

# 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\*

Julius Berner<sup>†</sup>, Philipp Grohs<sup>‡</sup>, and Arnulf Jentzen<sup>§</sup>

**Abstract.** The development of new classification and regression algorithms based on empirical risk minimization (ERM) over deep neural network hypothesis classes, coined *deep learning*, revolutionized the area of artificial intelligence, machine learning, and data analysis. In particular, these methods have been applied to the numerical solution of high-dimensional partial differential equations with great success. Recent simulations indicate that deep learning-based algorithms are capable of overcoming the curse of dimensionality for the numerical solution of Kolmogorov equations, which are widely used in models from engineering, finance, and the natural sciences. The present paper considers under which conditions ERM over a deep neural network hypothesis class approximates the solution of a  $d$ -dimensional Kolmogorov equation with affine drift and diffusion coefficients and typical initial values arising from problems in computational finance up to error  $\varepsilon$ . We establish that, with high probability over draws of training samples, such an approximation can be achieved with both the size of the hypothesis class and the number of training samples scaling only polynomially in  $d$  and  $\varepsilon^{-1}$ . It can be concluded that ERM over deep neural network hypothesis classes overcomes the curse of dimensionality for the numerical solution of linear Kolmogorov equations with affine coefficients.

**Key words.** deep learning, curse of dimensionality, Kolmogorov equation, generalization error, empirical risk minimization

**AMS subject classifications.** 60H30, 65C30, 62M45, 68T05

**DOI.** 10.1137/19M125649X

**1. Introduction.** In this introductory section we want to present and motivate our problem, provide the reader with background knowledge and references to previous research on the topic, and outline the important steps, as well as possible extensions, of our contribution.

**1.1. Problem statement.** Suppose we need to numerically approximate the end value<sup>1</sup>  $F_d(T, \cdot)$  at time  $T \in (0, \infty)$  of the solution  $F_d \in \mathcal{C}([0, T] \times \mathbb{R}^d, \mathbb{R})$  of a *linear Kolmogorov equation* which for an initial value  $\varphi_d \in \mathcal{C}(\mathbb{R}^d, \mathbb{R})$ , drift coefficient  $\mu_d \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$ , and diffusion

\*Received by the editors April 16, 2019; accepted for publication (in revised form) March 25, 2020; published electronically July 28, 2020.

<https://doi.org/10.1137/19M125649X>

**Funding:** This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2044-390685587, Mathematics Münster: Dynamics-Geometry-Structure, and by the Austrian Science Fund (FWF) under grant I3403-N32.

<sup>†</sup>Faculty of Mathematics, University of Vienna, Vienna, 1090, Austria ([julius.berner@univie.ac.at](mailto:julius.berner@univie.ac.at)).

<sup>‡</sup>Faculty of Mathematics and Research Platform DataScience@UniVienna, University of Vienna, Vienna, 1090, Austria ([philipp.grohs@univie.ac.at](mailto:philipp.grohs@univie.ac.at)).

<sup>§</sup>Department of Mathematics, ETH Zürich, Switzerland, and Faculty of Mathematics and Computer Science, University of Münster, Münster, 48149, Germany ([ajentzen@uni-muenster.de](mailto:ajentzen@uni-muenster.de)).

<sup>1</sup>We write the subscript  $d$  as we are interested in approximation rates w.r.t. to the dimension  $d \in \mathbb{N}$ .

coefficient  $\sigma_d \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^{d \times d})$  is defined as<sup>2</sup>

$$(1.1) \quad \begin{cases} \frac{\partial F_d}{\partial t}(t, x) = \frac{1}{2} \text{Trace}(\sigma_d(x)[\sigma_d(x)]^*(\text{Hess}_x F_d)(t, x)) + \mu_d(x) \cdot (\nabla_x F_d)(t, x), \\ F_d(0, x) = \varphi_d(x) \end{cases}$$

for every  $(t, x) \in [0, T] \times \mathbb{R}^d$ . Important special cases include the heat equation or the Black–Scholes equation from computational finance. For the latter partial differential equation (PDE) typically the coefficients  $\sigma_d$  and  $\mu_d$  are affine and the initial value  $\varphi_d$  can be represented as a composition of minima, maxima, and linear combinations such as

$$(1.2) \quad \varphi_d(x) = \min \left\{ \max \{D - c_d \cdot x, 0\}, D \right\}$$

with suitable coefficients  $D \in (0, \infty)$ ,  $c_d \in \mathbb{R}^d$  in the case of a European put option pricing problem. It is well known that standard numerical methods for solving PDEs, in particular those based on a discretization of the domain, suffer from the curse of dimensionality, meaning that their computational complexity grows exponentially in dimension  $d$  [3, 56].

If the goal is simply to evaluate  $F_d(T, \cdot)$  at a single value  $\xi \in \mathbb{R}^d$ , then under suitable assumptions Monte Carlo sampling methods are capable of overcoming the curse of dimensionality. These methods are based on the integral representation (Feynman–Kac formula)

$$(1.3) \quad F_d(T, \xi) = \mathbb{E}[\varphi_d(S_T^\xi)],$$

where  $(S_t^\xi)_{t \in [0, T]}$  is a stochastic process satisfying the stochastic differential equation (SDE)

$$dS_t^\xi = \sigma_d(S_t^\xi) dB_t^d + \mu_d(S_t^\xi) dt \quad \text{and} \quad S_0^\xi = \xi$$

for a  $d$ -dimensional  $(\mathcal{G}_t)$ -Brownian motion  $B^d$  on some filtered probability space  $(\Omega, \mathcal{G}, \mathbb{P}, (\mathcal{G}_t))$ . The evaluation of  $F_d(T, \xi)$  can then be computed by approximating the expectation in (1.3) by Monte Carlo integration, that is, by simulating i.i.d. samples  $(S^{(i)})_{i=1}^m$  drawn from the distribution of  $S_T^\xi$  and by approximating  $F_d(T, \xi)$  with the empirical average  $\frac{1}{m} \sum_{i=1}^m \varphi_d(S^{(i)})$ . It is a standard result that the number of samples  $m$  needed to obtain a desired accuracy  $\varepsilon$  depends only polynomially on the dimension  $d$  and  $\varepsilon^{-1}$  [28].

If the goal, however, is to approximate  $F_d(T, \cdot)$  not only at a single value but, for example, on a full hypercube  $[u, v]^d$ , there has been no known method which does not suffer from the curse of dimensionality. In particular, it has been completely out of range to provably approximate  $F_d(T, \cdot)$  on  $[u, v]^d$  in high dimensions, say,  $d \gg 100$ .

The present paper introduces and analyzes deep learning–based algorithms for the numerical approximation of  $F_d(T, \cdot)$  on a full hypercube  $[u, v]^d$ . We will prove that the resulting algorithms overcome the curse of dimensionality and can consequently be efficiently applied even in high dimensions. Our proofs will be based on tools from statistical learning theory and the following key properties of linear Kolmogorov equations:

P.1 The fact that one can reformulate (1.1) as a learning problem (see Lemma 3.2).

---

<sup>2</sup>For  $x \in \mathbb{R}^d$ ,  $y \in \mathbb{R}^d$  we denote by  $x \cdot y := \sum_{i=1}^d x_i y_i$  the standard scalar product of  $x$  and  $y$ .

- P.2 The fact that typical initial values arising from problems in computational finance, such as, for example, (1.2), are either exactly representable as neural networks with ReLU activation function (ReLU networks) or can be approximated by such neural networks without incurring the curse of dimensionality (see [29, section 4]).
- P.3 The fact that Property P.2 is preserved under the evolution of linear Kolmogorov equations (1.1) with affine diffusion and drift coefficients, which implies that  $F_d(T, \cdot)$  can be approximated by ReLU networks without incurring the curse of dimensionality (see Theorem 3.3).

**1.2. Deep learning and statistical learning theory.** In their most basic incarnation, deep learning-based algorithms start with training data

$$((X_d^{(i)}, Y_d^{(i)}))_{i=1}^m : \Omega \rightarrow ([u, v]^d \times [-D, D])^m.$$

To give a concrete example,  $X_d^{(i)}$  may consist of different  $28 \times 28$  pixel grayscale images of handwritten digits and  $Y_d^{(i)}$  may consist of corresponding probabilities describing the likelihood of a certain digit to be shown in image  $X_d^{(i)}$  [43]. The goal is then to find a functional relation between images and labels and use it for predictive purposes on unseen images.

Empirical risk minimization (ERM) attempts to solve this prediction problem by minimizing the empirical risk

$$(1.4) \quad f \mapsto \widehat{\mathcal{E}}_{d,m}(f) := \frac{1}{m} \sum_{i=1}^m (f(X_d^{(i)}) - Y_d^{(i)})^2$$

over a compact<sup>3</sup> hypothesis class  $\mathcal{H} \subseteq \mathcal{C}([u, v]^d, \mathbb{R})$ , resulting in a predictor

$$\widehat{f}_{d,m,\mathcal{H}} \in \operatorname{argmin}_{f \in \mathcal{H}} \widehat{\mathcal{E}}_{d,m}(f)$$

that is hoped to provide a good approximation of the desired functional relation in the training data. In deep learning, these hypothesis classes consist of deep neural networks with fixed activation function  $\rho \in \mathcal{C}(\mathbb{R}, \mathbb{R})$ , parameter bound  $R \in (0, \infty)$ , and architecture<sup>4</sup>  $\mathbf{a} = (a_0, a_1, \dots, a_L) \in \mathbb{N}^{L+1}$ , where  $a_0 = d$  and  $a_L = 1$ . We define the corresponding set of neural network parametrizations

$$\mathcal{P}_{\mathbf{a}, R} := \bigtimes_{l=1}^L \left( [-R, R]^{a_l \times a_{l-1}} \times [-R, R]^{a_l} \right),$$

and for a parametrization  $\boldsymbol{\theta} = ((W_l, B_l))_{l=1}^L \in \mathcal{P}_{\mathbf{a}, R}$  we define its realization function<sup>5</sup>

$$\mathcal{F}_\rho(\boldsymbol{\theta}) := \mathcal{A}_{W_L, B_L} \circ \rho_* \circ \mathcal{A}_{W_{L-1}, B_{L-1}} \circ \rho_* \circ \cdots \circ \rho_* \circ \mathcal{A}_{W_1, B_1} \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}),$$

<sup>3</sup>Note that we equip  $\mathcal{C}([u, v]^d, \mathbb{R})$  with the uniform norm  $\|\cdot\|_{\mathcal{L}^\infty}$  which for  $f \in \mathcal{C}([u, v]^d, \mathbb{R})$  is given by  $\|f\|_{\mathcal{L}^\infty} = \|f\|_{\mathcal{L}^\infty([u, v]^d)} := \max_{x \in [u, v]^d} |f(x)|$ .

<sup>4</sup>Typically one calls a neural network “deep” if the architecture satisfies  $L > 2$ .

<sup>5</sup>If there is no possibility of ambiguity, we use the term “neural network” interchangeably for the parametrization and the realization function. However, note that  $\boldsymbol{\theta} \in \mathcal{P}_{\mathbf{a}, R}$  uniquely induces  $\mathcal{F}_\rho(\boldsymbol{\theta}) \in \mathcal{C}(\mathbb{R}^d, \mathbb{R})$ , while in general there can be multiple nontrivially different parametrizations with the same realization function; see [13].

where  $\mathcal{A}_{W,B}(x) := Wx + B$  and  $\rho_*(x) = (\rho(x_i))_{i=1}^n$ , i.e.,  $\rho$  is applied componentwise. Then neural network hypothesis classes are typically of the form

$$(1.5) \quad \mathcal{N}_{\rho,\mathbf{a},R}^{u,v} := \left\{ \left( [u, v]^d \ni x \mapsto \mathcal{F}_\rho(\boldsymbol{\theta})(x) \right) : \boldsymbol{\theta} \in \mathcal{P}_{\mathbf{a},R} \right\}.$$

Despite the great practical success of the “deep learning paradigm,” for generic real world training data it is far out of reach to specify network architectures which guarantee a desired performance on unseen data; see also [62].

This type of problem can be theoretically studied using tools developed within the field of statistical learning theory. There it is typically postulated that  $((X_d^{(i)}, Y_d^{(i)}))_{i=1}^m$  are i.i.d. samples drawn from the distribution of some (unknown) data  $(X_d, Y_d)$  and that the optimal functional relation between  $X_d$  and  $Y_d$  is given by the regression function

$$f_d^* : \begin{cases} [u, v]^d & \rightarrow \mathbb{R} \\ x & \mapsto \mathbb{E}[Y_d | X_d = x], \end{cases}$$

which minimizes the risk  $f \mapsto \mathcal{E}_d(f) := \mathbb{E}[(f(X_d) - Y_d)^2]$  (see Lemma 2.2). The minimization of functionals of the form  $\mathcal{E}_d$  is commonly referred to as a

*statistical learning problem with data  $(X_d, Y_d)$  and quadratic loss function.*

Under strong regularity assumptions on the regression function  $f_d^*$  and the distribution of  $(X_d, Y_d)$  it is possible to obtain bounds on the sample size  $m$  and the “complexity” of the hypothesis class  $\mathcal{H}$  in order to guarantee, with high probability, an error

$$(1.6) \quad \mathbb{E}[(\hat{f}_{d,m,\mathcal{H}}(X_d) - f_d^*(X_d))^2] \leq \varepsilon;$$

see, for example, [4, 7, 18, 19, 31, 40, 46, 60]. In the above, the regularity of the regression function  $f_d^*$  quantifies how well  $f_d^*$  can be approximated by the hypothesis class  $\mathcal{H}$ .

In the case of the neural network hypothesis classes  $\mathcal{H} = \mathcal{N}_{\rho,\mathbf{a},R}^{u,v}$  these regularity assumptions are met if  $f_d^*$  satisfies certain smoothness assumptions; see [15, 16, 26, 51, 53, 57, 61]. Moreover, the complexity of  $\mathcal{N}_{\rho,\mathbf{a},R}^{u,v}$  can mainly be described by the size of the neural network parametrizations, i.e., the number of network parameters

$$P(\mathbf{a}) := \sum_{l=1}^L a_l a_{l-1} + a_L.$$

We will show in subsection 1.3 below that our specific deep learning-based method for numerically solving Kolmogorov equations allows us to rigorously apply tools from statistical learning theory, as we can overcome the following potential problems:

- R.1 The crucial assumption that the training data consists of i.i.d. samples drawn from an underlying probability distribution is usually debatable or at least hard to verify.
- R.2 Even if this assumption were satisfied, the underlying distribution of  $(X_d, Y_d)$  is typically unknown. Thus, it is hard to ensure a priori the regularity assumptions on  $f_d^*$ , which are needed to apply tools from statistical learning theory.

- R.3 Since the distribution of  $X_d$  is typically unknown, it is not clear how the quantity  $\mathbb{E}[(\hat{f}_{d,m,\mathcal{H}}(X_d) - f_d^*(X_d))^2]$  of (1.6) can be interpreted.
- R.4 Most of the classical techniques operate in an asymptotic regime where the number of training samples  $m$  exceeds the network size  $P(\mathbf{a})$ . However, in many applications the number of training samples is fixed, and it is not possible to generate more training data at will.

**1.3. Kolmogorov equations as learning problem.** We will reformulate the numerical approximation of  $F_d(T, \cdot)$  on  $[u, v]^d$  as a statistical learning problem and demonstrate that in this specific case none of the aforementioned Problems R.1–R.4 appears. Let  $X_d \sim \mathcal{U}([u, v]^d)$  be uniformly distributed on  $[u, v]^d$  and define  $Y_d := \varphi_d(S_T^{X_d})$ , where  $(S_t^{X_d})_{t \in [0, T]}$  is a stochastic process satisfying the SDE

$$(1.7) \quad dS_t^{X_d} = \sigma_d(S_t^{X_d}) dB_t^d + \mu_d(S_t^{X_d}) dt \quad \text{and} \quad S_0^{X_d} = X_d.$$

Under suitable conditions it then follows from the Feynman–Kac formula (1.3) that  $F_d(T, \cdot)$  is the minimizer of the risk functional  $\mathcal{E}_d$ , that is,  $f_d^*(x) = F_d(T, x)$  for a.e.  $x \in [u, v]^d$ ; see Lemma 3.2. As outlined in subsection 1.2, we thus have that

*the end value of the Kolmogorov equation  $F_d(T, \cdot)$  is the solution to the statistical learning problem with data  $(X_d, \varphi_d(S_T^{X_d}))$  and quadratic loss function.*

A natural next step is to apply the deep learning paradigm, that is, for  $m$  i.i.d. samples  $((X_d^{(i)}, Y_d^{(i)}))_{i=1}^m$  drawn from the distribution of  $(X_d, Y_d)$  to minimize the empirical risk (1.4) over a hypothesis class of neural networks  $\mathcal{N}_{\rho, \mathbf{a}, R}^{u, v}$ .

In [10] this idea has been implemented with suitable classes of deep neural networks of a given architecture as hypothesis class  $\mathcal{H}$ . In extensive numerical simulations it was observed that the proposed algorithm is efficient even in very high dimensions, suggesting that it does not suffer from the curse of dimensionality. Similar conclusions can be found in related work [11, 12, 21, 22, 25, 33, 34, 59], which covers topics ranging from American option pricing problems to fully nonlinear PDEs. Note that in [59] a nonquantitative analysis of the approximation error is given; all the other works are purely empirical. To the best of our knowledge, this is the first joint quantitative analysis of approximation and generalization error confirming the efficiency of deep learning–based methods applied to the numerical solution of high-dimensional PDEs.

Two main parameters influence the complexity of the algorithm described above: the number  $P(\mathbf{a}_{d,\varepsilon})$  of network parameters that need to be optimized as well as the number of training samples  $m_{d,\varepsilon}$  needed to guarantee that, with high probability, the estimate

$$(1.8) \quad \frac{1}{(v-u)^d} \|\hat{f}_{d,m_{d,\varepsilon},\mathcal{H}_{d,\varepsilon}} - F_d(T, \cdot)\|_{\mathcal{L}^2([u,v]^d)}^2 = \mathbb{E} \left[ (\hat{f}_{d,m_{d,\varepsilon},\mathcal{H}_{d,\varepsilon}}(X_d) - F_d(T, X_d))^2 \right] \leq \varepsilon$$

holds true. We are interested in the scaling of  $m_{d,\varepsilon}$  and  $P(\mathbf{a}_{d,\varepsilon})$  with respect to the precision  $\varepsilon$  and dimension  $d$ .

Observe that the data distribution  $(X_d, \varphi_d(S_T^{X_d}))$  is now explicitly known and i.i.d. samples of this distribution can be simulated as needed ( $X_d$  is uniformly distributed and can be simulated using a suitable random number generator, and  $S_T^{X_d}$  can be simulated by any

numerical solver for the SDE (1.7); see [28]). Moreover, the uniform distribution of the input data  $X_d$  gives rise to typical  $\mathcal{L}^2$ -error estimates (1.8) and, in the case of affine coefficients and suitable initial values, we can establish bounds on how well the regression function can be approximated by neural network hypothesis classes; see Theorem 3.3. In particular, contrary to conventional learning problems, in the statistical learning problem that arises from our reformulation of the Kolmogorov equation, none of the Problems R.1–R.4 described in subsection 1.2 occurs. We will therefore be able to rigorously invoke tools from statistical learning theory to obtain bounds on the quantities  $m_{d,\varepsilon}$  and  $P(\mathbf{a}_{d,\varepsilon})$  above.

**1.4. Contribution.** We show that whenever  $(\sigma_d)_{d \in \mathbb{N}}$  and  $(\mu_d)_{d \in \mathbb{N}}$  are affine functions (this includes the important case of the Black–Scholes equation in option pricing) and the initial values  $(\varphi_d)_{d \in \mathbb{N}}$  can be approximated by deep neural networks without the curse of dimensionality (this can easily shown to be true for a large number of relevant options such as basket call, basket put, call on max, and call on min options), there exists a polynomial  $p: \mathbb{R}^2 \rightarrow \mathbb{R}$  such that for every  $d \in \mathbb{N}$ ,  $\varepsilon \in (0, 1)$  it holds that

$$\max\{m_{d,\varepsilon}, P(\mathbf{a}_{d,\varepsilon})\} \leq p(\varepsilon^{-1}, d);$$

see Corollary 3.5. We conclude that the aforementioned deep learning–based algorithm *does not suffer from the curse of dimensionality*.

We briefly describe our proof strategy for bounding the error between the empirical risk minimizer and the end value of the Kolmogorov equation

$$\frac{1}{(v-u)^d} \|\widehat{f}_{d,m,\mathcal{H}} - F_d(T, \cdot)\|_{\mathcal{L}^2([u,v]^d)}^2$$

as in (1.8). By the so-called bias-variance decomposition we can represent this error as the sum of a generalization error and an approximation error, i.e.,

$$\underbrace{\mathcal{E}_d(\widehat{f}_{d,m,\mathcal{H}}) - \min_{f \in \mathcal{H}} \mathcal{E}_d(f)}_{\text{generalization error}} + \underbrace{\min_{f \in \mathcal{H}} \frac{1}{(v-u)^d} \|f - F_d(T, \cdot)\|_{\mathcal{L}^2([u,v]^d)}^2}_{\text{approximation error}};$$

see Lemma 2.2. Bounds on the size of neural networks with ReLU activation function

$$\rho(x) = \text{ReLU}(x) := \max\{x, 0\}$$

needed to approximate  $F_d(T, \cdot)$  up to a desired error have been analyzed in [29]. In Theorem 3.3 we notably extend these approximation results by proving corresponding bounds on the parameter magnitudes. This is done by analyzing the special structure of the solution to the SDE (1.7) in the case of affine coefficients  $\sigma_d$  and  $\mu_d$ , employing the Feynman–Kac formula (1.3), and constructing a neural network simulating Monte Carlo sampling. Together with the results of [29], we achieve that neural network hypothesis classes with ReLU activation function are capable of approximating the end values  $(F_d(T, \cdot))_{d \in \mathbb{N}}$  without incurring the curse of dimensionality whenever the same is true for the initial values  $(\varphi_d)_{d \in \mathbb{N}}$ .

We then leverage the approximation results as well as tools from [4, 18, 19] to obtain probabilistic estimates of the generalization error; see Theorem 3.4. These tools require bounds

on the covering numbers<sup>6</sup>  $\text{Cov}(\mathcal{H}, r)$  of hypothesis classes consisting of neural networks. To this end, we compute the Lipschitz constant of the operator  $\mathcal{F}_\rho$  which maps neural network parametrizations to the corresponding realization functions; see Theorem 2.6. Using a standard result on the covering number of balls in a Euclidean space, we obtain that<sup>7</sup>

$$\ln \text{Cov}(\mathcal{N}_{\text{ReLU}, \mathbf{a}, R}^{u, v}, r) \leq P(\mathbf{a}) \left[ \ln \left( \frac{4L^2 \max\{1, |u|, |v|\}}{r} \right) + L \ln(R\|\mathbf{a}\|_\infty) \right];$$

see Proposition 2.8. In conjunction with Hoeffding's inequality this allows us to uniformly (over the hypothesis class of neural networks) bound the error between the risk and the empirical risk. However, this requires that the regression function  $f_d^*$  as well as all functions in  $\mathcal{H}$  are uniformly bounded. To that end we assume the initial value  $\varphi_d$  to be bounded, which by the Feynman–Kac formula (1.3) implies that also the function  $f_d^* = F_d(T, \cdot)$  is bounded. Moreover, we introduce hypothesis classes of “clipped” neural networks

$$\mathcal{N}_{\rho, \mathbf{a}, R, D}^{u, v} := \left\{ \text{clip}_D \circ g : g \in \mathcal{N}_{\rho, \mathbf{a}, R}^{u, v} \right\},$$

where  $\text{clip}_D := \min\{|\cdot|, D\} \text{sgn}(\cdot)$  denotes a clipping function with clipping amplitude  $D$ . This can be interpreted as incorporating the prior knowledge about the boundedness of the regression function into our hypothesis class. In Appendix A.2 we show that the clipping function can be represented as a small ReLU network so that clipped ReLU networks are in fact standard neural networks.

Note that there exist different concepts and known results in order to bound the generalization error (see, for instance, [4, 6, 8, 9, 27, 49]). The present paper intends to stress the interplay between the approximation and generalization error and gives a complete proof in order to rigorously show the absence of the curse of dimensionality for our particular problem.

We are now ready to formulate a first specific result of this paper as an appetizer. It demonstrates that deep learning–based ERM succeeds in solving the option pricing problem for European put options without incurring the curse of dimensionality.

**Theorem 1.1 (pricing of options without curse of dimensionality).** *Let  $T, K, D \in [1, \infty)$ ,  $u \in \mathbb{R}$ , and  $v \in (u, \infty)$ , and let  $(\Omega, \mathcal{G}, \mathbb{P}, (\mathcal{G}_t))$  be a filtered probability space. For every dimension  $d \in \mathbb{N}$  let  $c_d \in [-D, D]^d$ , let the initial value  $\varphi_d \in \mathcal{C}(\mathbb{R}^d, \mathbb{R})$  satisfy for every  $x \in \mathbb{R}^d$  that*

$$\varphi_d(x) = \min \left\{ \max \{D - c_d \cdot x, 0\}, D \right\},$$

*let the drift and diffusion coefficients  $\mu_d \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$ ,  $\sigma_d \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^{d \times d})$  be affine functions satisfying for every  $x \in \mathbb{R}^d$  that*

$$\|\sigma_d(x)\|_2 + \|\mu_d(x)\|_2 \leq K(1 + \|x\|_2),$$

*and let  $F_d \in \mathcal{C}([0, T] \times \mathbb{R}^d, \mathbb{R})$  be the unique at most polynomially growing viscosity solution<sup>8</sup> of the corresponding  $d$ -dimensional Kolmogorov equation*

$$\begin{cases} \frac{\partial F_d}{\partial t}(t, x) = \frac{1}{2} \text{Trace}(\sigma_d(x)[\sigma_d(x)]^*(\text{Hess}_x F_d)(t, x)) + \mu_d(x) \cdot (\nabla_x F_d)(t, x), \\ F_d(0, x) = \varphi_d(x). \end{cases}$$

<sup>6</sup>The covering number  $\text{Cov}(\mathcal{H}, r)$  is the minimal number of balls of radius  $r$  covering  $\mathcal{H}$ ; see Setting 2.3.

<sup>7</sup>For  $p \geq 1$ , a finite index set  $I$ , and  $M \in \mathbb{R}^I$  we define  $\|M\|_\infty := \max_{i \in I} |M_i|$  and  $\|M\|_p := (\sum_{i \in I} |M_i|^p)^{1/p}$ .

<sup>8</sup>We refer the interested reader to [32] for the definition and properties of viscosity solutions.

For every  $d \in \mathbb{N}$  let the input data  $X_d \sim \mathcal{U}([u, v]^d)$  be uniformly distributed on  $[u, v]^d$  and  $\mathcal{G}_0$ -measurable, let  $B^d$  be a  $d$ -dimensional  $(\mathcal{G}_t)$ -Brownian motion, let  $(S_t^{X_d})_{t \in [0, T]}$  be a  $(\mathcal{G}_t)$ -adapted stochastic process with continuous sample paths satisfying the SDE

$$dS_t^{X_d} = \sigma_d(S_t^{X_d}) dB_t^d + \mu_d(S_t^{X_d}) dt \quad \text{and} \quad S_0^{X_d} = X_d$$

$\mathbb{P}$ -a.s. for every  $t \in [0, T]$ , define the label  $Y_d := \varphi_d(S_T^{X_d})$ , and let  $((X_d^{(i)}, Y_d^{(i)}))_{i \in \mathbb{N}}$  be i.i.d. random variables (training data) with  $(X_d^{(1)}, Y_d^{(1)}) \sim (X_d, Y_d)$ . Then there exists a constant  $C \in (0, \infty)$  such that the following holds: For every  $d, m \in \mathbb{N}$ ,  $\varepsilon, \varrho \in (0, 1)$  with

$$m \geq Cd\varepsilon^{-4}(1 + \ln(d\varepsilon^{-1}\varrho^{-1})) \quad (\text{number of samples})$$

there exist  $\mathbf{a} = (d, a_1, a_2, 1) \in \mathbb{N}^4$  and  $R \in [1, \infty)$  such that it holds that

- (i)  $\mathbb{P}\left[\frac{1}{(v-u)^d} \|\widehat{f}_{d,m,\mathcal{H}} - F_d(T, \cdot)\|_{\mathcal{L}^2([u,v]^d)}^2 \leq \varepsilon\right] \geq 1 - \varrho$ ,
- (ii)  $P(\mathbf{a}) \leq Cd\varepsilon^{-2}$  (number of parameters),
- (iii)  $R \leq Cd^{3/2}\varepsilon^{-1}$  (parameter bound), and
- (iv)  $\max\{a_1, a_2\} \leq Cd\varepsilon^{-1}$  (size of the architecture),

where  $\widehat{f}_{d,m,\mathcal{H}} \in \operatorname{argmin}_{f \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^m (f(X_d^{(i)}) - Y_d^{(i)})^2$  is a measurable empirical risk minimizer in the corresponding hypothesis class of clipped ReLU networks  $\mathcal{H} := \mathcal{N}_{\text{ReLU}, \mathbf{a}, R, D}^{u, v}$ .

A proof will be given in subsection 3.3. In a more general context, Theorem 3.4 states that a result analogous to Theorem 1.1 holds true whenever the initial values  $(\varphi_d)_{d \in \mathbb{N}}$  can be approximated by ReLU networks without the curse of dimensionality.

Note that our analysis does not consider the computational cost of solving the nonsmooth, nonconvex ERM problem (1.4). This is typically achieved by stochastic first order optimization methods whose theoretical analysis is still an open problem. While there are many interesting approaches to the latter question, they tend to require very strong assumptions (e.g., (almost) linearity, convexity, extreme overparametrization, or inverse stability of  $\mathcal{F}_\rho$  [2, 13, 17, 20, 38, 44, 45, 47, 58]), which we want to avoid in our analysis.

**1.5. Extensions.** Our results in section 3, in conjunction with the results of [29], can be applied to prove the absence of the curse of dimensionality in the pricing of (capped) basket call, basket put, call on max, and call on min options. Moreover, the results of section 2 hold for a general statistical learning problem within Setting 2.1. That is, ReLU network approximation results for the regression function translate directly into generalization results without incurring the curse of dimensionality. If suitable learning problems can be established, this work can extend various neural network approximation results for PDEs (see, e.g., [30, 41, 55]) to also consider the generalization error and get one step closer to a full error analysis. For instance, there are stronger approximation results for more restricted option pricing problems [23], and there are very recent approximation results for semilinear heat equations [36] and Kolmogorov equations with (time-inhomogeneous) nonlinear coefficients [37, 54] where dependence on the dimension is polynomial. Using a generalized version of Lemma 3.2, the findings of this paper can be used to prove that the corresponding ERM problem achieves, with high probability, a desired accuracy  $\varepsilon$  with the number of samples and the size of the

hypothesis class scaling only polynomially in  $d$  and  $\varepsilon^{-1}$ . In particular this means that the presented methods are not restricted to the case of linear Kolmogorov equations with affine drift and diffusion coefficients. Finally, note that one obtains similar results for any continuous piecewise linear activation function with a finite number of breakpoints; see the comment after Theorem 2.6 and [61, Proposition 1].

**1.6. Outline.** The outline is as follows. In section 2 we present our main result related to the generalization of neural networks in a rather general setting. Whenever the regression functions  $(f_d^*)_{d \in \mathbb{N}}$  can be approximated without the curse of dimensionality by clipped ReLU networks, we show that also the number  $m$  of required training samples to achieve a desired accuracy  $\varepsilon$  with high probability does not suffer from the curse of dimensionality. This result is proven using tools from statistical learning theory and covering number estimates of neural network hypothesis classes. In section 3 we reformulate the numerical approximation of  $F_d(T, \cdot)$  on  $[u, v]^d$  as a statistical learning problem and extend a result of [29] claiming that the end values  $(F_d(T, \cdot))_{d \in \mathbb{N}}$  can be approximated by clipped ReLU networks without the curse of dimensionality. Therefore our results from section 2 apply and give rise to quantitative polynomial bounds on the number of samples and the size of the network in Theorem 3.4.

**2. Results in statistical learning theory.** The present section develops generalization bounds for ERM problems in the spirit of [4, 18, 19].

**2.1. A generalization result based on covering numbers.** Setting 2.1 describes a standard statistical learning problem.

*Setting 2.1 (statistical learning problem).* Let  $u \in \mathbb{R}$ ,  $v \in (u, \infty)$ , and  $D \in [1, \infty)$ , and let  $(\Omega, \mathcal{G}, \mathbb{P})$  be a probability space. For every  $d \in \mathbb{N}$  let

$$X_d : \Omega \rightarrow [u, v]^d \quad (\text{input data}) \quad \text{and} \quad Y_d : \Omega \rightarrow [-D, D] \quad (\text{label})$$

be random variables, let  $\mathbb{P}_{X_d}$  be the image measure of  $X_d$  on the hypercube  $[u, v]^d$ , let

$$(X_d^{(i)}, Y_d^{(i)}) : \Omega \rightarrow [u, v]^d \times [-D, D], \quad i \in \mathbb{N} \quad (\text{training data}),$$

be i.i.d. random variables with  $(X_d^{(1)}, Y_d^{(1)}) \sim (X_d, Y_d)$ , and let  $f_d^* \in L^2(\mathbb{P}_{X_d})$  satisfy<sup>9</sup> for  $\mathbb{P}_{X_d}$ -a.s.  $x \in [u, v]^d$  that

$$f_d^*(x) = \mathbb{E}[Y_d | X_d = x] \quad (\text{regression function}).$$

For every  $d, m \in \mathbb{N}$  and Borel measurable function  $f : [u, v]^d \rightarrow \mathbb{R}$  define the *risk*  $\mathcal{E}_d(f) \in [0, \infty]$  and the *empirical risk*  $\widehat{\mathcal{E}}_{d,m}(f) : \Omega \rightarrow [0, \infty)$  by

$$\mathcal{E}_d(f) := \mathbb{E}\left[\left(f(X_d) - Y_d\right)^2\right] \quad \text{and} \quad \widehat{\mathcal{E}}_{d,m}(f) := \frac{1}{m} \sum_{i=1}^m \left(f(X_d^{(i)}) - Y_d^{(i)}\right)^2.$$

---

<sup>9</sup>We define the Hilbert space  $L^2(\mathbb{P}_{X_d})$  as the space of all Borel measurable functions  $f : [u, v]^d \rightarrow \mathbb{R}$  with finite norm  $\|f\|_{L^2(\mathbb{P}_{X_d})} = \left(\int_{[u,v]^d} f^2 d\mathbb{P}_{X_d}\right)^{1/2} < \infty$  where functions coinciding  $\mathbb{P}_{X_d}$ -a.s. are identified as usual.

For every  $d, m \in \mathbb{N}$  and every compact  $\mathcal{H} \subseteq \mathcal{C}([u, v]^d, \mathbb{R})$  (*hypothesis class*) let

$$(2.1) \quad f_{d,\mathcal{H}} \in \operatorname{argmin}_{f \in \mathcal{H}} \mathcal{E}_d(f) \quad (\text{best approximation}),$$

and for every  $d, m \in \mathbb{N}$ ,  $\omega \in \Omega$  and every compact  $\mathcal{H} \subseteq \mathcal{C}([u, v]^d, \mathbb{R})$  let

$$(2.2) \quad \hat{f}_{d,m,\mathcal{H}}(\omega) \in \operatorname{argmin}_{f \in \mathcal{H}} \hat{\mathcal{E}}_{d,m}(f)(\omega) \quad (\text{empirical regression function})$$

such that the mapping  $\Omega \ni \omega \mapsto \hat{f}_{d,m,\mathcal{H}}(\omega)$  is measurable.

We want to emphasize that the minima in (2.1) and (2.2) will be attained due to the compactness of our hypothesis class, but they need not be unique. We require the mapping  $\Omega \ni \omega \mapsto \hat{f}_{d,m,\mathcal{H}}(\omega)$  to be measurable in order to view the risk of the empirical regression function as a random variable  $\Omega \ni \omega \mapsto \mathcal{E}_d(\hat{f}_{d,m,\mathcal{H}}(\omega))$ . This ensures that the probability in the generalization error bound (Theorem 2.4) is well-defined. While this technical assumption is often not explicitly stated in the literature on statistical learning theory, it is actually crucial for analyzing the generalization error. In our setting (by choosing a suitable minimizer) measurability can indeed be satisfied; see [1, Theorem 18.19] and section S.1 of the supplementary material (M125649.01.pdf [local/web 231KB]).

The following lemma states that the regression function  $f_d^*$  indeed minimizes the risk and the function  $f_{d,\mathcal{H}}$  is a best approximation of  $f_d^*$  in  $\mathcal{H}$  with respect to the  $\mathcal{L}^2(\mathbb{P}_{X_d})$ -norm. Moreover, it offers a decomposition of the error between the empirical regression function  $\hat{f}_{d,m,\mathcal{H}}$  and the regression function  $f_d^*$ , often referred to as the bias-variance decomposition.

**Lemma 2.2 (bias-variance decomposition).** *Assume Setting 2.1. Let  $d, m \in \mathbb{N}$  and let  $\mathcal{H} \subseteq \mathcal{C}([u, v]^d, \mathbb{R})$  be compact. Then it holds that*

- (i)  $\mathcal{E}_d(f_d^*) = \min_{f \in \mathcal{L}^2(\mathbb{P}_{X_d})} \mathcal{E}_d(f)$ ,
- (ii)  $\|f_{d,\mathcal{H}} - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})} = \min_{f \in \mathcal{H}} \|f - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}$ , and
- (iii)  $\|\hat{f}_{d,m,\mathcal{H}} - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 = \underbrace{\mathcal{E}_d(\hat{f}_{d,m,\mathcal{H}}) - \mathcal{E}_d(f_{d,\mathcal{H}})}_{\text{generalization error (variance)}} + \underbrace{\|f_{d,\mathcal{H}} - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2}_{\text{approximation error (bias)}}.$

For a corresponding result see also [18]. The proof in Appendix A.1 is based on the fact that we consider the square loss, and thus the risk of  $f \in \mathcal{L}^2(\mathbb{P}_{X_d})$  can be represented as

$$\mathcal{E}_d(f) = \|f - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 + \mathcal{E}_d(f_d^*).$$

We now introduce the concept of covering numbers in order to bound the generalization error.

**Setting 2.3 (covering number).** For every  $r \in (0, 1)$ , every normed vector space  $(\mathcal{Z}, \|\cdot\|)$ , and every compact subset  $\mathcal{H} \subseteq \mathcal{Z}$  we define the  $r$ -covering number of  $\mathcal{H}$  w.r.t.  $\|\cdot\|$  by

$$\operatorname{Cov}(\mathcal{H}, \|\cdot\|, r) := \min \left\{ n \in \mathbb{N} : \text{ There exists } (f_i)_{i=1}^n \subseteq \mathcal{H} \text{ with } \mathcal{H} \subseteq \bigcup_{i=1}^n \operatorname{Ball}_r(f_i) \right\},$$

where  $\operatorname{Ball}_r(f) := \{g \in \mathcal{H} : \|f - g\| \leq r\}$  denotes the ball of radius  $r$  around  $f \in \mathcal{H}$ . If the norm is clear from the context, we will use the abbreviation  $\operatorname{Cov}(\mathcal{H}, r) := \operatorname{Cov}(\mathcal{H}, \|\cdot\|, r)$ .

Assume that the functions in our hypothesis class  $\mathcal{H}$  are uniformly bounded and that balls of radius  $r$  around the functions  $(f_i)_{i=1}^n$  cover  $\mathcal{H}$ . We can then use the (uniform) Lipschitz continuity of the (empirical) risk to bound the generalization error by

$$\begin{aligned}\mathcal{E}_d(\widehat{f}_{d,m,\mathcal{H}}) - \mathcal{E}_d(f_{d,\mathcal{H}}) &\leq \mathcal{E}_d(\widehat{f}_{d,m,\mathcal{H}}) - \widehat{\mathcal{E}}_{d,m}(\widehat{f}_{d,m,\mathcal{H}}) + \widehat{\mathcal{E}}_{d,m}(f_{d,\mathcal{H}}) - \mathcal{E}_d(f_{d,\mathcal{H}}) \\ &\leq 2r[\text{Lip}(\mathcal{E}_d) + \text{Lip}(\widehat{\mathcal{E}}_{d,m})] + 2\max_{i=1}^n |\mathcal{E}_d(f_i) - \widehat{\mathcal{E}}_{d,m}(f_i)|\end{aligned}$$

and employ Hoeffding's inequality and a union bound to obtain the following estimate.

**Theorem 2.4 (generalization error bound).** *Assume Settings 2.1 and 2.3. Let  $\varepsilon \in (0, 1)$ ,  $d, m \in \mathbb{N}$  and let  $\mathcal{H} \subseteq \mathcal{C}([u, v]^d, \mathbb{R})$  be compact with  $\sup_{f \in \mathcal{H}} \|f\|_{\mathcal{L}^\infty} \leq D$ . Then it holds that*

$$\mathbb{P} [\mathcal{E}_d(\widehat{f}_{d,m,\mathcal{H}}) - \mathcal{E}_d(f_{d,\mathcal{H}}) \leq \varepsilon] \geq 1 - 2\text{Cov}\left(\mathcal{H}, \frac{\varepsilon}{32D}\right) \exp\left(-\frac{m\varepsilon^2}{128D^4}\right).$$

This result is adapted from [4, Theorem 17.1], [19, Theorem 3.14], and [48, Example 3.31], and for the sake of completeness we present a detailed proof in section S.2 in the supplementary material (M125649\_01.pdf [[local](#)/[web](#) 231KB]).

**2.2. Covering numbers of neural network hypothesis classes.** As a natural next step we prove estimates on the covering numbers of neural network hypothesis classes in order to leverage the result of Theorem 2.4. Note that for different assumptions (i.e., boundedness assumptions on the activation function, different norms on the parameters, or evaluation of the neural networks on input data) similar approaches can be found in [4, 8].

The following setting describes suitable hypothesis classes based on neural networks. From now on we only consider neural networks with ReLU activation function and therefore omit writing the index  $\rho = \text{ReLU}$  in our notation.

**Setting 2.5 (neural networks).** Assume Setting 2.1. For every  $k, n \in \mathbb{N}$ ,  $W \in \mathbb{R}^{k \times n}$ ,  $B \in \mathbb{R}^k$  let  $\mathcal{A}_{W,B} \in \mathcal{C}(\mathbb{R}^n, \mathbb{R}^k)$  be the affine mapping which satisfies for every  $x \in \mathbb{R}^n$  that  $\mathcal{A}_{W,B}(x) := Wx + B$ . For every  $n \in \mathbb{N}$ ,  $x = (x_i)_{i=1}^n \in \mathbb{R}^n$  we define

$$\text{ReLU}_*(x) := (\max\{x_i, 0\})_{i=1}^n \quad (\text{componentwise rectified linear unit}).$$

For every  $L \in \mathbb{N}$ ,  $\mathbf{a} = (a_0, a_1, \dots, a_L) \in \mathbb{N}^{L+1}$  (*network architecture*) we define

$$\mathcal{P}_{\mathbf{a}} := \bigtimes_{l=1}^L (\mathbb{R}^{a_l \times a_{l-1}} \times \mathbb{R}^{a_l}) \quad (\text{set of neural network parametrizations}),$$

$L(\mathbf{a}) := L$  (*number of layers*), and

$$P(\mathbf{a}) := \sum_{l=1}^L a_l a_{l-1} + a_L \quad (\text{number of parameters}).$$

For every  $L \in \mathbb{N}$ ,  $\mathbf{a} \in \mathbb{N}^{L+1}$ ,  $\boldsymbol{\theta} = ((W_l, B_l))_{l=1}^L \in \mathcal{P}_{\mathbf{a}}$  we define the *neural network realization function*  $\mathcal{F}(\boldsymbol{\theta}) \in \mathcal{C}(\mathbb{R}^{a_0}, \mathbb{R}^{a_L})$  by

$$\mathcal{F}(\boldsymbol{\theta}) := \mathcal{A}_{W_L, B_L} \circ \text{ReLU}_* \circ \mathcal{A}_{W_{L-1}, B_{L-1}} \circ \text{ReLU}_* \circ \dots \circ \text{ReLU}_* \circ \mathcal{A}_{W_1, B_1}$$

and its restriction  $\mathcal{F}^{u,v}(\boldsymbol{\theta}) := \mathcal{F}(\boldsymbol{\theta})|_{[u,v]^{a_0}} \in \mathcal{C}([u,v]^{a_0}, \mathbb{R}^{a_L})$  to the hypercube  $[u,v]^{a_0}$ . For every  $d \in \mathbb{N}$  let the *admissible network architectures* be given by

$$\mathbf{A}_d := \bigcup_{L \in \mathbb{N}} \{(a_0, a_1, \dots, a_L) \in \mathbb{N}^{L+1} : a_0 = d, a_L = 1\}.$$

For every  $d \in \mathbb{N}$ ,  $\mathbf{a} \in \mathbf{A}_d$ ,  $R \in (0, \infty)$  (*parameter bound*) define the *set of bounded neural network parametrizations*

$$\mathcal{P}_{\mathbf{a}, R} := \bigtimes_{l=1}^L ([-R, R]^{a_l \times a_{l-1}} \times [-R, R]^{a_l}) = \{\boldsymbol{\theta} \in \mathcal{P}_{\mathbf{a}} : \|\boldsymbol{\theta}\|_{\infty} \leq R\},$$

the *hypothesis class of neural networks*

$$\mathcal{N}_{\mathbf{a}, R}^{u, v} := \mathcal{F}^{u, v}(\mathcal{P}_{\mathbf{a}, R}) = \left\{ \left( [u, v]^d \ni x \mapsto \mathcal{F}(\boldsymbol{\theta})(x) \right) : \boldsymbol{\theta} \in \mathcal{P}_{\mathbf{a}, R} \right\},$$

and the *hypothesis class of clipped neural networks*

$$\mathcal{N}_{\mathbf{a}, R, D}^{u, v} := \left\{ \text{clip}_D \circ g : g \in \mathcal{N}_{\mathbf{a}, R}^{u, v} \right\},$$

where  $\text{clip}_D \in \mathcal{C}(\mathbb{R}, \mathbb{R})$  (*clipping function*) satisfies for every  $x \in \mathbb{R}$  that

$$\text{clip}_D(x) := \min\{|x|, D\} \operatorname{sgn}(x).$$

The hypothesis classes  $\mathcal{N}_{\mathbf{a}, R, D}^{u, v}$  are somewhat nonstandard in the sense that the clipping function  $\text{clip}_D$  is applied to the output of a neural network realization. The reason for our choice of this definition is that Theorem 2.4 requires that the set of neural networks over which the ERM problem is solved consists of uniformly bounded functions. In Appendix A.2 we show that the clipping function  $\text{clip}_D$  can be represented as a small neural network, which implies that the seemingly nonstandard classes  $\mathcal{N}_{\mathbf{a}, R, D}^{u, v}$  are actually conventional neural network classes that can be trained with standard methods [35, 39, 42, 50].

The next theorem quantifies the Lipschitz continuity  $\operatorname{Lip}(\mathcal{F}^{u, v})$  of the operator

$$\mathcal{F}^{u, v} : (\mathcal{P}_{\mathbf{a}, R}, \|\cdot\|_{\infty}) \rightarrow (\mathcal{N}_{\mathbf{a}, R}^{u, v}, \|\cdot\|_{\mathcal{L}^{\infty}}),$$

which maps bounded neural network parametrizations with fixed architecture to the corresponding realization functions (restricted to  $[u, v]^d$ ).

**Theorem 2.6 (Lipschitz continuity of  $\mathcal{F}$ ).** *Assume Setting 2.5. Let  $d \in \mathbb{N}$ ,  $\mathbf{a} \in \mathbf{A}_d$ , and  $R \in [1, \infty)$ . Then for every  $\boldsymbol{\theta}, \boldsymbol{\eta} \in \mathcal{P}_{\mathbf{a}, R}$  it holds that*

$$\|\mathcal{F}^{u, v}(\boldsymbol{\theta}) - \mathcal{F}^{u, v}(\boldsymbol{\eta})\|_{\mathcal{L}^{\infty}} \leq 2 \max\{1, |u|, |v|\} L(\mathbf{a})^2 R^{L(\mathbf{a})-1} \|\mathbf{a}\|_{\infty}^{L(\mathbf{a})} \|\boldsymbol{\theta} - \boldsymbol{\eta}\|_{\infty}.$$

The proof is based on estimating the error amplification in each layer of the neural network and can be found in Appendix A.3. Similar results can be established for any Lipschitz continuous activation function; see also [52] for a general nonquantitative result. Note that Theorem 2.6 in particular implies that  $\mathcal{N}_{\mathbf{a}, R}^{u, v}$  and  $\mathcal{N}_{\mathbf{a}, R, D}^{u, v}$  are compact subsets of  $\mathcal{C}([u, v]^d, \mathbb{R})$  and thus valid hypothesis classes, as required by Setting 2.1. Next, we recall a basic result on the covering number of a hypercube w.r.t. the maximum norm  $\|\cdot\|_{\infty}$ . Note that a similar statement holds for any ball in a finite-dimensional Banach space; see [18, Proposition 5].

**Lemma 2.7 (covering numbers of balls).** Assume Setting 2.3. Let  $n \in \mathbb{N}$ ,  $R \in [1, \infty)$ , and  $r \in (0, 1)$ , and define  $\text{Ball}_R := \{\boldsymbol{\theta} \in \mathbb{R}^n : \|\boldsymbol{\theta}\|_\infty \leq R\}$ . Then it holds that

$$\ln \text{Cov}(\text{Ball}_R, \|\cdot\|_\infty, r) \leq n \ln \left\lceil \frac{R}{r} \right\rceil.$$

*Proof of Lemma 2.7.* The claim follows by a simple counting argument.  $\blacksquare$

Together with Theorem 2.6 this allows us to bound the covering number of our hypothesis class of (clipped) neural networks.

**Proposition 2.8 (covering numbers of neural network hypothesis classes).** Assume Settings 2.3 and 2.5. Let  $d \in \mathbb{N}$ ,  $\mathbf{a} \in \mathbf{A}_d$ ,  $r \in (0, 1)$ , and  $R \in [1, \infty)$ . Then it holds that

$$\begin{aligned} \ln \text{Cov}(\mathcal{N}_{\mathbf{a}, R, D}^{u, v}, r) &\leq \ln \text{Cov}(\mathcal{N}_{\mathbf{a}, R}^{u, v}, r) \\ &\leq P(\mathbf{a}) \left[ \ln \left( \frac{4L(\mathbf{a})^2 \max\{1, |u|, |v|\}}{r} \right) + L(\mathbf{a}) \ln(R\|\mathbf{a}\|_\infty) \right]. \end{aligned}$$

The proof in Appendix A.4 is based on the behavior of covering numbers under the action of a Lipschitz function, i.e.,

$$\text{Cov}(\mathcal{F}^{u, v}(\mathcal{P}_{\mathbf{a}, R}), r) \leq \text{Cov}\left(\mathcal{P}_{\mathbf{a}, R}, \frac{r}{\text{Lip}(\mathcal{F}^{u, v})}\right),$$

and uses the facts that the clipping function is nonexpansive, i.e.,  $\text{Lip}(\text{clip}_D) = 1$ , and that  $\mathcal{P}_{\mathbf{a}, R} \simeq \{\boldsymbol{\theta} \in \mathbb{R}^{P(\mathbf{a})} : \|\boldsymbol{\theta}\|_\infty \leq R\}$ .

**2.3. Analysis of the generalization error.** Combining Theorem 2.4 and Proposition 2.8, the following theorem describes our main result related to the generalization capabilities of hypothesis classes consisting of clipped ReLU networks.

**Theorem 2.9 (neural network generalization error bound).** Assume Setting 2.5. Let  $h \in \mathcal{C}((0, \infty)^5, \mathbb{R})$  satisfy for every  $x = (x_i)_{i=1}^5 \in (0, \infty)^5$  that

$$h(x) = 128D^4 x_1^2 \left[ \ln(2) + x_2 + x_3 x_4 x_5 + x_4 \ln(128D \max\{1, |u|, |v|\} x_1 x_5^2) \right],$$

let  $d, m \in \mathbb{N}$ ,  $\varepsilon, \varrho \in (0, 1)$ ,  $\mathbf{a} \in \mathbf{A}_d$ ,  $R \in [1, \infty)$  with

$$m \geq h(\varepsilon^{-1}, \ln(\varrho^{-1}), \ln(R\|\mathbf{a}\|_\infty), P(\mathbf{a}), L(\mathbf{a})),$$

and define  $\mathcal{H} := \mathcal{N}_{\mathbf{a}, R, D}^{u, v}$ . Then it holds that

$$\mathbb{P} \left[ \mathcal{E}_d(\widehat{f}_{d, m, \mathcal{H}}) - \mathcal{E}_d(f_{d, \mathcal{H}}) \leq \varepsilon \right] \geq 1 - \varrho.$$

*Proof of Theorem 2.9.* Proposition 2.8 implies that

$$\begin{aligned} m &\geq h(\varepsilon^{-1}, \ln(\varrho^{-1}), \ln(R\|\mathbf{a}\|_\infty), P(\mathbf{a}), L(\mathbf{a})) \\ &= 128D^4 \varepsilon^{-2} \left[ \ln(2\varrho^{-1}) + P(\mathbf{a}) \left( \ln(128D \max\{1, |u|, |v|\} \varepsilon^{-1} L(\mathbf{a})^2) + L(\mathbf{a}) \ln(R\|\mathbf{a}\|_\infty) \right) \right] \\ &\geq 128D^4 \varepsilon^{-2} \left[ \ln(2\varrho^{-1}) + \ln \text{Cov}\left(\mathcal{H}, \frac{\varepsilon}{32D}\right) \right]. \end{aligned}$$

Now Theorem 2.4 and a simple calculation ensures that

$$\mathbb{P} \left[ \mathcal{E}_d(\hat{f}_{d,m,\mathcal{H}}) - \mathcal{E}_d(f_{d,\mathcal{H}}) \leq \varepsilon \right] \geq 1 - 2\text{Cov} \left( \mathcal{H}, \frac{\varepsilon}{32D} \right) \exp \left( -\frac{m\varepsilon^2}{128D^4} \right) \geq 1 - \varrho,$$

and this proves the theorem.  $\blacksquare$

Next we show how Theorem 2.9 can be used to leverage bounds on the approximation error in order to obtain quantitative bounds on the generalization error.

**Corollary 2.10 (approximation implies generalization).** *Assume Setting 2.5. Let  $d \in \mathbb{N}$ ,  $\varepsilon \in (0, 1)$ ,  $\mathbf{a} \in \mathbf{A}_d$ ,  $R \in [1, \infty)$ , and  $g \in \mathcal{H} := \mathcal{N}_{\mathbf{a}, R, D}^{u,v}$  with*

$$\|g - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 \leq \varepsilon/2,$$

let  $h \in \mathcal{C}((0, \infty)^5, \mathbb{R})$  be given as in Theorem 2.9, and let  $m \in \mathbb{N}$ ,  $\varrho \in (0, 1)$  with

$$m \geq h(2\varepsilon^{-1}, \ln(\varrho^{-1}), \ln(R\|\mathbf{a}\|_\infty), P(\mathbf{a}), L(\mathbf{a})).$$

Then it holds that

$$\mathbb{P} \left[ \|\hat{f}_{d,m,\mathcal{H}} - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 \leq \varepsilon \right] \geq 1 - \varrho.$$

*Proof of Corollary 2.10.* Lemma 2.2 ensures that

$$\|f_{d,\mathcal{H}} - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 \leq \|g - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 \leq \varepsilon/2,$$

and hence  $\|\hat{f}_{d,m,\mathcal{H}} - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 \leq \mathcal{E}_d(\hat{f}_{d,m,\mathcal{H}}) - \mathcal{E}_d(f_{d,\mathcal{H}}) + \varepsilon/2$ . Theorem 2.9 (with  $\varepsilon \leftarrow \varepsilon/2$ ) now directly implies the desired claim.  $\blacksquare$

The previous result in particular implies that whenever the family  $(f_d^*)_{d \in \mathbb{N}}$  from the statistical learning problem of Setting 2.1 can be approximated by neural networks without the curse of dimensionality, then the number  $m$  of required training samples to achieve a desired accuracy with high probability does not suffer from the curse of dimensionality either. A compact version of this statement is given in the next result.

**Corollary 2.11 (approximation without curse implies generalization without curse).** *Assume Setting 2.5. Assume that there exists a polynomial  $q : \mathbb{R}^2 \rightarrow \mathbb{R}$  such that for every  $d \in \mathbb{N}$ ,  $\varepsilon \in (0, 1)$  there exist  $\mathbf{a}_{d,\varepsilon} \in \mathbf{A}_d$ ,  $R_{d,\varepsilon} \in [1, \infty)$ , and  $g_{d,\varepsilon} \in \mathcal{H}_{d,\varepsilon} := \mathcal{N}_{\mathbf{a}_{d,\varepsilon}, R_{d,\varepsilon}, D}^{u,v}$  with*

$$\max \{ \ln(R_{d,\varepsilon}), P(\mathbf{a}_{d,\varepsilon}) \} \leq q(d, \varepsilon^{-1}) \quad \text{and} \quad \|g_{d,\varepsilon} - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 \leq \varepsilon/2.$$

Then there exists a polynomial  $s : \mathbb{R}^2 \rightarrow \mathbb{R}$  such that for every  $d, m \in \mathbb{N}$ ,  $\varepsilon, \varrho \in (0, 1)$  with

$$m \geq s(d, \varepsilon^{-1})(1 + \ln(\varrho^{-1}))$$

it holds that

$$\mathbb{P} \left[ \|\hat{f}_{d,m,\mathcal{H}_{d,\varepsilon}} - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 \leq \varepsilon \right] \geq 1 - \varrho.$$

*Proof of Corollary 2.11.* Observe that for every  $d \in \mathbb{N}$ ,  $\varepsilon \in (0, 1)$  it holds that

$$\max\{\ln(\|\mathbf{a}_{d,\varepsilon}\|_\infty), L(\mathbf{a}_{d,\varepsilon})\} \leq P(\mathbf{a}_{d,\varepsilon}) \leq q(d, \varepsilon^{-1})$$

and that the function  $h \in \mathcal{C}((0, \infty)^5, \mathbb{R})$  from Theorem 2.9 satisfies for every  $x \in (0, \infty)^5$  that

$$h(x) \leq 128D^4x_1^2 \left[ 1 + x_4 \left( x_1 + x_3x_5 + 2x_5 + \ln(128D \max\{1, |u|, |v|\}) - 3 \right) \right] (1 + x_2).$$

Thus, Corollary 2.11 is a direct consequence of Corollary 2.10. ■

**3. Applications for the numerical approximation of high-dimensional PDEs.** In the present section we apply the general results of section 2 to the numerical solution of high-dimensional Kolmogorov equations.

**3.1. Kolmogorov equation as learning problem.** The following setting describes suitable Kolmogorov equations and the data for the corresponding statistical learning problems.

*Setting 3.1 (Kolmogorov equations).* Assume Setting 2.5. Let  $K \in (0, \infty)$ , for every  $d \in \mathbb{N}$  let  $\mu_d \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^d)$  (*drift coefficient*) and  $\sigma_d \in \mathcal{C}(\mathbb{R}^d, \mathbb{R}^{d \times d})$  (*diffusion coefficient*) be affine functions satisfying for every  $x \in \mathbb{R}^d$  that

$$\|\sigma_d(x)\|_2 + \|\mu_d(x)\|_2 \leq K(1 + \|x\|_2),$$

and let  $\varphi_d \in \mathcal{C}(\mathbb{R}^d, [-D, D])$  (*initial value*). Assume that  $(\varphi_d)_{d \in \mathbb{N}}$  can be approximated by neural networks in the following sense: Let  $\zeta \in [1, \infty)$  and  $\beta, \gamma, \kappa, \lambda, \nu \in [0, \infty)$ , and let

$$\mathbf{b}_{d,\varepsilon} \in \mathbf{A}_d, \quad \boldsymbol{\eta}_{d,\varepsilon} \in \mathcal{P}_{\mathbf{b}_{d,\varepsilon}}, \quad d \in \mathbb{N}, \quad \varepsilon \in (0, 1) \quad (\text{neural network approximation of } \varphi_d),$$

such that for every  $d \in \mathbb{N}$ ,  $\varepsilon \in (0, 1)$ ,  $x \in \mathbb{R}^d$  it holds that<sup>10</sup>

- (i)  $|\varphi_d(x) - \mathcal{F}(\boldsymbol{\eta}_{d,\varepsilon})(x)| \leq \varepsilon(1 + \|x\|_2^\nu)$ ,
- (ii)  $|\mathcal{F}(\boldsymbol{\eta}_{d,\varepsilon})(x)| \leq D$ ,
- (iii)  $\|\boldsymbol{\eta}_{d,\varepsilon}\|_\infty \leq \zeta d^\beta \varepsilon^{-\kappa}$ , and
- (iv)  $P(\mathbf{b}_{d,\varepsilon}) \leq \zeta d^\gamma \varepsilon^{-\lambda}$ .

Let  $T \in (0, \infty)$ , and for every  $d \in \mathbb{N}$  let  $F_d \in \mathcal{C}([0, T] \times \mathbb{R}^d, \mathbb{R})$  be the unique<sup>11</sup> function satisfying the following:

- (i)  $F_d(0, x) = \varphi_d(x)$  for every  $x \in \mathbb{R}^d$ ;
- (ii)  $F_d$  is at most polynomially growing, i.e., there exists  $\vartheta \in (0, \infty)$  such that for every  $x \in \mathbb{R}^d$  it holds that  $\max_{t \in [0, T]} F_d(t, x) \leq \vartheta(1 + \|x\|_2^\vartheta)$ ; and
- (iii)  $F_d$  is a viscosity *solution of the d-dimensional Kolmogorov equation*

$$\frac{\partial F_d}{\partial t}(t, x) = \frac{1}{2} \text{Trace}(\sigma_d(x)[\sigma_d(x)]^*(\text{Hess}_x F_d)(t, x)) + \mu_d(x) \cdot (\nabla_x F_d)(t, x)$$

for every  $(t, x) \in (0, T) \times \mathbb{R}^d$ .

<sup>10</sup>Due to the boundedness assumption on  $\varphi_d$  one can obtain the desired estimates in items (i) and (ii) by adapting known neural network approximation results; see [14] and Appendix A.2.

<sup>11</sup>For a proof see [29, Proposition 3.4(i)].

Let the probability space  $(\Omega, \mathcal{G}, \mathbb{P})$  be equipped with a filtration  $(\mathcal{G}_t)_{t \in [0, T]}$  which fulfills the usual conditions. For every  $d \in \mathbb{N}$  let  $(B_t^d)_{t \in [0, T]} : [0, T] \times \Omega \rightarrow \mathbb{R}^d$  be a  $d$ -dimensional  $(\mathcal{G}_t)$ -Brownian motion, and for every  $\mathcal{G}_0$ -measurable random variable  $\chi : \Omega \rightarrow \mathbb{R}^d$  denote by

$$(S_t^\chi)_{t \in [0, T]} : [0, T] \times \Omega \rightarrow \mathbb{R}^d \quad (\text{SDE solution process with initial value } \chi)$$

the unique  $(\mathcal{G}_t)$ -adapted stochastic process<sup>12</sup> with continuous sample paths satisfying the SDE

$$dS_t^\chi = \sigma_d(S_t^\chi) dB_t^d + \mu_d(S_t^\chi) dt \quad \text{and} \quad S_0^\chi = \chi$$

$\mathbb{P}$ -a.s. for every  $t \in [0, T]$ . For every  $d \in \mathbb{N}$  let the *input data*  $X_d : \Omega \rightarrow [u, v]^d$  be  $\mathcal{G}_0$ -measurable and uniformly distributed on  $[u, v]^d$  and define the *label* by  $Y_d := \varphi_d(S_T^{X_d})$ .

The next result shows that computing the end value  $[u, v]^d \ni x \mapsto F_d(T, x)$  of the solution to the Kolmogorov equation can be restated as a learning problem.

**Lemma 3.2 (Kolmogorov equation as learning problem).** *Assume Setting 3.1 and let  $d \in \mathbb{N}$ . Then for a.e.  $x \in [u, v]^d$  it holds that*

$$F_d(T, x) = f_d^*(x).$$

The result is based on work from [10] and the following formal calculation:

$$F_d(T, x) = \mathbb{E}[\varphi_d(S_T^x)] = \mathbb{E}[\varphi_d(S_T^{X_d}) | X_d = x] = \mathbb{E}[Y_d | X_d = x] = f_d^*(x)$$

for a.e.  $x \in [u, v]^d$ ; see Appendix A.5 for a rigorous proof.

**3.2. Neural network generalization results for solutions of Kolmogorov equations.** We first show that the end value  $[u, v]^d \ni x \mapsto F_d(T, x)$  of the solution to the Kolmogorov equation can be approximated by hypothesis classes consisting of clipped ReLU networks.

**Theorem 3.3 (neural network regularity result for Kolmogorov equations).** *Assume Setting 3.1. Then there exist  $C, c \in (0, \infty)$  such that the following holds: For every  $d \in \mathbb{N}$ ,  $\varepsilon \in (0, 1)$  there exist  $\mathbf{a} \in \mathbf{A}_d$ ,  $R \in [1, \infty)$ , and  $g \in \mathcal{N}_{\mathbf{a}, R, D}^{u, v}$  such that it holds that*

- (i)  $\frac{1}{(v-u)^d} \|g - F_d(T, \cdot)\|_{\mathcal{L}^2([u, v]^d)}^2 \leq \varepsilon$ ,
- (ii)  $P(\mathbf{a}) \leq Cd^{\nu\lambda/2+\gamma\varepsilon^{-\lambda/2-2}}$ ,
- (iii)  $R \leq Cd^{(\nu\kappa+3)/2+\beta\varepsilon^{-\kappa/2-1}}$ ,
- (iv)  $L(\mathbf{a}) = L(\mathbf{b}_{d, cd^{-\nu/2}\varepsilon^{1/2}})$ , and
- (v)  $\|\mathbf{a}\|_\infty \leq C\varepsilon^{-1}\|\mathbf{b}_{d, cd^{-\nu/2}\varepsilon^{1/2}}\|_\infty$ .

Except for property (iii) a similar result was shown in [29, Corollary 3.13]. We present the proof in Appendix A.6 and briefly sketch the idea in the following. First we observe that in our case of affine coefficients  $\sigma_d$  and  $\mu_d$  there exist random variables  $\mathfrak{M}$  and  $\mathfrak{N}$  such that

---

<sup>12</sup>The solution process is unique up to indistinguishability; see, for instance, [5, Theorem 6.2.2].

for all  $x \in \mathbb{R}^d$  it holds  $\mathbb{P}$ -a.s. that  $S_T^x = \mathfrak{M}x + \mathfrak{N}$ ; see Lemma A.3. Let  $((\mathfrak{M}^{(j)}, \mathfrak{N}^{(j)}))_{j \in \mathbb{N}}$  be i.i.d. samples with  $(\mathfrak{M}^{(1)}, \mathfrak{N}^{(1)}) \sim (\mathfrak{M}, \mathfrak{N})$ . Then for fixed  $x \in \mathbb{R}^d$  the mean squared error

$$\mathbb{E} \left[ \left( F_d(T, x) - \frac{1}{n} \sum_{j=1}^n \mathcal{F}(\boldsymbol{\eta}_{d,\delta})(\mathfrak{M}^{(j)}x + \mathfrak{N}^{(j)}) \right)^2 \right]$$

can be decomposed into the sum of the squared bias and the variance, i.e.,

$$\underbrace{\mathbb{E} \left[ \varphi_d(S_T^x) - \mathcal{F}(\boldsymbol{\eta}_{d,\delta})(S_T^x) \right]^2}_{\mathcal{O}(\delta^2)} + \underbrace{\mathbb{V} \left[ \frac{1}{n} \sum_{j=1}^n \mathcal{F}(\boldsymbol{\eta}_{d,\delta})(\mathfrak{M}^{(j)}x + \mathfrak{N}^{(j)}) \right]}_{\mathcal{O}(n^{-1})},$$

where we used the Feynman–Kac formula, our assumptions, and properties of Monte Carlo sampling. With more effort one can prove analogous estimates in the  $\mathcal{L}^2(\mathbb{P}_{X_d})$ -norm, and our setting implies that  $\mathbb{P}_{X_d} = \frac{1}{(v-u)^d} \lambda_{[u,v]^d}$ , where  $\lambda_{[u,v]^d}$  denotes the Lebesgue measure on  $[u, v]^d$ . This suggests that, given  $\varepsilon \in (0, 1)$ , for sufficient large  $n \in \mathbb{N}$  and small  $\delta \in (0, 1)$  there exists an outcome  $\omega \in \Omega$  such that with  $M^{(j)} := \mathfrak{M}^{(j)}(\omega)$  and  $N^{(j)} := \mathfrak{N}^{(j)}(\omega)$  it holds that

$$\frac{1}{(v-u)^d} \int_{[u,v]^d} \left( F_d(T, x) - \frac{1}{n} \sum_{j=1}^n \mathcal{F}(\boldsymbol{\eta}_{d,\delta})(M^{(j)}x + N^{(j)}) \right)^2 dx \leq \varepsilon.$$

In Lemma A.4 we specify a network architecture  $\mathbf{a} \in \mathbf{A}_d$  and a parametrization  $\boldsymbol{\theta} \in \mathcal{P}_{\mathbf{a}}$  such that for every  $x \in \mathbb{R}^d$  it holds that

$$\mathcal{F}(\boldsymbol{\theta})(x) = \frac{1}{n} \sum_{j=1}^n \mathcal{F}(\boldsymbol{\eta}_{d,\delta})(M^{(j)}x + N^{(j)}),$$

and we bound the parameter magnitudes of  $\boldsymbol{\theta}$  with the help of Lemma A.3.

Observe that the approximation result in Theorem 3.3 does not underlie the curse of dimensionality, and by Corollary 2.10 we can establish a generalization result that is free of the curse of dimensionality.

**Theorem 3.4 (neural network generalization result for Kolmogorov equations).** *Assume Setting 3.1 and let  $h \in \mathcal{C}((0, \infty)^5, \mathbb{R})$  be given as in Theorem 2.9. Then there exist  $C, c \in (0, \infty)$  such that the following holds: For every  $d, m \in \mathbb{N}$ ,  $\varepsilon, \varrho \in (0, 1)$  with*

$$m \geq h(2\varepsilon^{-1}, \ln(\varrho^{-1}), \ln(R\|\mathbf{a}\|_\infty), P(\mathbf{a}), L(\mathbf{a}))$$

there exist  $\mathbf{a} \in \mathbf{A}_d$  and  $R \in [1, \infty)$  such that it holds that

- (i)  $\mathbb{P} \left[ \frac{1}{(v-u)^d} \|\widehat{f}_{d,m,\mathcal{H}} - F_d(T, \cdot)\|_{\mathcal{L}^2([u,v]^d)}^2 \leq \varepsilon \right] \geq 1 - \varrho,$
- (ii)  $P(\mathbf{a}) \leq Cd^{\nu\lambda/2+\gamma}\varepsilon^{-\lambda/2-2},$
- (iii)  $R \leq Cd^{(\nu\kappa+3)/2+\beta}\varepsilon^{-\kappa/2-1},$
- (iv)  $L(\mathbf{a}) = L(\mathbf{b}_{d,cd^{-\nu/2}\varepsilon^{1/2}})$ , and

(v)  $\|\mathbf{a}\|_\infty \leq C\varepsilon^{-1} \|\mathbf{b}_{d,cd-\nu/2\varepsilon^{1/2}}\|_\infty$ ,  
 where  $\mathcal{H} = \mathcal{N}_{\mathbf{a},R,D}^{u,v}$ .

*Proof of Theorem 3.4.* This is a direct consequence of Theorem 3.3 (with  $\varepsilon \leftarrow \varepsilon/2$ ) and Corollary 2.10.  $\blacksquare$

We can also reformulate this in a more compact form.

**Corollary 3.5 (ERM for Kolmogorov equations without curse).** *Assume Setting 3.1. Then there exists a polynomial  $p: \mathbb{R}^2 \rightarrow \mathbb{R}$  such that the following holds: For every  $d, m \in \mathbb{N}$ ,  $\varepsilon, \varrho \in (0, 1)$  with*

$$m \geq p(d, \varepsilon^{-1})(1 + \ln(\varrho^{-1}))$$

*there exist  $\mathbf{a} \in \mathbf{A}_d$  and  $R \in [1, \infty)$  such that it holds that*

- (i)  $\mathbb{P} \left[ \frac{1}{(v-u)^d} \|\widehat{f}_{d,m,\mathcal{H}} - F_d(T, \cdot)\|_{\mathcal{L}^2([u,v]^d)}^2 \leq \varepsilon \right] \geq 1 - \varrho$  and
- (ii)  $\max\{R, P(\mathbf{a})\} \leq p(d, \varepsilon^{-1})$ ,

where  $\mathcal{H} = \mathcal{N}_{\mathbf{a},R,D}^{u,v}$ .

*Proof of Corollary 3.5.* This follows directly from Theorem 3.3 and Corollary 2.11.  $\blacksquare$

**3.3. Pricing of high-dimensional options.** The proof of Theorem 1.1 from the introductory section dealing with the pricing of high-dimensional European put options is now an easy consequence of the above theory.

*Proof of Theorem 1.1.* We first show that the approximation of  $(\varphi_d)_{d \in \mathbb{N}}$  by clipped neural networks according to Setting 3.1 is possible. Note that for every  $z \in \mathbb{R}$  it holds that  $\min\{z, D\} = D - \text{ReLU}_*(D - z)$ . This implies that for every  $d \in \mathbb{N}$ ,  $x \in \mathbb{R}^d$  it holds that

$$\varphi_d(x) = \min \{ \max \{D - c_d \cdot x, 0\}, D\} = \mathcal{F}(\boldsymbol{\eta}_d)(x),$$

where

$$\boldsymbol{\eta}_d := ((-[c_d]^*, D), (-1, D), (-1, D)) \in \mathcal{P}_{(d,1,1,1),D}.$$

Accordingly, Setting 3.1 is satisfied with

$$\zeta = \max\{D, 6\}, \quad \gamma = 1, \quad \beta = \kappa = \lambda = \nu = 0, \quad \mathbf{b}_{d,\varepsilon} = (d, 1, 1, 1), \quad \boldsymbol{\eta}_{d,\varepsilon} = \boldsymbol{\eta}_d.$$

Now Theorem 3.4 and a straightforward calculation prove the claim.  $\blacksquare$

**Appendix A. Proofs.** This appendix contains various proofs and additional material omitted from the main text.

### A.1. Bias-variance decomposition.

*Proof of Lemma 2.2.* For every  $f \in \mathcal{L}^2(\mathbb{P}_{X_d})$  it holds that

$$\begin{aligned} \mathcal{E}_d(f) &= \mathbb{E} \left[ (f(X_d) - f_d^*(X_d) + f_d^*(X_d) - Y_d)^2 \right] \\ (A.1) \quad &= \mathbb{E} \left[ (f(X_d) - f_d^*(X_d))^2 \right] + \mathbb{E} \left[ (f_d^*(X_d) - Y_d)^2 \right] \\ &\quad + 2\mathbb{E} \left[ (f(X_d) - f_d^*(X_d))(f_d^*(X_d) - Y_d) \right]. \end{aligned}$$

Observe that, due to the fact that it holds  $\mathbb{P}$ -a.s. that  $f_d^*(X_d) = \mathbb{E}[Y_d|X_d]$ , the tower property of the conditional expectation establishes for every  $f \in \mathcal{L}^2(\mathbb{P}_{X_d})$  that

$$\begin{aligned}\mathbb{E}[(f(X_d) - f_d^*(X_d))(f_d^*(X_d) - Y_d)] &= \mathbb{E}\left[\mathbb{E}\left[(f(X_d) - f_d^*(X_d))(f_d^*(X_d) - Y_d) \middle| X_d\right]\right] \\ &= \mathbb{E}[(f(X_d) - f_d^*(X_d))(f_d^*(X_d) - \mathbb{E}[Y_d|X_d])] = 0,\end{aligned}$$

which, together with (A.1), implies that

$$(A.2) \quad \|f - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 = \mathbb{E}\left[(f(X_d) - f_d^*(X_d))^2\right] = \mathcal{E}_d(f) - \mathcal{E}_d(f_d^*).$$

This proves items (i) and (ii) and shows that it holds that

$$\|\widehat{f}_{d,m,\mathcal{H}} - f_d^*\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}^2 = \mathcal{E}_d(\widehat{f}_{d,m,\mathcal{H}}) - \mathcal{E}_d(f_{d,\mathcal{H}}) + \mathcal{E}_d(f_{d,\mathcal{H}}) - \mathcal{E}_d(f_d^*).$$

Finally, applying (A.2) (with  $f \leftarrow f_{d,\mathcal{H}}$ ) proves the lemma. ■

**A.2. Clipped neural networks are standard neural networks.** We show that “clipped” neural network hypothesis classes  $\mathcal{N}_{\mathbf{a},R,D}^{u,v}$  are in fact subsets of “nonclipped” ones.

**Lemma A.1** (clipping function as neural network). *Assume Setting 2.5 and let*

$$\boldsymbol{\theta} := \left( \left( \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right), \left( \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} D \\ D \end{bmatrix} \right), ([[-1 \ 1], 0]) \right) \in \mathcal{P}_{(1,2,2,1),D}.$$

*Then it holds that*

$$\text{clip}_D = \mathcal{F}(\boldsymbol{\theta}).$$

*Proof of Lemma A.1.* A case distinction establishes that for every  $x \in \mathbb{R}$  it holds that

$$\mathcal{F}(\boldsymbol{\theta})(x) = -\text{ReLU}_*(D - \text{ReLU}_*(x)) + \text{ReLU}_*(-\text{ReLU}_*(-x) + D) = \text{clip}_D(x),$$

which proves the claim. ■

**Corollary A.2 (clipped neural networks are standard neural networks).** *Assume Setting 2.5. Let  $d \in \mathbb{N}$ ,  $R \in [D, \infty)$ , and  $\mathbf{a} = (a_0, a_1, \dots, a_L) \in \mathbf{A}_d$ , and define  $\mathbf{b} := (a_0, a_1, \dots, a_L, 2, 2, 1) \in \mathbf{A}_d$ . Then it holds that*

$$\mathcal{N}_{\mathbf{a},R,D}^{u,v} \subseteq \mathcal{N}_{\mathbf{b},R}^{u,v}.$$

*Proof of Corollary A.2.* The proof follows by the representation of the clipping function in Lemma A.1 and the fact that composition with a neural network does not change the magnitude of its parameters;<sup>13</sup> see also [53, Definition 2.2] for a formal definition. ■

---

<sup>13</sup>Because of that we did not choose the easier representation  $\text{clip}_D(x) = -D + \text{ReLU}_*(2D - \text{ReLU}_*(D - x))$ .

### A.3. Lipschitz continuity of the realization map.

*Proof of Theorem 2.6.* Define  $L := L(\mathbf{a})$  and  $\mathfrak{m} := \max\{1, |u|, |v|\}$ . We will show the following stronger statement. For every  $\boldsymbol{\theta}, \boldsymbol{\eta} \in \mathcal{P}_{\mathbf{a}, R}$  it holds that

$$(A.3) \quad \|\mathcal{F}^{u,v}(\boldsymbol{\theta}) - \mathcal{F}^{u,v}(\boldsymbol{\eta})\|_{\mathcal{L}^\infty} \leq \left[ \mathfrak{m}LR^{L-1}\|\mathbf{a}\|_\infty^L + \sum_{l=1}^L l(R\|\mathbf{a}\|_\infty)^{l-1} \right] \|\boldsymbol{\theta} - \boldsymbol{\eta}\|_\infty.$$

This directly implies the statement of Theorem 2.6, as it holds that

$$\sum_{l=1}^L l(R\|\mathbf{a}\|_\infty)^{l-1} \leq L^2(R\|\mathbf{a}\|_\infty)^{L-1} \leq \mathfrak{m}L^2R^{L-1}\|\mathbf{a}\|_\infty^L.$$

For the proof of (A.3) let us fix  $\boldsymbol{\theta}, \boldsymbol{\eta} \in \mathcal{P}_{\mathbf{a}, R}$  given by

$$\boldsymbol{\theta} = ((W_l, B_l))_{l=1}^L \quad \text{and} \quad \boldsymbol{\eta} = ((V_l, A_l))_{l=1}^L.$$

Let  $r := \|\boldsymbol{\theta} - \boldsymbol{\eta}\|_\infty$ , and for every  $s \in \{1, \dots, L\}$  define the partial parametrizations

$$\boldsymbol{\theta}(s) = ((W_l, B_l))_{l=1}^s \quad \text{and} \quad \boldsymbol{\eta}(s) = ((V_l, A_l))_{l=1}^s,$$

the partial realization functions  $f_s := \mathcal{F}^{u,v}(\boldsymbol{\theta}(s))$  and  $g_s := \mathcal{F}^{u,v}(\boldsymbol{\eta}(s))$ , the partial errors<sup>14</sup>

$$\epsilon_0 := 0 \quad \text{and} \quad \epsilon_s := \|f_s - g_s\|_{\mathcal{L}^\infty},$$

and the partial maxima

$$\mathfrak{m}_0 := \mathfrak{m} = \max\{1, |u|, |v|\} \quad \text{and} \quad \mathfrak{m}_s := \max\{1, \|f_s\|_{\mathcal{L}^\infty}, \|g_s\|_{\mathcal{L}^\infty}\}.$$

We are interested in estimating the error  $\epsilon_L$  and try to bound  $\mathfrak{m}_s$  relative to  $\mathfrak{m}_{s-1}$  and  $\epsilon_s$  relative to  $\epsilon_{s-1}$ . Note that for every  $s \in \{2, \dots, L\}$  it holds that

$$\|f_s\|_{\mathcal{L}^\infty} = \|W_s \text{ReLU}_*(f_{s-1}) + B_s\|_{\mathcal{L}^\infty} \leq R\|\mathbf{a}\|_\infty \mathfrak{m}_{s-1} + R.$$

Analogous computations for the case  $s = 1$  and the function  $g_s$  establish that for every  $s \in \{1, \dots, L\}$  it holds that  $\mathfrak{m}_s \leq R\|\mathbf{a}\|_\infty \mathfrak{m}_{s-1} + R$ . By induction this implies that

$$(A.4) \quad \mathfrak{m}_s \leq \mathfrak{m}(R\|\mathbf{a}\|_\infty)^s + R \sum_{l=0}^{s-1} (R\|\mathbf{a}\|_\infty)^l$$

for every  $s \in \{1, \dots, L\}$ . Moreover, note that for every  $s \in \{2, \dots, L\}$  it holds that

$$\begin{aligned} \epsilon_s &= \| [W_s \text{ReLU}_*(f_{s-1}) + B_s] - [V_s \text{ReLU}_*(g_{s-1}) + A_s] \|_{\mathcal{L}^\infty} \\ &\leq \| [W_s - V_s] \text{ReLU}_*(f_{s-1}) \|_{\mathcal{L}^\infty} + \| V_s [\text{ReLU}_*(f_{s-1}) - \text{ReLU}_*(g_{s-1})] \|_{\mathcal{L}^\infty} + r \\ &\leq \|\mathbf{a}\|_\infty (r\mathfrak{m}_{s-1} + R\epsilon_{s-1}) + r. \end{aligned}$$

Together with (A.4), one proves by induction that for every  $s \in \{1, 2, \dots, L\}$  it holds that

$$\epsilon_s \leq \left[ \mathfrak{m}sR^{s-1}\|\mathbf{a}\|_\infty^s + \sum_{l=1}^s l(R\|\mathbf{a}\|_\infty)^{l-1} \right] r.$$

Setting  $s = L$  proves the claim in (A.3). ■

<sup>14</sup>For vector-valued functions  $f \in \mathcal{C}([u, v]^d, \mathbb{R}^n)$  we define the uniform norm on  $[u, v]^d$  by  $\|f\|_{\mathcal{L}^\infty} = \|f\|_{\mathcal{L}^\infty([u, v]^d)} := \max_{x \in [u, v]^d} \|f(x)\|_\infty$ .

#### A.4. Covering numbers of neural network hypothesis classes.

*Proof of Proposition 2.8.* To simplify the notation we define  $\mathfrak{m} := \max\{1, |u|, |v|\}$ ,

$$\Delta := \frac{r}{2\mathfrak{m}L(\mathbf{a})^2R^{L(\mathbf{a})-1}\|\mathbf{a}\|_{\infty}^{L(\mathbf{a})}}, \quad \text{and} \quad N := \text{Cov}(\mathcal{P}_{\mathbf{a},R}, \|\cdot\|_{\infty}, \Delta).$$

Choose  $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_N \in \mathcal{P}_{\mathbf{a},R}$  such that for every  $\boldsymbol{\theta} \in \mathcal{P}_{\mathbf{a},R}$  there exists  $i \in \{1, 2, \dots, N\}$  with  $\|\boldsymbol{\theta} - \boldsymbol{\theta}_i\|_{\infty} \leq \Delta$ , which by Theorem 2.6 and the fact that  $\text{clip}_D$  is nonexpansive implies that

$$\begin{aligned} \|\text{clip}_D \circ \mathcal{F}^{u,v}(\boldsymbol{\theta}) - \text{clip}_D \circ \mathcal{F}^{u,v}(\boldsymbol{\theta}_i)\|_{\mathcal{L}^{\infty}} &\leq \|\mathcal{F}^{u,v}(\boldsymbol{\theta}) - \mathcal{F}^{u,v}(\boldsymbol{\theta}_i)\|_{\mathcal{L}^{\infty}} \\ &\leq 2\mathfrak{m}L(\mathbf{a})^2R^{L(\mathbf{a})-1}\|\mathbf{a}\|_{\infty}^{L(\mathbf{a})}\|\boldsymbol{\theta} - \boldsymbol{\theta}_i\|_{\infty} \leq r. \end{aligned}$$

Lemma 2.7 and identifying  $\mathcal{P}_{\mathbf{a},R} \simeq \{\boldsymbol{\theta} \in \mathbb{R}^{P(\mathbf{a})} : \|\boldsymbol{\theta}\|_{\infty} \leq R\}$  hence show that

$$\begin{aligned} \ln \text{Cov}(\mathcal{N}_{\mathbf{a},R,D}^{u,v}, \|\cdot\|_{\mathcal{L}^{\infty}}, r) &\leq \ln \text{Cov}(\mathcal{N}_{\mathbf{a},R}^{u,v}, \|\cdot\|_{\mathcal{L}^{\infty}}, r) \leq \ln \text{Cov}(\mathcal{P}_{\mathbf{a},R}, \|\cdot\|_{\infty}, \Delta) \\ &\leq P(\mathbf{a}) \ln \left( \left\lceil \frac{R}{\Delta} \right\rceil \right) \leq P(\mathbf{a}) \ln \left( \frac{2R}{\Delta} \right), \end{aligned}$$

and this proves the proposition. ■

#### A.5. Kolmogorov equation as learning problem.

*Proof of Lemma 3.2.* The proof is based on the Feynman–Kac formula for viscosity solutions of Kolmogorov equations, which states that for every  $x \in \mathbb{R}^d$  it holds that

$$(A.5) \quad F_d(T, x) = \mathbb{E}[\varphi_d(S_T^x)];$$

see [29, Corollary 2.23(ii)]. We claim that for every  $A \in \mathcal{B}([u, v]^d)$  it holds that

$$\mathbb{E}[\mathbb{1}_A(X_d)\varphi_d(S_T^{X_d})] = \int_A \mathbb{E}[\varphi_d(S_T^x)] d\mathbb{P}_{X_d}(x).$$

This would prove the lemma as it implies that for  $\mathbb{P}_{X_d}$ -a.s.  $x \in [u, v]^d$  it holds that

$$\mathbb{E}[\varphi_d(S_T^{X_d})|X_d = x] = \mathbb{E}[\varphi_d(S_T^x)],$$

which by (A.5) and Setting 3.1 ensures that for a.e.  $x \in [u, v]^d$  it holds that

$$f_d^*(x) = \mathbb{E}[Y_d|X_d = x] = \mathbb{E}[\varphi_d(S_T^{X_d})|X_d = x] = \mathbb{E}[\varphi_d(S_T^x)] = F_d(T, x).$$

For the proof of the claim let us fix  $A \in \mathcal{B}([u, v]^d)$  and let  $g_{\varepsilon} \in \mathcal{C}^{\infty}(\mathbb{R}^d, \mathbb{R})$ ,  $\varepsilon \in (0, 1)$ , be a family of mollifiers. For every  $\varepsilon \in (0, 1)$  we define the convolution with the indicator function  $\mathbb{1}_{A,\varepsilon} := \mathbb{1}_A * g_{\varepsilon} \in \mathcal{C}^{\infty}(\mathbb{R}^d, \mathbb{R})$  and the continuous and bounded mapping

$$\Phi_{\varepsilon} : \begin{cases} \mathcal{C}([0, T], \mathbb{R}^d) &\rightarrow \mathbb{R} \\ f &\mapsto \mathbb{1}_{A,\varepsilon}(f(0))\varphi_d(f(T)). \end{cases}$$

In [10, Lemma 2.6(v)] it is shown that for every  $\varepsilon \in (0, 1)$  it holds that

$$\mathbb{E}[\mathbb{1}_{A,\varepsilon}(X_d)\varphi_d(S_T^{X_d})] = \frac{1}{(v-u)^d} \int_{[u,v]^d} \mathbb{1}_{A,\varepsilon}(x) \mathbb{E}[\varphi_d(S_T^x)] dx.$$

The fact that  $\lim_{\varepsilon \rightarrow 0} \mathbb{1}_{A,\varepsilon}(x) = \mathbb{1}_A(x)$  for a.e.  $x \in \mathbb{R}^d$  (see, for instance, [24, Appendix C.5]) and the dominated convergence theorem prove the claim when letting  $\varepsilon$  tend to zero. ■

**A.6. Neural network approximation result for solutions of Kolmogorov equations.** The proof of Theorem 3.3 is given after the following two auxiliary lemmas. First, we show that given an SDE with affine coefficients  $\sigma_d$  and  $\mu_d$ , its solution  $S_T^x$  also admits a (random) affine representation.

**Lemma A.3 (representation of SDE solutions).** *Assume Setting 3.1. Let  $d \in \mathbb{N}$ , let  $e_i \in \mathbb{R}^d$ ,  $i \in \{1, \dots, d\}$ , be the standard basis in  $\mathbb{R}^d$ , for every  $p, z \in [0, \infty)$  let*

$$\mathfrak{c}_p(z) := 2^{p/2} (z + KT + \max\{2, p\}K\sqrt{T})^p \exp(pK^2T[\sqrt{T} + \max\{2, p\}]^2),$$

and define the random variables  $\mathfrak{M} : \Omega \rightarrow \mathbb{R}^{d \times d}$  and  $\mathfrak{N} : \Omega \rightarrow \mathbb{R}^d$  by

$$\mathfrak{M} := [S_T^{e_1} - S_T^0 \quad S_T^{e_2} - S_T^0 \quad \dots \quad S_T^{e_d} - S_T^0] \quad \text{and} \quad \mathfrak{N} := S_T^0.$$

Then for every  $x \in \mathbb{R}^d$  it holds  $\mathbb{P}$ -a.s. that  $S_T^x = \mathfrak{M}x + \mathfrak{N} = \mathcal{A}_{\mathfrak{M}, \mathfrak{N}}(x)$  and it holds that<sup>15</sup>

- (i)  $\mathbb{E}[\|\mathfrak{M}\|_2 + \|\mathfrak{N}\|_2] \leq 3\mathfrak{c}_1(1)d$  and
- (ii)  $\|\mathbb{E}[\|\mathcal{A}_{\mathfrak{M}, \mathfrak{N}}\|_2^\nu]\|_{\mathcal{L}^2(\mathbb{P}_{X_d})} \leq \mathfrak{c}_\nu(\max\{1, |u|, |v|\})d^{\nu/2}$ .

*Proof of Lemma A.3.* A proof of the first claim can be found in [29, Lemmas 2.7 and 2.15]. For the proof of items (i) and (ii) note that for every  $p \in [0, \infty)$ ,  $x \in \mathbb{R}^d$  it holds that

$$\mathbb{E}[\|\mathcal{A}_{\mathfrak{M}, \mathfrak{N}}(x)\|_2^p] = \mathbb{E}[\|S_T^x\|_2^p] \leq \left(\mathbb{E}[\|S_T^x\|_2^{\max\{2,p\}}]\right)^{p/\max\{2,p\}} \leq \mathfrak{c}_p(\|x\|_2);$$

see [29, Proposition 2.14]. Together with the facts that it holds that

$$\mathbb{E}[\|\mathfrak{M}\|_2 + \|\mathfrak{N}\|_2] \leq \mathbb{E}\left[\|S_T^0\|_2 + \sum_{i=1}^d \|S_T^{e_i} - S_T^0\|_2\right] \leq (d+1)\mathbb{E}[\|S_T^0\|_2] + \sum_{i=1}^d \mathbb{E}[\|S_T^{e_i}\|_2]$$

and that

$$\|\mathbb{E}[\|\mathcal{A}_{\mathfrak{M}, \mathfrak{N}}\|_2^\nu]\|_{\mathcal{L}^2(\mathbb{P}_{X_d})} \leq \left(\int_{[u,v]^d} [\mathfrak{c}_\nu(\|x\|_2)]^2 d\mathbb{P}_{X_d}(x)\right)^{1/2} \leq \mathfrak{c}_\nu(\sqrt{d} \max\{1, |u|, |v|\}),$$

this implies the desired estimates. ■

In the next lemma we show that the average of the composition of a neural network with different affine functions can be represented by a single neural network, and we bound the number and size of its parameters.

**Lemma A.4 (compositions of neural networks and affine functions).** *Assume Setting 2.5. Let  $d, n \in \mathbb{N}$ ,  $\mathbf{b} \in \mathbf{A}_d$ ,  $\boldsymbol{\eta} \in \mathcal{P}_{\mathbf{b}}$ , and*

$$((M^{(j)}, N^{(j)}))_{j=1}^n \in (\mathbb{R}^{d \times d} \times \mathbb{R}^d)^n.$$

Then there exist  $\mathbf{a} \in \mathbf{A}_d$  and  $\boldsymbol{\theta} \in \mathcal{P}_{\mathbf{a}}$  such that it holds that

---

<sup>15</sup>Recall that for a matrix  $M \in \mathbb{R}^{d \times d}$  we denote by  $\|M\|_2 := (\sum_{i,j=1}^d M_{ij}^2)^{1/2}$  its Frobenius norm.

- (i)  $\mathcal{F}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{j=1}^n \mathcal{F}(\boldsymbol{\eta}) \circ \mathcal{A}_{M^{(j)}, N^{(j)}},$
- (ii)  $P(\mathbf{a}) \leq n^2 P(\mathbf{b}),$
- (iii)  $\|\boldsymbol{\theta}\|_\infty \leq \sqrt{d} \|\boldsymbol{\eta}\|_\infty \max_{j=1}^n (\|M^{(j)}\|_2 + \|N^{(j)}\|_2 + 1),$
- (iv)  $L(\mathbf{a}) = L(\mathbf{b}),$  and
- (v)  $\|\mathbf{a}\|_\infty = n \|\mathbf{b}\|_\infty.$

*Proof of Lemma A.4.* With the exception of item (iii) this result is proven in [29, Lemma 3.8]. There it is shown that for  $\boldsymbol{\eta} = ((V_l, A_l))_{l=1}^L$  a suitable parametrization  $\boldsymbol{\theta} = ((W_l, B_l))_{l=1}^L$  is given by  $W_L := \begin{bmatrix} \frac{1}{n} V_L & \frac{1}{n} V_L & \dots & \frac{1}{n} V_L \end{bmatrix}$ ,  $B_L := A_L$ ,

$$W_1 := \begin{bmatrix} V_1 M^{(1)} \\ \vdots \\ V_1 M^{(n)} \end{bmatrix}, \quad B_1 := \begin{bmatrix} V_1 N^{(1)} + A_1 \\ \vdots \\ V_1 N^{(n)} + A_1 \end{bmatrix}, \quad \text{and} \quad W_l := \begin{bmatrix} V_l & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & V_l \end{bmatrix}, \quad B_l := \begin{bmatrix} A_l \\ \vdots \\ A_l \end{bmatrix},$$

$l \in \{2, \dots, L-1\}$ . Now observe that

$$\|W_1\|_\infty \leq \sqrt{d} \|\boldsymbol{\eta}\|_\infty \max_{j=1}^n \|M^{(j)}\|_2 \quad \text{and} \quad \|B_1\|_\infty \leq \sqrt{d} \|\boldsymbol{\eta}\|_\infty \max_{j=1}^n (\|N^{(j)}\|_2 + 1),$$

which proves the lemma.  $\blacksquare$

Now we are ready to prove Theorem 3.3.

*Proof of Theorem 3.3.* Fix  $d \in \mathbb{N}$ ,  $\varepsilon \in (0, 1)$  and define  $\mathfrak{m} := \max\{1, |u|, |v|\}$ . Let  $\mathfrak{M}$ ,  $\mathfrak{N}$ ,  $\mathfrak{c}_1(1)$ , and  $\mathfrak{c}_\nu(\mathfrak{m})$  be given as in Lemma A.3, let  $((\mathfrak{M}^{(j)}, \mathfrak{N}^{(j)}))_{j \in \mathbb{N}}$  be i.i.d. random variables with  $(\mathfrak{M}^{(1)}, \mathfrak{N}^{(1)}) \sim (\mathfrak{M}, \mathfrak{N})$ , and let

$$(A.6) \quad n \in [16D^2\varepsilon^{-1}, 32D^2\varepsilon^{-1}] \cap \mathbb{N} \quad \text{and} \quad \delta := (8\mathfrak{c}_\nu(\mathfrak{m}))^{-1} d^{-\nu/2} \varepsilon^{1/2}.$$

Define  $g := \mathcal{F}(\boldsymbol{\eta}_{d,\delta})$  and note that Setting 3.1, Lemma A.3, and the Feynman–Kac formula [29, Corollary 2.23(ii)] establish that for every  $x \in \mathbb{R}^d$  it holds that

$$(A.7) \quad |g(x)| \leq D, \quad |\varphi_d(x) - g(x)| \leq \delta(1 + \|x\|_2^\nu), \quad \text{and} \quad F_d(T, x) = \mathbb{E} [(\varphi_d \circ \mathcal{A}_{\mathfrak{M}, \mathfrak{N}})(x)].$$

We now use techniques from [29, Proof of Proposition 3.4] to show that the random variable  $G: \Omega \rightarrow [0, \infty)$ , given by

$$G := \varepsilon^{-1/2} \underbrace{\left\| F_d(T, \cdot) - \frac{1}{n} \sum_{j=1}^n g \circ \mathcal{A}_{\mathfrak{M}^{(j)}, \mathfrak{N}^{(j)}} \right\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}}_{:= G_1} + \underbrace{(6\mathfrak{c}_1(1)dn)^{-1} \max_{j=1}^n (\|\mathfrak{M}^{(j)}\|_2 + \|\mathfrak{N}^{(j)}\|_2)}_{:= G_2},$$

satisfies  $\mathbb{E}[G] \leq 1$ . First, note that (A.6) and (A.7), together with Jensen's inequality, Fubini's

theorem, the Bienaymé formula (see also [29, Lemma 2.3]), and Lemma A.3, ensure that

$$\begin{aligned}
\mathbb{E}[G_1] &\leq \left\| \mathbb{E}[(\varphi_d - g) \circ \mathcal{A}_{\mathfrak{M}, \mathfrak{N}}] \right\|_{\mathcal{L}^2(\mathbb{P}_{X_d})} + \mathbb{E} \left[ \left\| \mathbb{E}[g \circ \mathcal{A}_{\mathfrak{M}, \mathfrak{N}}] - \frac{1}{n} \sum_{j=1}^n g \circ \mathcal{A}_{\mathfrak{M}^{(j)}, \mathfrak{N}^{(j)}} \right\|_{\mathcal{L}^2(\mathbb{P}_{X_d})} \right] \\
&\leq \left\| \mathbb{E}[\delta(1 + \|\mathcal{A}_{\mathfrak{M}, \mathfrak{N}}\|_2^\nu)] \right\|_{\mathcal{L}^2(\mathbb{P}_{X_d})} \\
&\quad + \left( \mathbb{E} \left[ \int_{[u,v]^d} \left( \mathbb{E}[(g \circ \mathcal{A}_{\mathfrak{M}, \mathfrak{N}})(x)] - \frac{1}{n} \sum_{j=1}^n (g \circ \mathcal{A}_{\mathfrak{M}^{(j)}, \mathfrak{N}^{(j)}})(x) \right)^2 d\mathbb{P}_{X_d}(x) \right] \right)^{1/2} \\
&\leq \delta(1 + \|\mathbb{E}[\|\mathcal{A}_{\mathfrak{M}, \mathfrak{N}}\|_2^\nu]\|_{\mathcal{L}^2(\mathbb{P}_{X_d})}) + \left( \int_{[u,v]^d} \mathbb{V} \left[ \frac{1}{n} \sum_{j=1}^n (g \circ \mathcal{A}_{\mathfrak{M}^{(j)}, \mathfrak{N}^{(j)}})(x) \right] d\mathbb{P}_{X_d}(x) \right)^{1/2} \\
&\leq 2\mathfrak{c}_\nu(\mathfrak{m})d^{\nu/2}\delta + Dn^{-1/2} \leq \frac{1}{2}\varepsilon^{1/2}.
\end{aligned}$$

Next, observe that Lemma A.3 establishes that

$$\mathbb{E}[G_2] \leq \mathbb{E} \left[ \sum_{j=1}^n \left( \|\mathfrak{M}^{(j)}\|_2 + \|\mathfrak{N}^{(j)}\|_2 \right) \right] \leq n\mathbb{E}[\|\mathfrak{M}\|_2 + \|\mathfrak{N}\|_2] \leq 3\mathfrak{c}_1(1)dn,$$

which proves that  $\mathbb{E}[G] = \varepsilon^{-1/2}\mathbb{E}[G_1] + (6\mathfrak{c}_1(1)dn)^{-1}\mathbb{E}[G_2] \leq 1$ . Thus, there exists  $\omega \in \Omega$  such that  $G(\omega) \leq 1$  (see [29, Proposition 3.3]), and with

$$M^{(j)} := \mathfrak{M}^{(j)}(\omega) \quad \text{and} \quad N^{(j)} := \mathfrak{N}^{(j)}(\omega), \quad j \in \{1, \dots, n\},$$

it holds that

$$(A.8) \quad \frac{1}{(v-u)^d} \left\| F_d(T, \cdot) - \frac{1}{n} \sum_{j=1}^n g \circ \mathcal{A}_{M^{(j)}, N^{(j)}} \right\|_{\mathcal{L}^2([u,v]^d)}^2 = G_1^2(\omega) \leq G^2(\omega)\varepsilon \leq \varepsilon$$

and that

$$\max_{j=1}^n \left( \|M^{(j)}\|_2 + \|N^{(j)}\|_2 \right) = G_2(\omega) \leq 6\mathfrak{c}_1(1)G(\omega)dn \leq 192D^2\mathfrak{c}_1(1)d\varepsilon^{-1}.$$

By Lemma A.4, our assumptions, and (A.6) there exist  $\mathbf{a} \in \mathbf{A}_d$  and  $\boldsymbol{\theta} \in \mathcal{P}_{\mathbf{a}}$  satisfying the following:

- (i)  $\text{clip}_D \circ \mathcal{F}(\boldsymbol{\theta}) = \mathcal{F}(\boldsymbol{\theta}) = \frac{1}{n} \sum_{j=1}^n \mathcal{F}(\boldsymbol{\eta}_{d,\delta}) \circ \mathcal{A}_{M^{(j)}, N^{(j)}} = \frac{1}{n} \sum_{j=1}^n g \circ \mathcal{A}_{M^{(j)}, N^{(j)}}$ ;
- (ii)  $P(\mathbf{a}) \leq n^2 P(\mathbf{b}_{d,\delta}) \leq 32^2 D^4 \zeta d^\gamma \varepsilon^{-2} \delta^{-\lambda} \leq C d^{\nu\lambda/2 + \gamma} \varepsilon^{-\lambda/2 - 2}$ ;
- (iii)  $\|\boldsymbol{\theta}\|_\infty \leq \sqrt{d} \|\boldsymbol{\eta}_{d,\delta}\|_\infty (192D^2\mathfrak{c}_1(1)d\varepsilon^{-1} + 1) \leq C d^{(\nu\kappa+3)/2 + \beta} \varepsilon^{-\kappa/2 - 1}$ ;
- (iv)  $L(\mathbf{a}) = L(\mathbf{b}_{d,\delta}) = L(\mathbf{b}_{d,cd^{-\nu/2}\varepsilon^{1/2}})$ ; and
- (v)  $\|\mathbf{a}\|_\infty = n \|\mathbf{b}_{d,\delta}\|_\infty \leq 32D^2\varepsilon^{-1} \|\mathbf{b}_{d,\delta}\|_\infty \leq C \varepsilon^{-1} \|\mathbf{b}_{d,cd^{-\nu/2}\varepsilon^{1/2}}\|_\infty$ ,

where  $C := \zeta \max \{32^2 D^4 (8\mathfrak{c}_\nu(\mathfrak{m}))^\lambda, (192D^2\mathfrak{c}_1(1) + 1) (8\mathfrak{c}_\nu(\mathfrak{m}))^\kappa\}$  and  $c := (8\mathfrak{c}_\nu(\mathfrak{m}))^{-1}$ . Together with (A.8) this proves the theorem.  $\blacksquare$

**Acknowledgments.** The authors are grateful to Shahar Mendelson and Stefan Steinerberger for their useful comments.

## REFERENCES

- [1] C. ALIPRANTIS AND K. BORDER, *Infinite Dimensional Analysis: A Hitchhiker's Guide*, 3rd ed., Springer, 2007.
- [2] Z. ALLEN-ZHU, Y. LI, AND Z. SONG, *A convergence theory for deep learning via over-parameterization*, in International Conference on Machine Learning, 2019, pp. 242–252.
- [3] W. AMES, *Numerical Methods for Partial Differential Equations*, Comput. Sci. Sci. Comput., Elsevier Science, 2014.
- [4] M. ANTHONY AND P. BARTLETT, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, 2009.
- [5] L. ARNOLD, *Stochastic Differential Equations*, Wiley-Interscience, 1974.
- [6] S. ARORA, R. GE, B. NEYSHABUR, AND Y. ZHANG, *Stronger generalization bounds for deep nets via a compression approach*, in International Conference on Machine Learning, 2018, pp. 254–263.
- [7] P. L. BARTLETT, O. BOUSQUET, AND S. MENDELSON, *Local Rademacher complexities*, Ann. Statist., 33 (2005), pp. 1497–1537.
- [8] P. L. BARTLETT, D. J. FOSTER, AND M. TELGARSKY, *Spectrally-normalized margin bounds for neural networks*, in Advances in Neural Information Processing Systems, 2017, pp. 6240–6249.
- [9] P. L. BARTLETT, N. HARVEY, C. LIAW, AND A. MEHRABIAN, *Nearly-tight VC-dimension and pseudo-dimension bounds for piecewise linear neural networks*, J. Mach. Learn. Res., 20 (2019), 63.
- [10] C. BECK, S. BECKER, P. GROHS, N. JAAFARI, AND A. JENTZEN, *Solving Stochastic Differential Equations and Kolmogorov Equations by Means of Deep Learning*, preprint, <https://arxiv.org/abs/1806.00421>, 2018.
- [11] C. BECK, W. E, AND A. JENTZEN, *Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations*, J. Nonlinear Sci., 29 (2019), pp. 1563–1619.
- [12] S. BECKER, P. CHERIDITO, AND A. JENTZEN, *Deep optimal stopping*, J. Mach. Learn. Res., 20 (2019), pp. 1–25.
- [13] J. BERNER, D. ELBRÄCHTER, AND P. GROHS, *How degenerate is the parametrization of neural networks with the ReLU activation function?*, in Advances in Neural Information Processing Systems 32, Curran Associates, 2019, pp. 7790–7801.
- [14] J. BERNER, D. ELBRÄCHTER, P. GROHS, AND A. JENTZEN, *Towards a regularity theory for ReLU networks—chain rule and global error estimates*, in the 13th International Conference on Sampling Theory and Applications (SampTA), IEEE, 2019, pp. 1–5.
- [15] H. BÖLCSKEI, P. GROHS, G. KUTYNIOK, AND P. PETERSEN, *Optimal approximation with sparsely connected deep neural networks*, SIAM J. Math. Data Sci., 1 (2019), pp. 8–45, <https://doi.org/10.1137/18M118709X>.
- [16] M. BURGER AND A. NEUBAUER, *Error bounds for approximation with neural networks*, J. Approx. Theory, 112 (2001), pp. 235–250.
- [17] A. CHOROMANSKA, M. HENAFF, M. MATHIEU, G. B. AROUS, AND Y. LECUN, *The loss surfaces of multilayer networks*, Proc. Mach. Learn. Res., 38 (2015), pp. 192–204.
- [18] F. CUCKER AND S. SMALE, *On the mathematical foundations of learning*, Bull. Amer. Math. Soc., 39 (2002), pp. 1–49.
- [19] F. CUCKER AND D. X. ZHOU, *Learning Theory: An Approximation Theory Viewpoint*, Cambridge Monogr. Appl. Comput. Math., Cambridge University Press, 2007.
- [20] S. S. DU, J. D. LEE, H. LI, L. WANG, AND X. ZHAI, *Gradient descent finds global minima of deep neural networks*, in International Conference on Machine Learning, 2019, pp. 1675–1685.
- [21] W. E, J. HAN, AND A. JENTZEN, *Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations*, Commun. Math. Stat., 5 (2017), pp. 349–380.
- [22] W. E AND B. YU, *The deep Ritz method: A deep learning-based numerical algorithm for solving variational*

- problems*, Commun. Math. Stat., 6 (2018), pp. 1–12.
- [23] D. ELBRÄCHTER, P. GROHS, A. JENTZEN, AND C. SCHWAB, *DNN Expression Rate Analysis of High-Dimensional PDEs: Application to Option Pricing*, preprint, <https://arxiv.org/abs/1809.07669>, 2018.
- [24] L. EVANS, *Partial Differential Equations*, 2nd ed., Grad. Stud. Math., American Mathematical Society, 2010.
- [25] M. FUJII, A. TAKAHASHI, AND M. TAKAHASHI, *Asymptotic expansion as prior knowledge in deep learning method for high dimensional BSDEs*, Asia-Pacific Financial Markets, 26 (2019), pp. 391–408.
- [26] K.-I. FUNAHASHI, *On the approximate realization of continuous mappings by neural networks*, Neural Networks, 2 (1989), pp. 183–192.
- [27] N. GOLOWICH, A. RAKHIN, AND O. SHAMIR, *Size-independent sample complexity of neural networks*, in Conference on Learning Theory, 2018, pp. 297–299.
- [28] C. GRAHAM AND D. TALAY, *Stochastic Simulation and Monte Carlo Methods: Mathematical Foundations of Stochastic Simulation*, Stoch. Model. Appl. Probab. 68, Springer, 2013.
- [29] 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*, preprint, <https://arxiv.org/abs/1809.02362>, 2018; to appear in Mem. Amer. Math. Soc.
- [30] P. GROHS, F. HORNUNG, A. JENTZEN, AND P. ZIMMERMANN, *Space-Time Error Estimates for Deep Neural Network Approximations for Differential Equations*, preprint, <https://arxiv.org/abs/1908.03833>, 2019.
- [31] L. GYÖRFI, M. KOHLER, A. KRZYZAK, AND H. WALK, *A Distribution-Free Theory of Nonparametric Regression*, Springer Science & Business Media, 2006.
- [32] M. HAIRER, M. HUTZENTHALER, AND A. JENTZEN, *Loss of regularity for Kolmogorov equations*, Ann. Probab., 43 (2015), pp. 468–527.
- [33] J. HAN, A. JENTZEN, AND W. E, *Solving high-dimensional partial differential equations using deep learning*, Proc. Natl. Acad. Sci. USA, 115 (2018), pp. 8505–8510.
- [34] P. HENRY-LABORDERE, *Deep primal-dual algorithm for BSDEs: Applications of machine learning to CVA and IM*, SSRN Electronic J., (2017), <https://doi.org/10.2139/ssrn.3071506>.
- [35] G. HINTON, L. DENG, D. YU, G. E. DAHL, A. R. MOHAMED, N. JAITLY, A. SENIOR, V. VANHOUCKE, P. NGUYEN, T. N. SAINATH, AND B. KINGSBURY, *Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups*, IEEE Signal Process. Mag., 29 (2012), pp. 82–97.
- [36] M. HUTZENTHALER, A. JENTZEN, T. KRUSE, AND T. A. NGUYEN, *A proof that rectified deep neural networks overcome the curse of dimensionality in the numerical approximation of semilinear heat equations*, SN Partial Differential Equations Appl., 1 (2020), pp. 1–34.
- [37] A. JENTZEN, D. SALIMOVA, AND T. WELTI, *A Proof That Deep Artificial Neural Networks Overcome the Curse of Dimensionality in the Numerical Approximation of Kolmogorov Partial Differential Equations with Constant Diffusion and Nonlinear Drift Coefficients*, preprint, <https://arxiv.org/abs/1809.07321>, 2018.
- [38] K. KAWAGUCHI, *Deep learning without poor local minima*, in Advances in Neural Information Processing Systems 29, Curran Associates, 2016, pp. 586–594.
- [39] D. P. KINGMA AND J. BA, *Adam: A Method for Stochastic Optimization*, preprint, <https://arxiv.org/abs/1412.6980>, 2014.
- [40] V. KOLTCHINSKII, *Introduction*, in Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems, Springer, 2011, pp. 1–16.
- [41] G. KUTYNIOK, P. PETERSEN, M. RASLAN, AND R. SCHNEIDER, *A Theoretical Analysis of Deep Neural Networks and Parametric PDEs*, preprint, <https://arxiv.org/abs/1904.00377>, 2019.
- [42] Y. LECUN, Y. BENGIO, AND G. HINTON, *Deep learning*, Nature, 521 (2015), pp. 436–444.
- [43] Y. LECUN, C. CORTES, AND C. J. C. BURGES, *The MNIST Database of Handwritten Digits*, 1998, <http://yann.lecun.com/exdb/mnist/> [online; accessed August 22, 2018].
- [44] Y. LI AND Y. LIANG, *Learning overparameterized neural networks via stochastic gradient descent on structured data*, in Advances in Neural Information Processing Systems, 2018, pp. 8157–8166.
- [45] Y. LI AND Y. YUAN, *Convergence analysis of two-layer neural networks with ReLU activation*, in Advances in Neural Information Processing Systems, 2017, pp. 597–607.

- [46] P. MASSART, *Concentration Inequalities and Model Selection*, Springer, 2007.
- [47] S. MEI, A. MONTANARI, AND P.-M. NGUYEN, *A mean field view of the landscape of two-layer neural networks*, Proc. Natl. Acad. Sci. USA, 115 (2018), pp. E7665–E7671.
- [48] M. MOHRI, A. ROSTAMIZADEH, A. TALWALKAR, AND F. BACH, *Foundations of Machine Learning*, Adapt. Comput. Mach. Learn., MIT Press, 2012.
- [49] B. NEYSHABUR, S. BHOJANAPALLI, D. MCALLESTER, AND N. SREBRO, *Exploring generalization in deep learning*, in Advances in Neural Information Processing Systems, 2017, pp. 5947–5956.
- [50] M. NIELSEN, *Neural Networks and Deep Learning*, 2015, <http://neuralnetworksanddeeplearning.com/chap1.html> [online; accessed March 05, 2018].
- [51] D. PEREKRESTENKO, P. GROHS, D. ELBRÄCHTER, AND H. BÖLCSKEI, *The Universal Approximation Power of Finite-Width Deep ReLU Networks*, preprint, <https://arxiv.org/abs/1806.01528>, 2018.
- [52] P. PETERSEN, M. RASLAN, AND F. VOIGTLAENDER, *Topological Properties of the Set of Functions Generated by Neural Networks of Fixed Size*, preprint, <https://arxiv.org/abs/1806.08459>, 2018.
- [53] P. PETERSEN AND F. VOIGTLAENDER, *Optimal approximation of piecewise smooth functions using deep ReLU neural networks*, Neural Networks, 108 (2018), pp. 296–330.
- [54] C. REISINGER AND Y. ZHANG, *Rectified Deep Neural Networks Overcome the Curse of Dimensionality for Nonsmooth Value Functions in Zero-Sum Games of Nonlinear Stiff Systems*, preprint, <https://arxiv.org/abs/1903.06652>, 2019.
- [55] C. SCHWAB AND J. ZECH, *Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ*, Anal. Appl., 17 (2019), pp. 19–55.
- [56] R. SEYDEL, *Tools for Computational Finance*, Universitext, Springer London, 2012.
- [57] U. SHAHAM, A. CLONINGER, AND R. R. COIFMAN, *Provable approximation properties for deep neural networks*, Appl. Comput. Harmon. Anal., 44 (2018), pp. 537–557.
- [58] O. SHAMIR AND T. ZHANG, *Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes*, in Proceedings of the 30th International Conference on Machine Learning, PMLR, 2013, pp. 71–79.
- [59] J. SIRIGNANO AND K. SPILIOPOULOS, *DGM: A deep learning algorithm for solving partial differential equations*, J. Comput. Phys., 375 (2018), pp. 1339–1364.
- [60] S. A. VAN DE GEER, *Applications of Empirical Process Theory*, Camb. Ser. Stat. Probab. Math. 6, Cambridge University Press, 2000.
- [61] D. YAROTSKY, *Error bounds for approximations with deep ReLU networks*, Neural Networks, 94 (2017), pp. 103–114.
- [62] C. ZHANG, S. BENGIO, M. HARDT, B. RECHT, AND O. VINYALS, *Understanding deep learning requires rethinking generalization*, in 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 2017.