &\leq \alpha(d+4) W^{K-1} R^{K-1} \beta^K |\theta - \vartheta|_\infty, \end{aligned} \tag{B.14}$$

where we used that $\beta(W\alpha + 1)/(WR\beta - 1) \leq \beta(2\alpha + 1)/(2R\beta - 1) \leq 3\alpha$ when $W \geq 2, R \geq 1, \alpha \geq 1$. □

Lemma B.4. *Let $d, L, W \in \mathbb{N}$ with $L, W \geq 2$, $a, b \in \mathbb{R}$ with $a < b$ and $R \geq 1$. Moreover, let $\theta, \vartheta \in \Theta_{L,W,R}$, $\alpha = \max\{1, |a|, |b|, \|\sigma\|_\infty\}$ and $\beta = \max\{1, \|\sigma'\|_\infty, \|\sigma''\|_\infty, \|\sigma'''\|_\infty\}$. Then it holds for all $1 \leq k \leq L$ and $x \in [a,b]^d$ that*

$$\left|J_k^\theta(x)_i - J_k^\vartheta(x)_i\right|_\infty \leq \beta(1 + \alpha(d+4)W^{k-1}R^k\beta^{k-1} + R(\alpha W + 1))|\theta - \vartheta|_\infty \quad \textit{and} \tag{B.15}$$

$$\left|H_k^\theta(x)_i - H_k^\vartheta(x)_i\right|_\infty \leq 2\beta R(1 + \alpha(d+4)W^{k-1}R^k\beta^{k-1} + R(\alpha W + 1))|\theta - \vartheta|_\infty. \tag{B.16}$$

Proof. Let w_i^T be the i-th row of $W^{\theta,k}$, let $\tilde{w}_i^T$ be the i-th row of $W^{\vartheta,k}$ and set $b := b^{\theta,k}$ and $\tilde{b} := b^{\vartheta,k}$. Let $F = f_{k-1}^\theta \circ \cdots \circ f_1^\theta$ and $\tilde{F} = f_{k-1}^\vartheta \circ \cdots \circ f_1^\vartheta$. For $1 \leq i \leq l_k$, we have that

$$J_k^\theta(x)_i = \sigma'(w_i^T \cdot F(x) + b_i) \cdot w_i^T \in \mathbb{R}^{1 \times l_{k-1}} \tag{B.17}$$

$$H_k^\theta(x)_i = \sigma''(w_i^T \cdot F(x) + b_i) \cdot w_i \cdot w_i^T \in \mathbb{R}^{l_{k-1} \times l_{k-1}} \tag{B.18}$$

and analogously for $J_k^\vartheta(x)_i$ and $H_k^\vartheta(x)_i$. The triangle inequality and the Lipschitz continuity of σ' gives us that

$$\begin{aligned}\left|J_k^\theta(x)_i - J_k^\vartheta(x)_i\right|_\infty &\leq \|\sigma'\|_\infty |w_i - \tilde{w}_i|_\infty + \left|\sigma'(w_i^T \cdot F(x) + b_i) - \sigma'(\tilde{w}_i^T \cdot \tilde{F}(x) + \tilde{b}_i)\right| |\tilde{w}_i|_\infty \\ &\leq \beta|\theta - \vartheta|_\infty + \|\sigma''\|_\infty R\left|w_i^T \cdot (F(x) - \tilde{F}(x)) + (w_i - \tilde{w}_i)^T \cdot \tilde{F}(x) + b_i - \tilde{b}_i\right| \\ &\leq \beta|\theta - \vartheta|_\infty + \|\sigma''\|_\infty R\left(l_{k-1}R\left|F(x) - \tilde{F}(x)\right|_\infty + (l_{k-1}\|\sigma\|_\infty + 1)|\theta - \vartheta|_\infty\right).\end{aligned} \tag{B.19}$$

Using that $\left|F(x) - \tilde{F}(x)\right|_\infty \leq \alpha(d+4)W^{k-2}R^{k-2}\beta^{k-1}|\theta - \vartheta|_\infty$ (Lemma B.3) for $k \geq 2$ and $l_{k-1} \leq W$, we get

$$\left|J_k^\theta(x)_i - J_k^\vartheta(x)_i\right|_\infty \leq \beta(1 + \alpha(d+4)W^{k-1}R^k\beta^{k-1} + R(\alpha W + 1))|\theta - \vartheta|_\infty \tag{B.20}$$

for $k \geq 2$. One can check that the inequality also holds for $k = 1$.

For the Hessian matrix, the triangle inequality and the Lipschitz continuity of σ'' gives us that

$$\begin{aligned}\left|H_k^\theta(x)_i - H_k^\vartheta(x)_i\right|_\infty \leq& \|\sigma''\|_\infty \left|w_i \cdot w_i^T - \tilde{w}_i \cdot \tilde{w}_i^T\right|_\infty + \left|\sigma''(w_i^T \cdot F(x) + b_i) - \sigma''(\tilde{w}_i^T \cdot \tilde{F}(x) + \tilde{b}_i)\right| \left|\tilde{w}_i \cdot \tilde{w}_i^T\right|_\infty \\ \leq& 2\beta R|\theta - \vartheta|_\infty + \|\sigma'''\|_\infty R^2(\alpha W + 1)|\theta - \vartheta|_\infty + \|\sigma'''\|_\infty R^3 W\left|F(x) - \tilde{F}(x)\right|_\infty\end{aligned} \tag{B.21}$$

Using Lemma B.3 again, we get

$$\left|H_k^\theta(x)_i - H_k^\vartheta(x)_i\right|_\infty \leq 2\beta R(1 + \alpha(d+4)W^{k-1}R^k\beta^{k-1} + R(\alpha W + 1))|\theta - \vartheta|_\infty \tag{B.22}$$

for $k \geq 2$. One can check that the inequality also holds for $k = 1$. □

The following lemma states that the Jacobian and Hessian matrix of a neural network are Lipschitz continuous in the parameter vector for any input $x \in [a,b]^d$.

Lemma B.5. *Let $d, L, W \in \mathbb{N}$ with $L, W \geq 2$, $a, b \in \mathbb{R}$ with $a < b$ and $R \geq 1$. Moreover, let $\theta, \vartheta \in \Theta_{L,W,R}$, $\alpha = \max\{1, |a|, |b|, \|\sigma\|_\infty\}$ and $\beta = \max\{1, \|\sigma'\|_\infty, \|\sigma''\|_\infty, \|\sigma'''\|_\infty\}$. Then it holds that for all $x \in [a,b]^d$ that*

$$\left|J[\Psi^\theta](x) - J[\Psi^\vartheta](x)\right|_\infty \leq 2\alpha(d+7)LR^{2L-1}W^{2L-2}\beta^{L-1}|\theta - \vartheta|_\infty, \tag{B.23}$$

$$\left|H[\Psi^\theta](x) - H[\Psi^\vartheta](x)\right|_\infty \leq 4\alpha(d+7)L^2R^{3L-1}W^{3L-3}\beta^L|\theta - \vartheta|_\infty. \tag{B.24}$$

Proof. We will prove the formulas by repeatedly using the triangle inequality and using the representations proven in Lemma B.2. To do so, we need to introduce some notation. Define for $0 \leq l \leq L + k - 1$ the object $\phi^l \in \{\theta, \vartheta\}^{2L}$ such that

$$\phi_j^l = \begin{cases} \vartheta & j \leq l, \\ \theta & j > l. \end{cases} \quad \text{and} \quad A_j^{k,l} = \begin{cases} (J_j^{\phi_j^l})^T & 1 \leq j \leq k-1, \\ J_{L+k-j}^{\phi_k^l} & k \leq j \leq L-1 \\ H_k^{\phi_L^l} & j = L \\ J_{L+k-j}^{\phi_k^l} & L+1 \leq j \leq L+k-1. \end{cases} \tag{B.25}$$

In particular, $\phi_j^{k,0} = \theta$ and $\phi_j^{k,L+k-1} = \vartheta$ for all j. To simplify notation, we write

$$h_k^l = (J_1^{\phi_1^l})^T \cdots (J_{k-1}^{\phi_{k-1}^l})^T \cdot \left(J_L^{\phi_k^l} \cdots J_{k+1}^{\phi_{L-1}^l} \cdot H_k^{\phi_L^l} \right) \cdot J_{k-1}^{\phi_{L+1}^l} \cdots J_1^{\phi_{L+k-1}^l} = \prod_{j=1}^{L+k-1} A_j^{k,l}. \tag{B.26}$$

The triangle inequality and Lemma B.2 then give that

$$\left| H^\theta - H^\vartheta \right|_\infty \leq \sum_{k=1}^{L} \sum_{l=1}^{L+k-1} \left| h_k^{l-1} - h_k^l \right|_\infty. \tag{B.27}$$

Observe that $A_j^{k,l-1} - A_j^{k,l} = 0$ for $j \neq l$. Therefore

$$\begin{aligned} \left| h_k^{l-1} - h_k^l \right|_\infty &= \left| A_1^{k,l} \cdots A_{l-1}^{k,l} \cdot (A_l^{k,l-1} - A_l^{k,l}) \cdot A_{l+1}^{k,l} \cdots A_{L+k-1}^{k,l} \right|_\infty \\ &\leq (l_1 \cdots l_{k-1})^2 \cdot l_k \cdots l_{L-1} \cdot R^{L+k-2} \left| A_l^{k,l-1} - A_l^{k,l} \right|_\infty \\ &\leq W^{L+k-2} R^{L+k-2} \left| A_l^{k,l-1} - A_l^{k,l} \right|_\infty. \end{aligned} \tag{B.28}$$

From Lemma B.4, it follows that

$$\left| A_l^{k,l-1} - A_l^{k,l} \right|_\infty \leq 2\beta R (1 + \alpha(d+4) W^{k-1} R^k \beta^{k-1} + R(\alpha W + 1)) |\theta - \vartheta|_\infty \tag{B.29}$$

Writing $\gamma := 1 + R(\alpha W + 1)$ we get

$$\begin{aligned} \left| H^\theta - H^\vartheta \right|_\infty &\leq \sum_{k=1}^{L} (L+k-1) W^{L+k-2} R^{L+k-2} \cdot 2\beta R (1 + \alpha(d+4) W^{k-1} R^k \beta^{k-1} + R(\alpha W + 1)) |\theta - \vartheta|_\infty \\ &\leq \sum_{k=1}^{L} 2L W^{2L-2} R^{2L-2} \cdot 2\beta R \alpha (d+7) W^{L-1} R^L \beta^{L-1} |\theta - \vartheta|_\infty \\ &\leq 4\alpha(d+7) L^2 R^{3L-1} W^{3L-3} \beta^L |\theta - \vartheta|_\infty. \end{aligned} \tag{B.30}$$

In an entirely similar fashion we obtain

$$\left| J^\theta - J^\vartheta \right|_\infty \leq \sum_{k=1}^{L} W^{L-1} R^{L-1} \left| J_k^\theta - J_k^\vartheta \right|_\infty \leq 2\alpha(d+7) L R^{2L-1} W^{2L-2} \beta^{L-1} |\theta - \vartheta|_\infty. \tag{B.31}$$

□

Appendix C. Additional material for Section 4

Lemma C.1 (Hoeffding's inequality). *Let $\epsilon, c > 0$, $N \in \mathbb{N}$, let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space and let $X_n : \Omega \to [0, c]$ be independent random variables. Then it holds that*

$$\mathbb{P}\left(\frac{1}{N} \left(\sum_{i=1}^{N} (X_i - \mathbb{E}[X_i]) \right) \geq \epsilon \right) \leq \exp\left(\frac{-2\epsilon^2 N}{c^2} \right). \tag{C.1}$$

Lemma C.2. *Let $x \in \mathbb{R}$ and $\sigma(x) = \tanh x = \frac{e^{-x} - e^x}{e^{-x} + e^x}$. It holds that $\sigma'(x) = 1 - (\sigma(x))^2$ and $\sigma''(x) = -2\sigma(x)/(1 - (\sigma(x))^2)$. In addition, it holds that $\left\|\sigma'\right\|_\infty = 1$ and $\left\|\sigma''\right\|_\infty = 4/3\sqrt{3} \leq 1$ and $\left\|\sigma'''\right\|_\infty = 2$.*

The following lemma provides estimate on the various PINN residuals. It is based on the fact that neural networks and their derivatives are Lipschitz continuous in the parameter vector, the proof of which can be found in Appendix B.

Lemma C.3. *Let $d, L, W \in \mathbb{N}$, $R \geq 1$, $a, b \in \mathbb{R}$ with $a < b$ and let $u_\theta : [a,b]^d \to \mathbb{R}$, $\theta \in \Theta$, be tanh neural networks with smooth activation function σ, at most $L-1$ hidden layers, width at most W and weights and biases bounded by R. Let the PINN generalization $\mathcal{E}_G^q$ and training $\mathcal{E}_T^q$ errors be defined as in Section 2.3 for linear Kolmogorov PDEs (cf. Section 2.1). Let $\alpha = \max\{1, |a|, |b|, \|\sigma\|_\infty\}$ and $\beta = \max\{1, \left\|\sigma'\right\|_\infty, \left\|\sigma''\right\|_\infty, \left\|\sigma'''\right\|_\infty\}$ and assume that $\max\{\|\varphi\|_\infty, \|\psi\|_\infty\} \leq \max_{\theta \in \Theta} \|u_\theta\|_\infty$. Let $\mathfrak{L}_Q^q$ denote the Lipschitz constant of $\mathcal{E}_Q^q$, for $q = i, t, s$ and $Q = G, T$. Then it holds that*

$$\mathfrak{L}_Q^q \leq 2^5 \max_{x \in D} \left(1 + \sum_{i=1}^{d} \left| \mu(x)_i \right| + \sum_{i,j=1}^{d} \left| (\sigma(x)\sigma(x)^*)_{ij} \right| \right)^2 (d+7)^2 L^4 R^{6L-1} W^{6L-6} \beta^{2L}. \tag{C.2}$$

Proof. Without loss of generality, we only focus on $\mathcal{E}_G^q$, for $q = i, s, t$. We see for $q = i, t, s$

$$\left|\mathcal{E}_G^q(\theta) - \mathcal{E}_T^q(\vartheta)\right|_\infty \le 2 \max_\theta \left\|\mathcal{R}_q[u_\theta]\right\|_\infty \left\|\mathcal{R}_q[u_\theta] - \mathcal{R}_q[\Phi^\vartheta]\right\|_\infty \tag{C.3}$$

For $q = t, s$ and $(x, t) \in D \times [0, T]$, it follows from Lemma B.3 that

$$\left|\mathcal{R}_q[u_\theta](x, t) - \mathcal{R}_q[\Phi^\vartheta](t, x)\right| \le (d + 4) W^{L-1} R^{L-1} |\theta - \vartheta|_\infty, \tag{C.4}$$

and similarly using Lemma B.5 that

$$\begin{aligned} \left|\mathcal{R}_i[u_\theta](t, x) - \mathcal{R}_i[\Phi^\vartheta](t, x)\right| &\le (1 + \left|\mu(x)\right|_1) \left|J^\theta - J^\vartheta\right|_\infty + \left|\sigma(x)\sigma(x)^*\right|_1 \left|H_x^\theta - H_x^\vartheta\right|_\infty \\ &\le 4\alpha(1 + \left|\mu(x)\right|_1 + \left|\sigma(x)\sigma(x)^*\right|_1)(d + 7) L^2 R^{3L-1} W^{3L-3} \beta^L |\theta - \vartheta|_\infty, \end{aligned} \tag{C.5}$$

where we let $\left|\cdot\right|_p$ denote the vector p-norm of the vectorized version of a general tensor (cf. (B.9)). Next, we calculate using again Lemma B.5 (by setting $\vartheta = 0$) and $\max\{\|\varphi\|_\infty, \|\psi\|_\infty\} \le \max_{\theta\in\Theta} \|u_\theta\|_\infty$ for $q = t, s$ that

$$\max_\theta \left\|\mathcal{R}_i[u_\theta]\right\|_\infty \le 4\alpha C(d + 7) L^2 R^{3L} W^{3L-3} \beta^L, \quad \max_\theta \left\|\mathcal{R}_q[u_\theta]\right\|_\infty \le 2WR, \tag{C.6}$$

where $C = \max_{x\in D}(1 + \left|\mu(x)\right|_1 + \left|\sigma(x)\sigma(x)^*\right|_1)$. Combining all the previous results prove the stated bound. □