Journal of Machine Learning Research 27 (2026) 1-22 Submitted 3/24; Published 3/26

Optimal Approximation and Generalization Errors for Deep Convolutional Neural Networks

Jinxin Wang JINXINWANGXJTU@GMAIL.COM
Shao-Bo Lin* SBLIN1983@GMAIL.COM
Center for Intelligent Decision-Making and Machine Learning
School of Management
Xi'an Jiaotong University
Xi'an, China

Editor: Quanquan Gu

Abstract

This paper focuses on approximation and learning performances of deep convolutional neural networks with zero-padding and max-pooling. We prove that, to approximate r-smooth function, the approximation rates of deep convolutional neural networks with depth L are of order $(L/\log L)^{-2r/d}$, which is optimal up to a logarithmic factor. Furthermore, we deduce almost optimal generalization errors for implementing empirical risk minimization over deep convolutional neural networks. Our theoretical results are verified by several numerical experiments to show the power of the convolutional structure, zero-padding and max-pooling.

Keywords: Deep learning, learning theory, deep convolutional neural networks, pooling

1. Introduction

Deep learning (LeCun et al., 2015) has made great breakthrough and profound impacts in numerous application regions including the computer science, life science and management science. One of the most important reasons for its success is the adoption of structured networks (Goodfellow et al., 2016) to autonomously encode the a-priori information and significantly reduce the number of tunable parameters in the training process. Deep convolutional neural networks (DCNNs), a widely used structured deep neural networks, have triggered enormous research activities in numerical applications (Gonzalez, 2018; Rawat and Wang, 2017; Yoo, 2015) and theoretical analysis (Zhou, 2020a; Fang et al., 2020; Mao et al., 2021).

In this paper, we focus on approximation and learning performances of DCNNs induced by the rectifier linear unit (ReLU), $\sigma(t) := \max\{t, 0\}$. For $\vec{v} \in \mathbb{R}^{d'}$ with $d' \in \mathbb{N}$, the one-dimensional and one-channel convolution is defined by

$$(\vec{w} \star \vec{v})_j = \sum_{k=j-s}^{j} w_{j-k} v_{k+s}, \qquad j = 1, \ldots, d' - s, \tag{1}$$

*. Corresponding author

where $\vec{w} = (w_j)_{j=-\infty}^{\infty}$ is a filter of length s, i.e. $w_j \neq 0$ only for $0 \leq j \leq s$. Then the classical DCNN is given by

$$\mathcal{N}^{\star}_{L,s}(x) := \vec{a}_L \cdot \sigma \circ \mathcal{C}^{\star}_{L,\vec{w}^L,\vec{b}^L} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^{\star}_{1,\vec{w}^1,\vec{b}^1}(x), \qquad x \in \mathbb{R}^d, \tag{2}$$

where $\vec{w}^\ell$ is the filter of length s, $\vec{b}^\ell \in \mathbb{R}^{d_\ell}$ is the bias vector with $d_\ell = d_{\ell-1} - s$ and $d_0 = d$,

$$\mathcal{C}^{\star}_{\ell,\vec{w}^\ell,\vec{b}^\ell}(\vec{u}) := \vec{w}^\ell \star \vec{u} + \vec{b}^\ell, \qquad \vec{u} \in \mathbb{R}^{d_{\ell-1}},$$

$\vec{a}_L \in \mathbb{R}^{d_L}$, and σ acts on vectors componentwise. Since DCNN defined by (2) possesses a contracting nature in the sense that the width of the network shrinks with respect to the depth, DCNN is not a universal approximant as the minimal width requirement for the universality of deep neural networks is $d + 1$ (Hanin, 2019).

Zero-padding is a feasible approach to avoid the aforementioned non-universality of DCNN. Define the convolution with zero-padding by

$$(\vec{w} * \vec{v})_j = \sum_{k=1}^{d'} w_{j-k} v_k, \qquad j = 1, \ldots, d' + s \tag{3}$$

and corresponding DCNN with zero-padding by

$$\mathcal{N}_{L,s}(x) = \vec{a}_L \cdot \sigma \circ \mathcal{C}_{L,\vec{w}^L,\vec{b}^L} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}_{1,\vec{w}^1,\vec{b}^1}(x), \qquad x \in \mathbb{R}^d, \tag{4}$$

where

$$\mathcal{C}_{\ell,\vec{w}^\ell,\vec{b}^\ell}(\vec{u}) := \vec{w}^\ell * \vec{u} + \vec{b}^\ell, \qquad \vec{u} \in \mathbb{R}^{d_{\ell-1}} \tag{5}$$

and $d_\ell = d_{\ell-1} + s$. As DCNN with zero-padding defined by (4) exhibits an expansive nature, we write it as eDCNN for the sake of brevity and denote by $\mathcal{H}_{L,s}$ the set of all eDCNNs. The power of zero-padding has been explored in (Zhou, 2020b), where the universal approximation of eDCNN without additional fully connected layers was established. Furthermore, it was shown in (Han et al., 2023) that the translation-equivalence of DCNN can also be enhanced by zero-padding. The problem is, however, that optimal approximation and generalization errors for eDCNN remain open, though some sub-optimal results have been presented in (Zhou, 2018, 2020a; Lin et al., 2022; Zhou and Huo, 2024).

The purpose of this paper is to derive optimal approximation and generalization errors for eDCNN. We prove that, to approximate the well known r-smooth functions, equipped with the well known max-pooling scheme, eDCNN succeeds in yielding an approximation rate of order $(L/\log L)^{-2r/d}$, which is essentially better than the existing rates $L^{-r/d}$ for both eDCNN and eDCNN with pooling established in (Zhou, 2020a,b). We also prove that the derived approximation rates for eDCNN with max-pooling cannot be essentially improved up to a logarithmic factor. Based on the derived (almost) optimal approximation rates, we deduce (almost) optimal generalization errors for implementing empirical risk minimization (ERM) over eDCNN with max-pooling, which shows that eDCNN with max-pooling is also one of the most powerful tools for the learning purpose. This together with the translation-equivalence of eDCNN discussed in (Han et al., 2023) shows the power of the convolutional structure in deep learning. Our numerical results provide several intuitive evidences on the

excellent approximation and learning performances of eDCNN with max-pooling, compared with numerous network structures such as the classical fully connected neural networks and deep convolutional neural networks without zero-padding.

The rest of the paper is organized as follows. In the next section, we conduct approximation error analysis for eDCNN with max-pooling and derive almost optimal approximation errors. In Section 3, we deduce almost optimal generalization error for eDCNN with max-pooling in the framework of learning theory. In Section 4, we numerically verify the excellent performances of eDCNN with max-pooling via several toy simulations. In the last section, we draw a simple conclusion. All the proofs of theoretical results are given in Appendix.

2. Approximation Rates Analysis

As the width of eDCNN increases linearly with respect to the depth, it is preferable to adopt some pooling schemes (Zhou, 2020a) to shrink the network size, among which max-pooling is the most popular. The max-pooling operator $\mathcal{S}_{d',u}: \mathbb{R}^{d'} \to \mathbb{R}^{[d'/u]}$ for a vector $\vec{v} \in \mathbb{R}^{d'}$ with pooling size u is defined by

$$\mathcal{S}_{d',u}(\vec{v}) = \big(\max_{j=1,\ldots,u} v_{(k-1)u+j}\big)_{k=1}^{[d'/u]}, \tag{6}$$

where $[a]$ denotes the integer part of the real number a. We force the width of eDCNN to be independent of the number of layers by using the max-pooling operator defined by (6). For $s \geq 2$, write $L_{1,\max} := \left\lceil \frac{(2d+10)d}{s-1} \right\rceil$, $d_{1,\max} := d + L_{1,\max} s$ and $d_{\max} := 2d + 10 + \left\lceil \frac{(2d+10)^2}{s-1} \right\rceil s$. Define the max-pooling scheme as

$$\mathcal{P}_{\max}(\vec{v}) := \begin{cases} \mathcal{S}_{d_{1,\max},d}(\vec{v}) & \text{if } |\vec{v}| = d_{1,\max} \text{ and } \ell \leq L_{1,\max}, \\ \mathcal{S}_{d_{\max},2d+10}(\vec{v}) & \text{if } |\vec{v}| = d_{\max}, \\ \vec{v} & \text{otherwise}, \end{cases} \tag{7}$$

which means that the output of the $L_{1,\max}$-th layer is max-pooled with pooling size d and the output of the layers with sizes $d_{\max}$ are pooled with pooling size $2d+10$. We then define eDCNN with max-pooling by

$$\mathcal{N}_{L,s}^{pool}(x) := \vec{a} \cdot \mathcal{P}_{\max} \circ \sigma \circ \mathcal{C}_{L,\vec{w}^L,\vec{b}^L} \circ \cdots \circ \mathcal{P}_{\max} \circ \sigma \circ \mathcal{C}_{1,\vec{w}^1,\vec{b}^1}(x). \tag{8}$$

Denote by $\Phi_{L,s}^{pool}$ the set of eDCNNs formed as (8). It is easy to find that the width of the network defined in (8) is always smaller than $d_{\max}$. It should be highlighted that the network structure in (8) is fixed once s and d are specified and there are totally $\mathcal{O}(L)$ free parameters in $\mathcal{N}_{L,s}^{pool}$. Our following result shows that even for $\mathcal{O}(L)$ tunable parameters, eDCNN with max-pooling can approximate r-smooth function with an order of $(L/\log L)^{-2r/d})$, which is essentially better than $\mathcal{O}(L^{-r/d})$, the best rates for shallow approximation with $\mathcal{O}(L)$ free parameters (Yarotsky, 2017; Guo et al., 2019) and existing approximation rates for eDCNN (Zhou, 2020a,b).

Let $\mathbb{I}^d := [0,1]^d$. For $c_0 > 0$ and $r = s + \mu$ with $s \in \mathbb{N}_0 := \{0\} \cup \mathbb{N}$ and $0 < \mu \leq 1$, $f: \mathbb{I}^d \to \mathbb{R}$ is said to be (r, c_0)-smooth if f is s-times differentiable and its s-th partial derivative satisfies the condition

$$\left| \frac{\partial^s f}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}(x) - \frac{\partial^s f}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}(x') \right| \leq c_0 \|x - x'\|_2^{\mu}, \quad \forall\, x, x' \in \mathcal{I}^d \tag{9}$$

for every $\alpha_j \in \mathbb{N}_0$, $j = 1, \ldots, d$ with $\alpha_1 + \cdots + \alpha_d = s$. Denote by $Lip^{(r,c_0)}$ the set of all (r, c_0)-smooth functions and $Lip_M^{(r,c_0)} := \{f \in Lip^{(r,c_0)} : \|f\|_{L^\infty(\mathbb{I}^d)} \leq M\}$. We then present our first main result as follows.

Theorem 1 *Let $r, c_0 > 0$, $s \geq 2$ and $M > 0$. There holds*

$$C_1(L \log L)^{-\frac{2r}{d}} \leq dist\left(Lip_M^{(r,c_0)}, \pi_M \Phi_{L,s}^{pool}, L^\infty(\mathbb{I}^d)\right) \leq C_2(L/\log L)^{-\frac{2r}{d}}, \tag{10}$$

where $\pi_M \mathbb{A} = \{\pi_M f : f \in \mathbb{A}\}$ with $\pi_M f(x) = sign(f(x)) \max\{|f(x)|, M\}$ denotes the truncation of the set $\mathbb{A}$, C_1, C_2 are constants depending only on r, c_0, s, d and the Hausdoff distance for $\mathbb{A}, \mathbb{B} \subseteq L^\infty(\mathbb{I}^d)$ is given by

$$dist(\mathbb{A}, \mathbb{B}, L^\infty(\mathbb{I}^d)) := \sup_{f \in \mathbb{A}} dist(f, \mathbb{B}, L^\infty(\mathbb{I}^d)) := \sup_{f \in \mathbb{A}} \inf_{g \in \mathbb{B}} \|f - g\|_{L^\infty(\mathbb{I}^d)}.$$

The derived approximation errors in (10) are essentially smaller than the linear n-width (Pinkus, 2012; Kurková and Sanguineti, 2002; Maiorov, 2006) and therefore show that eDCNN with max-pooling outperforms linear approaches. It can be found in Theorem 1 that up to a logarithmic factor, the derived approximation rates are optimal in the sense of width theory developed in (Pinkus, 2012; Kurková and Sanguineti, 2002). Recalling that there are totally $\mathcal{O}(L)$ parameters involved in eDCNN with max-pooling, approximation rates of order $(L/\log L)^{-2r/d}$ demonstrate the power of depth since optimal approximation rates for shallow nets with L parameters are only of order $L^{-r/d}$. Similar results for deep fully connected networks (DFCNs) have been presented in (Yarotsky and Zhevnerchuk, 2020; Lu et al., 2021), in which approximation rates of order $(L/\log L)^{-2r/d}$ were established for DFCNs with fixed width $2d + 10$. Theorem 1 shows that eDCNNs with max-pooling perform at least not worse than DFCNs. However, it has been verified in (Han et al., 2023) that eDCNN succeeds in encoding the translation-equivalence into the network structure which is beyond the capability of DFCNs. Theorem 1 together with results in (Han et al., 2023) thus rigorously shows the power of the convolution structure over fully connection. The detailed comparisons can be found in Table 1. It can be found in the table that our approach forces the width of eDCNN by adopting max-pooling and therefore performs better than existing approximation rates for eDCNN (Zhou, 2018, 2020a,b; Zhou and Huo, 2024).

3. Generalization Errors Analysis

In this section, we study the learning performance of eDCNN with max-pooling. Our analysis is carried out in the standard least square regression framework (Györfi et al., 2002; Cucker and Zhou, 2007), in which the samples $D := \{(x_i, y_i)\}_{i=1}^m$ are assumed to be drawn independently and identically according to an unknown but definite distribution $\rho := \rho_X \times \rho(y|x)$ with ρ_X the marginal distribution and $\rho(y|x)$ the conditional distribution conditioned on x. Our aim is to find an estimator in $\Phi_{s,L}^{pool}$ to approximate the well known regression function $f_\rho = \int_{\mathcal{Y}} y d\rho(y|x)$, which minimizes the generalization error $\mathcal{E}(f) := \int (f(x) - y)^2 d\rho$. Denoting by $L^2_{\rho_X}$ the space of ρ-square integrable functions endowed with norm $\|\cdot\|_\rho$, it is easy to check

$$\mathcal{E}(f) - \mathcal{E}(f_\rho) = \|f - f_\rho\|_\rho^2, \qquad f \in L^2_{\rho_X}. \tag{11}$$

Reference	Structure	Depth	Width	Rate
(Yarotsky, 2017)	fully connected	$\log N$	N	$(N/\log N)^{-r/d}$
(Han et al., 2022)	sparsely connected	$\mathcal{O}(d)$	$> N$	$N^{-r/d}$
(Lu et al., 2021)	fully connected	N	$\mathcal{O}(d)$	$(N/\log N)^{-2r/d}$
(Zhou, 2020b)	convolutional	N	$d+Ns$	$N^{-r/d}(r\geq 2)$
(Zhou, 2020a)	convolutional-pooling	N	$\max\{N, d+Ls\}$	$N^{-r/d}$
Ours	convolutional-pooling	N	$\mathcal{O}(d)$	$(N/\log N)^{-2r/d}$

Table 1: Comparisons between existing results in approximating functions in $Lip^{(r,c_0)}$ for deep nets with $\mathcal{O}(N)$ free parameters under the L^∞ metric. Here, s denotes the filter length of DCNN and width denotes the maximal width of layers.

Define

$$f_{D,L,s}^{pool} \in \arg\min_{f\in\Phi_{L,s}^{pool}} \frac{1}{m}\sum_{i=1}^{m}(f(x_i)-y_i)^2 \tag{12}$$

to be an arbitrary global minimum of the empirical risk minimization with least squares loss. We present our second main result in the following theorem.

Theorem 2 *Let $r, c_0 > 0$, $s \geq 2$, $M > 0$ and $0 < \delta < 1$. If $|y_i| \leq M$, $f_\rho \in Lip^{(r,c_0)}$ and $L \sim m^{\frac{d}{4r+2d}}$, then with confidence $1-\delta$ there holds*

$$\mathcal{E}(\pi_M f_{D,L,s}^{pool}) - \mathcal{E}(f_\rho) \leq C_3 m^{-\frac{2r}{2r+d}} (\log m)^{\max\{\frac{2r}{d},2\}}, \tag{13}$$

where C_3 is a constant depending only on r, c_0, s, d and M.

If $f_\rho \in Lip^{(r,c_0)}$, it can be found in (Györfi et al., 2002, Chap.3) that there is not any learning algorithm based on D such that the generalization error is essentially better than $\mathcal{O}(m^{-\frac{2r}{2r+d}})$. Theorem 2 thus shows that implementing ERM over eDCNN is one of the best learning algorithms in learning smooth regression functions. In particular, we can derive the following corollary.

Corollary 3 *Let $r, c_0 > 0$, $s \geq 2$ and $M > 0$. If $|y_i| \leq M$ and $L \sim m^{\frac{d}{4r+2d}}$, then*

$$C_4 m^{-\frac{2r}{2r+d}} \leq \sup_{f_\rho\in Lip^{(r,c_0)}} E\left[\|\pi_M f_{D,L,s}^{pool} - f_\rho\|_\rho^2\right] \leq C_5 m^{-\frac{2r}{2r+d}} (\log m)^{\max\{\frac{2r}{d},2\}}, \tag{14}$$

where C_4, C_5 are constants depending only on r, c_0, s, d and M.

Though similar optimal generalization errors have been derived for DFCNs (Schmidt-Hieber, 2020; Han et al., 2022; Chui et al., 2020), it remains open whether learning with eDCNN can achieve the same rates. Indeed, only universal consistency for learning with eDCNN has been verified in (Lin et al., 2022) and sub-optimal generalization errors of order $m^{\frac{-r}{2r+d}}$ have been deduced in (Mao et al., 2021; Zhou and Huo, 2024). Corollary 3 presents a new record for learning with eDCNN. It should be mentioned that optimal

generalization errors for eDCNNs with hybrid structure involving both convolutional layer and fully connected layers have been established in (Fang and Cheng, 2023). The main novelty of our results is that there are not any fully connected layers being added to the network, making the analysis take the depth L as the only parameter. We end this section by three important remarks.

Remark 4 *It can be found in Corollary 3 that there is an additional logarithmic term in the generalization error bounds. The main reasons are two folds: additional logarithmic term in the approximation error* (10) *and the uniform approach based on covering numbers in the sample error. The logarithmic term in approximation error is mainly caused by the approximation rates of deep fully connected nets (Lu et al., 2021), especially the product-gate property (Lu et al., 2021, Lemma 4.2). It might be removed by taking specific sparse structure rather than deep fully connected nets, just as (Petersen and Voigtlaender, 2018; Han et al., 2022) did to remove the logarithmic term in the approximation rates of deep fully connected nets in (Yarotsky, 2017). However, if sparse structure is involved, the current matrix factorization in our proof is no more the most suitable tool and we need more sophisticated tools to build the relation between eDCNN and deep sparsely connected networks. The logarithmic term in sample error is mainly caused by the concentration inequality (Lemma 13 below) based on covering number. Similar to all existing approximation rates for linear approaches (Györfi et al., 2002, Chap.11), shallow nets (Maiorov, 2006) and kernel methods (Steinwart and Christmann, 2008), removing the logarithmic term needs novel analysis approaches, like (Lin et al., 2017) introduced a novel integral operator approach for kernel ridge regression. The main difficulty is due to the nonlinear nature of eDCNN, making the similar integral operator approach infeasible.*

Remark 5 *This paper focuses on one-channel convolution and derives (almost) optimal generalization errors for the corresponding eDCNN. It should be highlighted that similar results can be extended to multi-channel convolution (with finite channels) directly. In fact, assuming that there are U channels, it is easy to derive generalization error similar to* (14) *with C_5 depending on U, since we can only use one channel to derive the same approximation error* (10) *and the sample error is about U times as the bound of one-channel's. However, we cannot use our approach to derive better generalization errors than Corollary 3 to embody the theoretical advantages of the multi-channel. As the generalization errors for smooth regression functions are already (almost) optimal for eDCNN with one-channel, additional a-priori information concerning regression functions may be needed to figure out the theoretical advantage of the multi-channel.*

Remark 6 *In this paper, we are only concerned with the generalization performance of the global minima of* (12)*, neglecting the detailed implementation algorithm. As illustrated in (Allen-Zhu et al., 2019), the convergence guarantee of gradient-based algorithms generally requires over-parameterization while our generalization error analysis needs under-parameterization to balance sample error with approximation error, according to the well-known bias-variance trade-off principle (Györfi et al., 2002). Practical convergence of SGD/Adam for under-parameterized deep nets relative to the theoretical ERM would bridge the gap between theory and practice.*

4. Experiments

In this section, we conduct toy simulations to validate the excellent approximation and learning performances of eDCNN with max-pooling. We consider the following deep networks: (1) deep fully connected network (DFCN); (2) deep convolutional neural network without any zero-padding or max-pooling schemes, denoted as cDCNN due to its contracting nature; (3) deep convolutional neural network with fully connected layers (cDCNN-fc); (4) deep convolutional neural network with zero-padding (eDCNN); (5) the proposed deep convolutional neural network with both zero-padding and max-pooling, which is denoted as eDCNN-maxpooling. For all network architectures, we utilize the ReLU activation function.

Input	Function formula	m	$m^{'}$
$x \in [-1/2, 1/2]^4$	$f_1(x) = \begin{cases} (1-2.2\|x\|_2)^6 \left(35(2.2\|x\|_2)^2 + 18(2.2\|x\|_2) + 3\right), & 2.2\|x\|_2 \le 1 \\ 0, & \text{otherwise,} \end{cases}$	5,000	1,000
$x \in [0,1]^{100}$	$f_2(x) = \exp\left(\frac{1}{100}\sum_{k=1}^{100} \sin^2\left(\frac{\pi x^{(k)}}{2}\right)\right)$	2,000	500

Table 2: Synthetic data generation.

We consider two synthetic functions. The first one is a smooth radial function and the other is a high-dimensional smooth function. The training inputs $\{x_i\}_{i=1}^m$ are drawn independently from the uniform distribution on $[-1/2, 1/2]^4$ for the first function and on $[0,1]^{100}$ for the second. The corresponding responses $\{y_i\}_{i=1}^m$ are generated according to the regression model $y_i = f_j(x_i) + \varepsilon_i$ for $i = 1, \ldots, m$ and $j = 1, 2$, where the noise ε_i is independent and follows the normal distribution $\mathcal{N}(0, \tau^2)$ with variance τ^2. Testing sets $\{(x_i', y_i')\}_{i=1}^{m'}$ are generated in the same manner but without noise. Further details are summarized in Table 2.

Optimizer	Initial learning rate	Scheduler	Train epoch	Batch size
Adam	$\{0.003, 0.001, 0.0006\}$	ExponentialLR($\gamma = 0.95$), step every 400 epochs	4,000	200

Table 3: Training configurations.

All experiments are implemented in PyTorch and conducted on a workstation equipped with an Intel i9-13900HX CPU, 64 GB of RAM, and an NVIDIA RTX-4070 GPU. The training configurations, including optimizer, learning rate schedule, number of epochs, and batch size, are summarized in Table 3. Moreover, all compared deep nets are initialized using PyTorch's default random initialization. For each convolutional layer in different DCNNs, the learnable weights (except bias b values) are sampled from $\mathcal{U}(-\sqrt{k}, \sqrt{k})$, where $k = \frac{1}{C_{\text{in}} \cdot \text{filter\_length}}$ with single input channel $C_{\text{in}} = 1$ and filter_length $= s$. After training, we evaluate the networks by computing the root mean square error (RMSE) on the testing set, and the reported results are averaged over 10 independent trials. The code for numerical experiments can be found in `https://github.com/DataScienceGroupOfManagement/eDCNN-maxpooling`.

4.1 Approximation performance verification

In this simulation, we compare the proposed eDCNN-maxpooling with DFCN, cDCNN, cDCNN-fc and eDCNN in approximating different simulation functions.

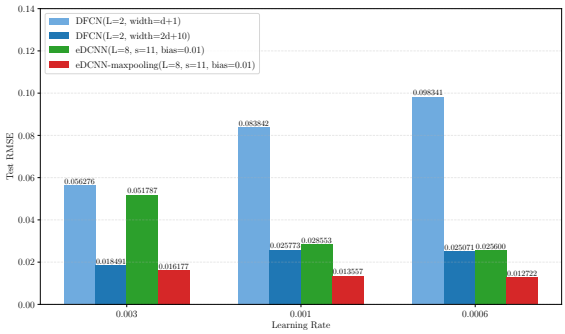
	Depth L	Filter length s	Initial bias b on f_1	Initial bias b on f_2
cDCNN	$\{4, 6, 8, \cdots, 14\}$	$\{3, 5, 7, 9\}$	$\{0.03, 0.01, 0.006\}$	$\{0.06, 0.03, 0.01\}$
cDCNN-fc	$\{2, 3, \cdots, 10\}$	$\{3, 5, 7, 9\}$	$\{0.03, 0.01, 0.006\}$	$\{0.06, 0.03, 0.01\}$
eDCNN	$\{4, 6, 8, \cdots, 16\}$	$\{5, 7, 9, \cdots, 15\}$	$\{0.03, 0.01, 0.006\}$	$\{0.06, 0.03, 0.01\}$
eDCNN-maxpooling	$\{4, 6, 8, \cdots, 16\}$	$\{5, 7, 9, \cdots, 15\}$	$\{0.03, 0.01, 0.006\}$	$\{0.06, 0.03, 0.01\}$

Table 4: Candidate hyper-parameters for DCNNs.

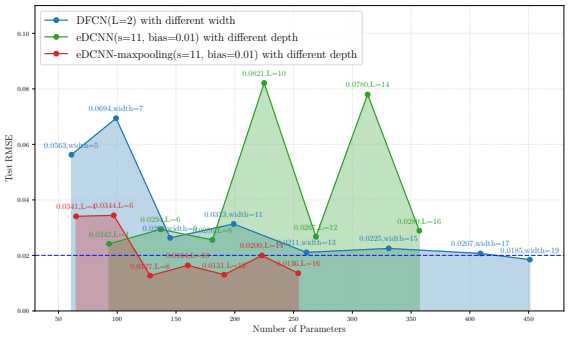
The first part investigates the relationship between test RMSE and different network architectures. For DCNN variants, the candidate hyper-parameters about the number of convolution layers (or depth L), the filter length s, and the initial bias value b for f_1 and f_2 are summarized in Table 4, and the best configurations are selected via grid search. For eDCNN and eDCNN-maxpooling, the padding size in each convolution is $s - 1$. And for eDCNN-maxpooling, a max-pooling layer with a pooling size $u = 2$ is employed after every $\frac{L}{2}$ convolutional layers. For cDCNN-fc network, one fully connected layer is added after the convolutional layers. When it comes to DFCN, we adopt the fixed width with $d + 1$ or $2d + 10$ according to the theory in (Hanin, 2019) and (Lu et al., 2021).

The second part investigates the relationship between test RMSE and the number of trainable parameters. We present the results for DFCN with varying network widths, as well as for eDCNN and eDCNN-maxpooling with different depths. To be detailed, for the 4-dimensional function f_1, the widths of DFCN vary within the set $\{5, 7, 9, \cdots, 19\}$. The depths of eDCNN and eDCNN-maxpooling are set within the range $\{4, 6, 8, \cdots, 16\}$. When it comes to the 100-dimensional function f_2, the widths of DFCN vary within set $\{2, 4, \cdots, 10, 20, 40, 60\}$. The depths of eDCNN and eDCNN-maxpooling are set within the range $\{2, 4, 6, 8\}$. Since the objective of this subsection is to evaluate the approximation capabilities of diverse deep nets, there is no noise in the training data. The results on two functions are presented in Figure 1.

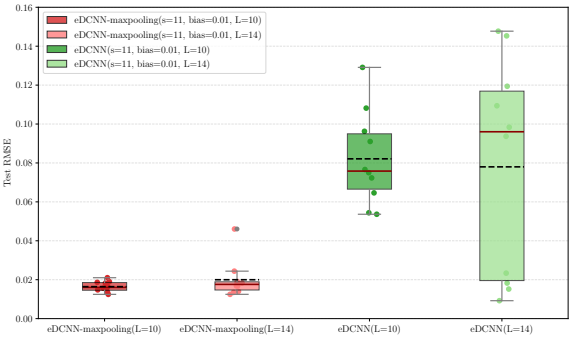
From the experimental results, we can draw the following conclusions: 1) eDCNN-maxpooling can achieve comparable approximation results, compared to DFCN and eDCNN. For instance, when applied to function f_1 (see results in Figure 1(a)), these networks yield a predictive accuracy with an RMSE of approximately 0.025. In a more detailed comparison, eDCNN-maxpooling exhibits not only superior performance relative to DFCN and eDCNN, but also adequate stability across different learning rates. For function f_2 (see results in Figure 1(d)), all three network architectures achieve exceptionally good approximation performance, with RMSE of approximately 0.01. Additionally, cDCNN exhibits relatively poor performance on function f_2, while incorporating a fully connected layer significantly enhances its approximation capabilities. 2) By deepening the network, eDCNN with max-pooling can achieve an approximation performance at least not worse than that of DFCN, despite employing fewer parameters, as illustrated in Figure 1(b) and Figure 1(e). For the function f_1, by employing an appropriate network depth, both eDCNN and eDCNN-maxpooling can outperform DFCN in terms of approximation performance while using fewer



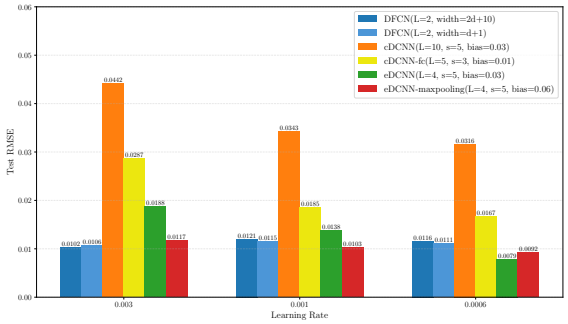
(a) Relation between test RMSE and learning rates on f_1.



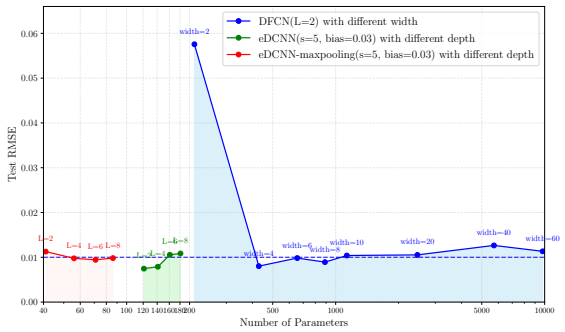
(b) Relation between test RMSE and the number of trainable parameters on f_1.



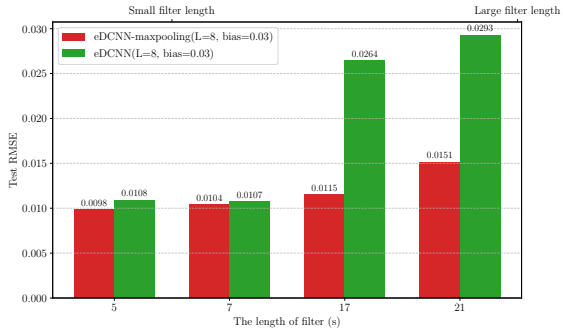
(c) Variance of test RMSE for eDCNN and eDCNN-maxpooling with $L = 10, 14$ on f_1.



(d) Relation between test RMSE and learning rates on f_2.



(e) Relation between test RMSE and the number of trainable parameters on f_2.



(f) Relationship between test RMSE and filter lengths of convolution on f_2.

Figure 1: Comparison of the test RMSE among DFCN, cDCNN, cDCNN-fc, eDCNN and eDCNN-maxpooling architectures in approximating the two simulation functions.

parameters. Furthermore, eDCNN-maxpooling exhibits more stable results across different network depths. For the high-dimensional function f_2, under the ideal parameter settings of $s = 5$ and $b = 0.03$, both eDCNN and eDCNN-maxpooling yield satisfactory approximation performance. These results highlight the structural advantage of eDCNN architectures. 3) The max-pooling mechanism effectively controls the capacity of the eDCNN network architecture, which is pivotal for eDCNN-maxpooling to achieve superior approximation performance. The advantages of eDCNN-maxpooling compared to eDCNN are presented in Figure 1(b) and Figure 1(c). From the theoretical consideration, the reason for the stable phenomenon of eDCNN-maxpooling is due to the contraction effect of max-pooling, making the hypothesis space and consequently sample error much smaller, which is consistent with the observations in Figure 1(c). Form the implementation viewpoint, the reasons are more sophisticate. It might be deduced by either randomness of the initialization points or the selection of learning rates (step size). To make the comparison fair, we use the same learning rate of 0.0006 as well as the same initialization method for both eDCNN and eDCNN-maxpooling. The multiple random experiments in Figure 1(c) clearly demonstrate the stable phenomenon that the variance of the test RMSE of eDCNN-maxpooling is significantly lower than that of eDCNN. For function f_2, Figure 1(f) depicts the relationship between approximation results and the filter lengths for eDCNN and eDCNN-maxpooling, given $L = 8$. The incorporation of max-pooling is found to enhance the approximation performance of the network architecture, with the benefit being more pronounced when larger filter lengths are employed.

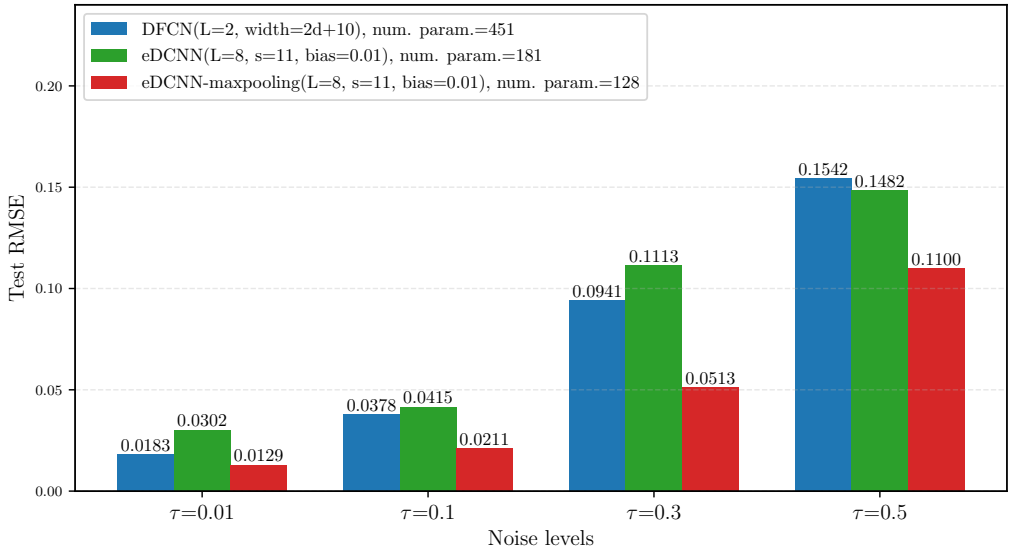
In summary, these findings validate the advantages of the eDCNN-maxpooling architecture in approximating smooth functions, and simultaneously reflect the necessity and effectiveness of introducing the max-pooling mechanism into the eDCNN architecture, which are consistent with the results presented in Section 2.

4.2 Learning performance verification

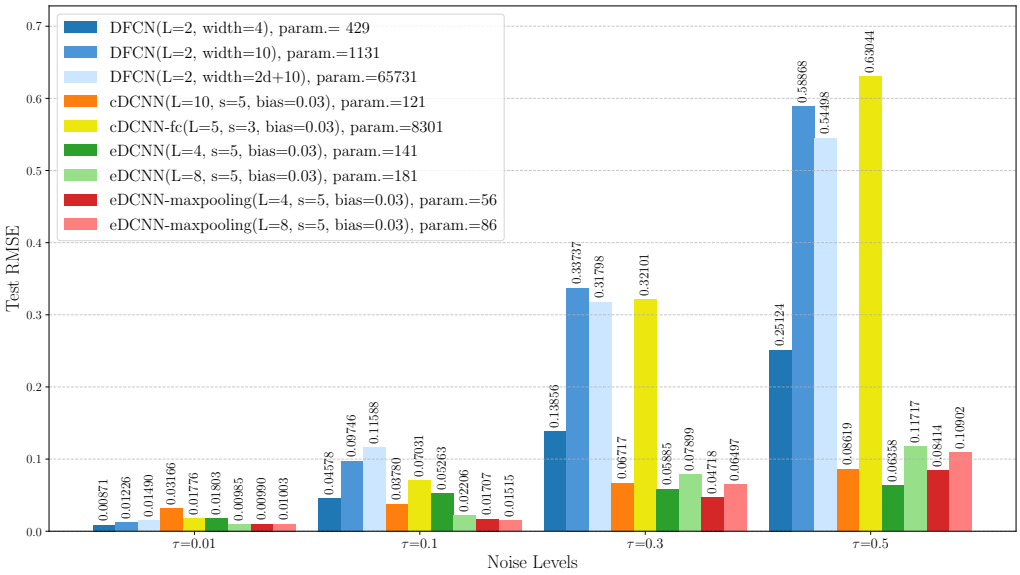
We evaluate the learning performance of the proposed eDCNN with max-pooling. Using two simulation functions with varying levels of Gaussian noise, we compare its performance against a range of other deep network architectures.

Specifically, the standard deviation τ of Gaussian noise varies from the set $\{0.01, 0.1, 0.3, 0.5\}$. For f_1, we mainly compare three network architectures: DFCN, eDCNN, and eDCNN-maxpooling. For cDCNN and cDCNN-fc, due to their contracting nature, they are not suitable for low-dimensional functions. While for high-dimensional function f_2, a detailed comparison of the generalization performance of these five network architectures on noisy data is carried out. The specific hyper-parameter settings and the number of trainable parameters for each network architecture are also presented in Figure 2.

Based on these simulation results, the following conclusions can be drawn: 1) The learning performance of eDCNN-maxpooling is at least not worse than the referenced networks including DFCN, cDCNN, cDCNN-fc and eDCNN. As the noise level increases, the test RMSE of different network architectures exhibits a similar overall upward trend. This phenomenon is observed for two simulation functions. Specifically, under low-noise conditions (e.g., noise deviation $\tau = 0.01$ or 0.1), various network architectures can achieve superior generalization performance (as reflected by low RMSE values). However, under high-noise



(a) Relationship between test RMSE and data with different noise levels on f_1.



(b) Relationship between test RMSE and data with different noise levels on f_2.

Figure 2: Comparison of learning performance across different network architectures for the two simulation functions, considering varying levels of noise.

conditions (e.g., $\tau = 0.3$ or 0.5), complex models with larger parameter counts (such as wider DFCN and cDCNN-fc) exhibit significant performance degradation. This deterioration stems from their increased susceptibility to noise overfitting. Conversely, network architectures with fewer parameters, such as cDCNN, eDCNN and eDCNN-maxpooling, maintain superior robustness, preserving better generalization capabilities; 2) The integration of max-pooling mechanism significantly enhances the generalization capability of deep convolutional architectures. Figure 2(a) illustrates the superior generalization advantage of eDCNN-maxpooling over DFCN and eDCNN on function f_1. Similarly, the comparative analysis of eDCNN variants in Figure 2(b) reveals that the max-pooling operation maintains its effectiveness regardless of different noise magnitudes. This is attributed to the fact that the max-pooling operation shrinks the network size and effectively controls the capacity of the network architecture. Consequently, it mitigates the issue of overfitting to noise caused by an overly large model capacity. Furthermore, comparative studies across network configurations (e.g., DFCN variants with differing widths, cDCNN vs. cDCNN-fc) consistently demonstrate that an appropriate network capacity is pivotal for achieving excellent generalization performance.

To sum up, these findings clearly verify the effectiveness of eDCNN with max-pooling in learning smooth regression functions, which are consistent with the results of Theorem 2 and Corollary 3.

5. Conclusion

In this paper, we investigate the approximation and learning performances of deep convolutional neural networks with zero-padding (eDCNN). By leveraging a well-established max-pooling scheme, we rigorously derive (almost) optimal approximation rates and (almost) optimal generalization errors for the proposed eDCNN architecture. Our theoretical analysis, combined with numerical experiments, demonstrates the superiority of eDCNN with max-pooling. These findings provide a springboard in understanding and designing high-performance deep learning architectures in practical scenarios.

While the proposed eDCNN achieves excellent approximation and learning performances, it is still valuable to remove the logarithmic terms in the derived rates by developing more advanced mathematical tools. Additionally, it is interesting to develop eDCNN's theoretical frameworks from single-channel to multi-channel configurations, thereby bridging the gap between theoretical results and practical implementations. Besides the approximation and generalization performances of eDCNN with max-pooling, developing provable optimization algorithms to solve (12) and study its convergence and landscape are also interesting. We will keep our study in these research topics and report our progress in future.

Acknowledgment

The work is supported by the National Natural Science Foundation of China (Grant Number 62276209). The authors would like to thank AE and the anonymous reviewers for their encouraging comments and valuable suggestions.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

Charles K Chui, Shao-Bo Lin, Bo Zhang, and Ding-Xuan Zhou. Realization of spatial sparseness by deep relu nets with massive data. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Felipe Cucker and Ding Xuan Zhou. *Learning Theory: an Approximation Theory Viewpoint*, volume 24. Cambridge University Press, 2007.

Zhiying Fang and Guang Cheng. Optimal learning rates of deep convolutional neural networks: Additive ridge functions. *Transactions on Machine Learning Research*, 2023.

Zhiying Fang, Han Feng, Shuo Huang, and Ding-Xuan Zhou. Theory of deep convolutional neural networks ii: Spherical analysis. *Neural Networks*, 131:154–162, 2020.

Rafael C Gonzalez. Deep convolutional neural networks [lecture notes]. *IEEE Signal Processing Magazine*, 35(6):79–87, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

Zheng-Chu Guo, Lei Shi, and Shao-Bo Lin. Realizing data features by deep nets. *IEEE Transactions on Neural Networks and Learning Systems*, 31(10):4036–4048, 2019.

László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-free Theory of Nonparametric Regression*, volume 1. Springer, 2002.

Zhi Han, Siquan Yu, Shao-Bo Lin, and Ding-Xuan Zhou. Depth selection for deep relu nets in feature extraction and generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):1853–1868, 2022.

Zhi Han, Baichen Liu, Shao-Bo Lin, and Ding-Xuan Zhou. Deep convolutional neural networks with zero-padding: Feature extraction and learning. *arXiv preprint arXiv:2307.16203*, 2023.

Boris Hanin. Universal function approximation by deep neural nets with bounded width and relu activations. *Mathematics*, 7(10):992, 2019.

Vera Kurková and Marcello Sanguineti. Comparison of worst case errors in linear and neural network approximation. *IEEE Transactions on Information Theory*, 48(1):264–275, 2002.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Shao-Bo Lin, Xin Guo, and Ding-Xuan Zhou. Distributed learning with regularized least squares. *Journal of Machine Learning Research*, 18(1):3202–3232, 2017.

Shao-Bo Lin, Kaidong Wang, Yao Wang, and Ding-Xuan Zhou. Universal consistency of deep convolutional neural networks. *IEEE Transactions on Information Theory*, 68(7): 4610–4617, 2022.

Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, 2021.

Vitaly Maiorov. Approximation by neural networks and learning theory. *Journal of Complexity*, 22(1):102–117, 2006.

Tong Mao and Ding-Xuan Zhou. Approximation of functions from korobov spaces by deep convolutional neural networks. *Advances in Computational Mathematics*, 48(6):1–26, 2022.

Tong Mao, Zhongjie Shi, and Ding-Xuan Zhou. Theory of deep convolutional neural networks iii: Approximating radial functions. *Neural Networks*, 144:778–790, 2021.

Hrushikesh Narhar Mhaskar. Approximation properties of a multilayered feedforward artificial neural network. *Advances in Computational Mathematics*, 1(1):61–80, 1993.

Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural Networks*, 108:296–330, 2018.

Allan Pinkus. *N-widths in Approximation Theory*, volume 7. Springer Science & Business Media, 2012.

Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, 2017.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48(4):1875–1897, 2020.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science & Business Media, 2008.

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

Dmitry Yarotsky and Anton Zhevnerchuk. The phase diagram of approximation rates for deep neural networks. *Advances in Neural Information Processing Systems*, 33:13005–13015, 2020.

Hyeon-Joong Yoo. Deep convolution neural networks in computer vision: a review. *IEIE Transactions on Smart Processing & Computing*, 4(1):35–43, 2015.

Ding-Xuan Zhou. Deep distributed convolutional neural networks: Universality. *Analysis and Applications*, 16(06):895–919, 2018.

Ding-Xuan Zhou. Theory of deep convolutional neural networks: Downsampling. *Neural Networks*, 124:319–327, 2020a.

Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794, 2020b.

Ding-Xuan Zhou and Kurt Jetter. Approximation with polynomial kernels and svm classifiers. *Advances in Computational Mathematics*, 25(1):323–344, 2006.

Tian-Yi Zhou and Xiaoming Huo. Learning ability of interpolating deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 68:101582, 2024.

Appendix: Proofs of Theoretical Results

In the appendix, we present the proofs of our main results. The technical novelties in our proofs are two folds, compared with the existing literature (Zhou, 2020a,b; Mao and Zhou, 2022; Zhou and Huo, 2024) in eDCNN approximation, in which the approximation rates are sub-optimal as shown in Table 1. On one hand, the max-pooling rather than a localization-based pooling in (Zhou, 2020a) is imposed in this paper so that we can get a unified convolutional representation for deep fully connected networks and therefore the relation (28) below holds. On the other hand, based on the convolutional representation and the recently developed approximation of smooth functions by deep but narrow fully connected networks (Yarotsky and Zhevnerchuk, 2020; Lu et al., 2021) (see Lemma 7 below), we derive almost optimal approximation rates for eDCNN with max-pooling. It should be highlighted that the adoption of results in (Yarotsky and Zhevnerchuk, 2020; Lu et al., 2021) is one of the most important factor to our breakthrough of approximation rates of eDCNN, since existing approximation results (Zhou, 2020a,b; Mao and Zhou, 2022; Zhou and Huo, 2024) utilize eDCNN to approximate wide but shallow nets at first and then obtain provable approximation errors by the corresponding approximation results in (Barron, 1993; Mhaskar, 1993; Yarotsky, 2017), which only achieves an approximation rate of order $\mathcal{O}(L^{-r/d})$ for $\mathcal{O}(L)$ free parameters.

To prove Theorem 1, we need several preliminary lemmas. The first one provided in (Yarotsky and Zhevnerchuk, 2020, Theorem 4.1) shows the approximation rates of DFCN with fixed width $2d+10$.

Lemma 7 *Let $r, c_0 > 0$, there exists a constant $\tilde{C}_1$ depending only on r and d such that*

$$dist\Big(Lip^{(r,c_0)}, \Psi_{L,2d+10,\ldots,2d+10}, L^\infty(\mathbb{I}^d)\Big) \leq \tilde{C}_1 L^{-2r/d} \log^{2r/d} L,$$

where $\Psi_{L,2d+10,\ldots,2d+10}$ is the set of DFCNs with depth L and width $2d+10$ in each layer.

Let ν be a probability measure on $\mathbb{I}^d$. For a function $f : \mathbb{I}^d \to \mathbb{R}$, set $\|f\|_{L^p(\nu)} := \left\{\int_{\mathbb{I}^d} |f(x)|^p d\nu\right\}^{1/p}$. Denote by $L^p(\nu)$ the set of all functions satisfying $\|f\|_{L^p(\nu)} < \infty$. For $\mathcal{V} \subset L^p(\nu)$, denote by $\mathcal{N}(\epsilon, \mathcal{V}, \|\cdot\|_{L^p(\nu)})$ the covering number (Györfi et al., 2002, Def. 9.3) of $\mathcal{V}$ in $L^p(\nu)$, which is the number of elements in a least ε-net of $\mathcal{V}$ with respect to $\|\cdot\|_{L^p(\nu)}$. In particular, denote by $\mathcal{N}_p(\epsilon, \mathcal{V}, x_1^m) := \mathcal{N}(\epsilon, \mathcal{V}, \|\cdot\|_{L^p(\nu_m)})$ with ν_m the empirical measure

with respect to $x_1^m = (x_1, \ldots, x_m) \in (\mathbb{I}^d)^m$. The second lemma that can be derived by the same method as (Lin et al., 2022, Lemma 4) builds the covering number estimate of $\Phi_{L,s}^{pool}$.

Lemma 8 *Let ν be a probability measure on $\mathbb{I}^d$. For any $0 < \varepsilon \leq M$, there holds*

$$\log_2 \mathcal{N}_1(\epsilon, \pi_M \Phi_{L,s}^{pool}, L^1(\nu)) \leq c^* L^2 \log L \log \frac{M}{\epsilon},$$

where c^* is a constant depending only on s and d.

The next lemma derived in (Guo et al., 2019) presents a relation between the covering number and approximation.

Lemma 9 *Let $n \in \mathbb{N}$ and $V \subseteq L_1(\mathbb{I}^d)$. For arbitrary $\varepsilon > 0$, if*

$$\mathcal{N}(\varepsilon, V, L_1(\mathbb{I}^d)) \leq \tilde{C}_1 \left(\frac{\tilde{C}_2 n^\beta}{\varepsilon} \right)^n \tag{15}$$

with $\beta, \tilde{C}_1, \tilde{C}_2 \geq 0$, then

$$dist(Lip_M^{(r,c_0)}, V, L_1(\mathbb{I}^d)) \geq C'(n \log_2(n+1))^{-r/d}, \tag{16}$$

where C' is a constant independent of n or ε.

The fourth lemma is the convolution factorization lemma provided in (Zhou, 2020b, Theorem 3).

Lemma 10 *Let $S \geq 0, 2 \leq s \leq d$ and $\vec{u} = (u_k)_{-\infty}^{\infty}$ be supported on $\{0, \ldots, S\}$. Then there exists $L < \frac{S}{s-1} + 1$ filter vectors $\{\vec{w}^\ell\}_{\ell=1}^L$ supported on $\{0, \ldots, s\}$ such that $\vec{u} = \vec{w}^L * \cdots * \vec{w}^1$.*

Define

$$T_{\tilde{d},d'}^{\vec{w}} := \begin{bmatrix} w_0 & 0 & 0 & \cdots & 0 \\ w_1 & w_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ w_{d'-1} & w_{d'-2} & \cdots & \cdots & w_0 \\ w_{d'} & w_{d'-1} & \cdots & 0 \cdots & w_1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ w_{\tilde{d}-d'} & \cdots & \cdots & \cdots & w_{\tilde{d}-2d'+1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ w_{\tilde{d}-2} & w_{\tilde{d}-3} & \cdots & w_{\tilde{d}-d'} & w_{\tilde{d}-d'-1} \\ w_{\tilde{d}-1} & w_{\tilde{d}-2} & \cdots & w_{\tilde{d}-d'+1} & w_{\tilde{d}-d'} \end{bmatrix}. \tag{17}$$

Observe that if W is supported in $\{0, \ldots, S\}$ for some $S \in \mathbb{N}$, the entry $(T_{D,d'}^{\vec{w}})_{i,k} = w_{i-k}$ vanishes when $i - k > S$. The fifth lemma proved in (Zhou, 2018) establishes the relation between the convolution and matrix multiplication.

Lemma 11 *Let $2 \le s \le d'$, $d'_\ell = d' + \ell s$ and $d'_0 = d'$. If $\{\vec{w}^\ell\}_{\ell=1}^L$ is supported on $\{0, \dots, s\}$, then*

$$T^{\vec{w}^\ell * \cdots \vec{w}^2 * \vec{w}^1}_{d'_\ell, d'} = T^{\vec{w}^\ell}_{d'_\ell, d'_{\ell-1}} \cdots T^{\vec{w}^2}_{d'_2, d'_1} T^{\vec{w}^1}_{d'_1, d'} \tag{18}$$

holds for any $\ell \in \{1, 2, \dots, L\}$.

For a sequence $\vec{w}$ supported on $\{0, 1, \dots, s\}$, write $\|\vec{w}\|_1 = \sum_{k=-\infty}^{\infty} |w_k|$ and $\|\vec{w}\|_\infty = \max_{-\infty \le k \le \infty} |w_k|$. Define $B^0 := \max_{x \in \mathbb{I}^d} \max_{k=1,\dots,d} |x^{(k)}|$ and

$$B^\ell := \|\vec{w}^\ell\|_1 B^{\ell-1} \cdots B^1 B^0, \qquad \ell \ge 1.$$

Then for any $j = 1, \dots, d_\ell$, direct computation yields

$$\max_{x \in \mathbb{I}^d} \left| \left(T^{\vec{w}^\ell}_{d_\ell, d_{\ell-1}} \dots T^{\vec{w}^1}_{d_1, d_0} x \right)_j \right| \le B^\ell \tag{19}$$

and for $1 \le k \le \ell - 1$

$$\left| \left(T^{\vec{w}^\ell}_{d_\ell, d_{\ell-1}} \dots T^{\vec{w}^{k+1}}_{d_{k+1}, d_k} B^k \mathbf{1}_{d_k} \right)_j \right| \le B^\ell, \tag{20}$$

where $\vec{a}_j$ denotes the j-th element of the vector $\vec{a}$ and $\mathbf{1}_{d_k} = (1, \dots, 1)^T \in \mathbb{R}^{d_k}$. For any $d'_0 = d' \in \mathbb{N}$ and $d'_\ell = d' + \ell s$, define the restricted convolution operator by

$$\mathcal{C}^R_{\ell, \vec{w}^\ell, b^\ell}(x) := \vec{w}^\ell * x + b^\ell \mathbf{1}_{d'_\ell} \tag{21}$$

for $\vec{w}^\ell$ supported on $\{0, 1, \dots, s\}$ and $b^\ell \in \mathbb{R}$. The following lemma derived in (Han et al., 2023) presents the relation between deep convolution neural network and matrix multiplication.

Lemma 12 *Let $\ell \in \mathbb{N}$, $2 \le s \le d$ and $\mathcal{C}^R_{\ell, \vec{w}^\ell, b^\ell}$ be defined by (21) with $\vec{w}^\ell$ supported on $\{0, 1, \dots, s\}$ and $b^\ell = 2^{\ell-1} B^\ell$, then*

$$\begin{aligned} \sigma \circ \mathcal{C}^R_{\ell, \vec{w}^\ell, b^\ell} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^R_{1, \vec{w}^1, b^1}(x) &= T^{\vec{w}^\ell}_{d_\ell, d_{\ell-1}} \cdots T^{\vec{w}^2}_{d_2, d_1} T^{\vec{w}^1}_{d_1, d} x + b^\ell \mathbf{1}_{d_\ell} \\ &+ \sum_{k=1}^{\ell-1} T^{\vec{w}^\ell}_{d_\ell, d_{\ell-1}} \cdots T^{\vec{w}^{k+1}}_{d_{k+1}, d_k} b^k \mathbf{1}_{d_k}. \end{aligned} \tag{22}$$

By the help of the above lemmas, we are in a position to prove Theorem 1 as follows.
Proof [Proof of Theorem 1] We divide the proof into four steps: matrix factorization, role of max-pooling, convolutional representation and approximation rate derivation.

Step 1. Matrix factorization: Given a $\tilde{d} \times d'$ matrix W, as shown in Figure 3, denote

$$\vec{u}^T = (W_{1,d'}, W_{1,d'-1}, \dots, W_{1,1}, W_{2,d'}, \dots, W_{2,1}, \dots, W_{\tilde{d},1}) =: (W_0, \dots, W_{\tilde{d}d'-1})$$

by stacking the rows of W. It follows from Lemma 10 with $S = \tilde{d}d' - 1$ and $s \ge 2$ that there exist $\hat{L} < \frac{\tilde{d}d'-1}{s-1} + 1$ filter vectors $\{\vec{w}^\ell\}_{\ell=1}^{\hat{L}}$ satisfying $\vec{u} = \vec{w}^{\hat{L}} * \cdots * \vec{w}^1$. Setting

$$\begin{pmatrix} W_{1,1} & W_{1,2} & \cdots & W_{1,d'} \\ W_{2,1} & W_{2,2} & \cdots & W_{2,d'} \\ \vdots & \vdots & \vdots & \vdots \\ W_{\tilde{d},1} & W_{\tilde{d},2} & \cdots & W_{\tilde{d},d'} \end{pmatrix} \Rightarrow \begin{pmatrix} W_{1,d'} \\ W_{1,d'-1} \\ \vdots \\ W_{1,1} \\ W_{2,d'} \\ \vdots \\ W_{2,1} \\ \vdots \\ W_{\tilde{d},d'} \\ \vdots \\ W_{\tilde{d},1} \end{pmatrix} \Rightarrow \begin{pmatrix} W_0 \\ W_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ W_{\tilde{d}d'-1} \end{pmatrix} = \vec{u}$$

Figure 3: Stacking of the rows of W.

$T^{\vec{u}} := (W_{k-j})_{k=1,\ldots,d'_{\hat{L}},j=1,\ldots,d'}$ as the $d'_{\hat{L}} \times d'$, then $T^{\vec{u}}$ is formed as (17), that is,

$$T^{\vec{u}} := \begin{bmatrix} W_0 & 0 & 0 & \cdots & 0 \\ W_1 & W_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ W_{d'-1} & W_{d'-2} & \cdots & \cdots & W_0 \\ W_{d'} & W_{d'-1} & \cdots & 0\cdots & W_1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ W_{d'_{\hat{L}}-d'} & \cdots & \cdots & \cdots & W_{d'_{\hat{L}}-2d'+1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ W_{\tilde{d}-2} & W_{d'_{\hat{L}}-3} & \cdots & W_{d'_{\hat{L}}-d'} & W_{d'_{\hat{L}}-d'-1} \\ W_{d'_{\hat{L}}-1} & W_{d'_{\hat{L}}-2} & \cdots & W_{d'_{\hat{L}}-d'+1} & W_{d'_{\hat{L}}-d'} \end{bmatrix}.$$

Recalling Figure 3 and the above expression of $T^{\vec{u}}$, we get that for $j = 1, \ldots, \tilde{d}$, the $d'j$-th row of $T^{\vec{u}}$ is exactly the j-th raw of W. If we set $L = \lceil \frac{d'\tilde{d}}{s-1} \rceil$, then $\hat{L} \leq L$. Taking $\vec{w}^{\hat{L}+1} = \cdots = \vec{w}^{L}$ to be the delta sequence, we have $\vec{u} = \vec{w}^{L} * \cdots * \vec{w}^{1}$. But Lemma 11 implies

$$T^{\vec{u}} = T^{\vec{w}^{L}} \cdots T^{\vec{w}^{1}}. \tag{23}$$

Therefore the $d'j$-th item of $T^{\vec{u}}x$ is the j-th item of Wx for $j = 1, \ldots, \tilde{d}$. Set

$$b^{\ell} := 2^{\ell-1} B^{\ell}, \qquad \ell = 1, 2, \ldots, L.$$

Since

$$\vec{w}^{\ell} * \vec{v} = T^{\vec{w}^{\ell}}_{d_\ell, d_{\ell-1}} \vec{v}, \tag{24}$$

it follows from Lemma 12 with $\ell = L-1$ that

$$\begin{aligned}\sigma \circ \mathcal{C}^R_{L-1,\vec{w}^{L-1},b^{L-1}} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^R_{1,\vec{w}^1,b^1}(x) &= \vec{w}^{L-1} * \cdots * \vec{w}^1 * x + b^{L-1}\mathbf{1}_{d_{L-1}} \\ &+ \sum_{k=1}^{L-2} T^{\vec{w}^{L-1}}_{d_{L-1},d_{L-2}} \cdots T^{\vec{w}^{k+1}}_{d_{k+1},d_k} b^k \mathbf{1}_{d_k}.\end{aligned}$$

Therefore, we get from (24) and (23) that

$$T^{\vec{u}}x = \vec{w}^L * \vec{w}^{L-1} * \cdots * \vec{w}^1 * x = \vec{w}^L * \sigma \circ \mathcal{C}^R_{L-1,\vec{w}^{L-1},b^{L-1}} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^R_{1,\vec{w}^1,b^1}(x) - \vec{B}^{d_L}, \quad (25)$$

where

$$\begin{aligned}\vec{B}^{d_L} &:= \sum_{k=1}^{L} T^{\vec{w}^L}_{d_L,d_{L-1}} \cdots T^{\vec{w}^{k+1}}_{d_{k+1},d_k} b^k \mathbf{1}_{d_k} \\ &= w^L * b^{L-1}\mathbf{1}_{d_{L-1}} + w^L * \sum_{k=1}^{L-2} T^{\vec{w}^{L-1}}_{d_{L-1},d_{L-2}} \cdots T^{\vec{w}^{k+1}}_{d_{k+1},d_k} b^k \mathbf{1}_{d_k}.\end{aligned}$$

Step 2. Role of max-pooling: For any $\vec{\theta} = (\theta_1, \ldots, \theta_{\tilde{d}})^T \in \mathbb{R}^{\tilde{d}}$, define $\vec{b}^{d_L} = (b_1, \ldots, b_{d_L})^T$ as the vector satisfying

$$b_k := \begin{cases} (\vec{\theta}^{d_L})_k - (\vec{B}^{d_L})_k, & k = jd' \\ -2(\vec{B}^{d_L})_k, & \text{otherwise} \end{cases}$$

with $(\vec{\theta}^{d_L})_k = \theta_j$ for $k = jd'$ and 0 otherwise, where $(\vec{B}^{d_L})_k$ denotes the k-th component of the vector $\vec{B}^{d_L}$. For $k \neq jd'$, we get from (19) and (25) that

$$\left(\vec{w}^L * \sigma \circ \mathcal{C}^R_{L-1,\vec{w}^{L-1},b^{L-1}} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^R_{1,\vec{w}^1,b^1}(x)\right)_k + b_k = \left(T^{\vec{u}}x\right)_k + (\vec{B}^{d_L})_k + b_k \leq 0, \quad \forall x \in \mathbb{I}^d,$$

which together with the definition of σ yields

$$\sigma\left(\left(\vec{w}^L * \sigma \circ \mathcal{C}^R_{L-1,\vec{w}^{L-1},b^{L-1}} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^R_{1,\vec{w}^1,b^1}(x)\right)_k + b_k\right) = 0.$$

Then, $\sigma(t) \geq 0$ for any $t \in \mathbb{R}$, (25) and the definition of b_k yield

$$\begin{aligned}&\max_{k=jd',\ldots,jd'+d'-1} \left\{\sigma\left(\left(\vec{w}^L * \sigma \circ \mathcal{C}^R_{L-1,\vec{w}^{L-1},b^{L-1}} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^R_{1,\vec{w}^1,b^1}(x)\right)_k + b_k\right)\right\} \\ =\ &\sigma\left(\left(\vec{w}^L * \sigma \circ \mathcal{C}^R_{L-1,\vec{w}^{L-1},b^{L-1}} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^R_{1,\vec{w}^1,b^1}(x)\right)_{jd'} + b_{jd'}\right) \\ =\ &\sigma\left((T^{\hat{u}}x)_{jd'} + (\vec{B}^{d_L})_{jd'} + (\vec{\theta}^{d_L})_{jd'} - (\vec{B}^{d_L})_{jd'}\right). \end{aligned} \quad (26)$$

Step 3. Convolutional representation: Since the $d'j$-th item of $T^{\vec{u}}x$ is the j-th item of Wx and $(\vec{\theta}^{d_L})_{jd'} = \theta_j$, we get from (6) and (26) that

$$\mathcal{S}_{d'_L,d'} \circ \sigma\left(\vec{w}^L * \sigma \circ \mathcal{C}^R_{L-1,\vec{w}^{L-1},b^{L-1}} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^R_{1,\vec{w}^1,b^1}(x) + \vec{b}^{d_L}\right) = \sigma(Wx + \vec{\theta}), \quad (27)$$

which shows that under proper max-pooling scheme, a single fully connected layer can be represented by multiple convolution structures. Let $d' = d$ and $\tilde{d} = 2d + 10$, we have from (27) that there exist $L_1 = \left\lceil \frac{d(2d+10)}{s-1} \right\rceil$ sequences $\{\vec{w}^{\ell,1}\}_{\ell=1}^{L_1}$, a vector $\vec{b}_{d_{L_1},1}$ and $b^{1,1}, \ldots, b^{L_1-1,1} \in \mathbb{R}$ such that

$$\mathcal{S}_{d+L_1 s,d} \circ \sigma \left(\vec{w}^{L_1,1} * \sigma \circ \mathcal{C}^R_{L_1-1,\vec{w}^{L_1-1,1},b^{L_1-1,1}} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^R_{1,\vec{w}^{1},b^{1,1}}(x) + \vec{b}^{d_{L_1,1}} \right) = \sigma(W_1 x + \vec{\theta}_1).$$

Similarly, for $j > 1$, let $d' = \tilde{d} = 2d + 10$, there exist $L_j = \left\lceil \frac{(2d+10)^2}{s-1} \right\rceil$ sequences $\{\vec{w}^{\ell,j}\}_{\ell=1}^{L_j}$, a vector $\vec{b}_{d_{L_j},j}$ and $b^{1,j}, \ldots, b^{L_j-1,j} \in \mathbb{R}$ such that

$$\mathcal{S}_{d+sL_j,2d+10} \circ \sigma \left(\vec{w}^{L_j,1} * \sigma \circ \mathcal{C}^R_{L_j-1,\vec{w}^{L_j-1,1},b^{L_j-1,j}} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{C}^R_{1,\vec{w}^{1},b^{1,j}}(x) + \vec{b}^{d_{L_1,1}} \right) = \sigma(W_j x + \vec{\theta}_j).$$

All these together with (8) and the definition of $\Phi^{pool}_{L,s}$ show

$$\Phi_{L,2d+10,\ldots,2d+10} \subset \Phi^{pool}_{L,s}. \tag{28}$$

Step 4. Approximation rates derivation: For any $f \in Lip^{(r,c_0)}$ with $\|f\|_{L_\infty} \leq M$, we get from (28) and Lemma 7 that

$$\begin{aligned} \text{dist}\left(f, \Phi^{pool}_{L,s}, L^\infty(\mathbb{I}^d)\right) &\leq \text{dist}\left(f, \Phi_{L,2d+10,\ldots,2d+10}, L^\infty(\mathbb{I}^d)\right) \\ &\leq \tilde{C}_1 L^{-2r/d} \log^{2r/d} L. \end{aligned} \tag{29}$$

This proves the upper bound of (10) by noting the definition of the truncation operator π_M. We then turn to proving the lower bound. It follows from Lemma 8 that (15) in Lemma 9 is satisfied with $V = \pi_M \Phi^{pool}_{L,s}$, $\tilde{C}_1 = 1$, $\beta = 0$, $\tilde{C}_2 = M$ and $n = c_1^* L^2 \log L$. Hence, it follows from Lemma 9 that

$$\begin{aligned} \text{dist}(Lip^{(r,c_0)}_M, \pi_M \Phi^{pool}_{L,s}, L_\infty(\mathbb{I}^d)) &\geq \text{dist}(Lip^{(r,c_0)}_M, \pi_M \Phi^{pool}_{L,s}, L_1(\mathbb{I}^d)) \\ &\geq C' \left[L^2 \log L \log(L^2 \log L)\right]^{-\frac{r}{d}} \geq C_1 (L \log L)^{-\frac{2r}{d}}. \end{aligned}$$

This completes the proof of Theorem 1. ■

To prove Theorem 2, we need the following concentration inequality which can be found in (Györfi et al., 2002, Theorem 11.4).

Lemma 13 *Assume $|y| \leq B$ and $B \geq 1$. Let $\mathcal{F}$ be a set of functions $f : \mathbb{I}^d \to \mathbb{R}$ satisfying $|f(x)| \leq B$. Then for each $m \geq 1$, with confidence at least*

$$1 - 14 \max_{x_1^m \in (\mathbb{I}^d)^m} \mathcal{N}_1\left(\frac{\beta\epsilon}{20B}, \mathcal{F}, x_1^m\right) \exp\left(-\frac{\epsilon^2(1-\epsilon)\alpha m}{214(1+\epsilon)B^4}\right),$$

there holds

$$\mathcal{E}(f) - \mathcal{E}(f_\rho) - (\mathcal{E}_D(f) - \mathcal{E}_D(f_\rho)) \leq \epsilon(\alpha + \beta + \mathcal{E}(f) - \mathcal{E}(f_\rho)), \qquad \forall f \in \mathcal{F},$$

where $\alpha, \beta > 0$, $0 < \epsilon \leq 1/2$ and $\mathcal{E}_D(f) = \frac{1}{m}\sum_{i=1}^m (f(x_i) - y_i)^2$.

We then prove Theorem 2 as follows.

Proof [Proof of Theorem 2] Due to (29), there exists a $f_{L,s}^0 \in \Phi_{L,s}^{pool}$ such that for any $f_\rho \in Lip^{(r,c_0)}$ there holds

$$\mathcal{E}(f_{L,s}^0) - \mathcal{E}(f_\rho) = \|f_\rho - f_{L,s}^0\|_{L^\infty(\mathbb{I}^d)}^2 \le \tilde{C}_2 L^{-4r/d} \log^{4r/d} L. \tag{30}$$

The classical error decomposition (Zhou and Jetter, 2006) then yields

$$\begin{aligned}
\mathcal{E}(\pi_M f_{D,L,s}^{pool}) - \mathcal{E}(f_\rho) &\le \mathcal{E}(f_{L,s}^0) - \mathcal{E}(f_\rho) + \overbrace{\mathcal{E}(\pi_M f_{D,L,s}^{pool}) - \mathcal{E}(f_\rho) - (\mathcal{E}_D(\pi_M f_{D,L,s}^{pool}) - \mathcal{E}_D(f_\rho))}^{\mathcal{U}_1} \\
&- \overbrace{[\mathcal{E}(f_{L,s}^0) - \mathcal{E}(f_\rho) - (\mathcal{E}_D(f_{L,s}^0) - \mathcal{E}_D(f_\rho))]}^{\mathcal{U}_2} \\
&\le \tilde{C}_2 L^{-4r/d} \log^{2r/d} L + \mathcal{U}_1 + \mathcal{U}_2.
\end{aligned} \tag{31}$$

The bound of $\mathcal{U}_1$ is well known by noting that $\|f_{D,L,s}^{pool}\|_{L^\infty} \le \|f_{D,L,s}^{pool} - f_\rho\|_{L^\infty} + M$. For example, it was derived in (Zhou and Huo, 2024, Lemma 7) that for any $0 < \delta < 1$, with confidence $1 - \delta$, there holds

$$\mathcal{U}_1 \le 4 \log \frac{2}{\delta} \frac{(\|f_{D,L,s}^{pool} - f_\rho\|_{L^\infty} + M)\|f_{D,L,s}^{pool} - f_\rho\|_{L^\infty}}{\sqrt{m}} \le c_1^* \frac{L^{-2r/d} \log^{2r/d} L}{\sqrt{m}} \log \frac{2}{\delta}, \tag{32}$$

where c_1^* is a constant depending only on r, d, s, M. We then turn to bounding $\mathcal{U}_2$ by using Lemma 13. According to Lemma 8, we have for any $\beta > 0$,

$$\max_{x_1^m \in (\mathbb{I}^d)^m} \mathcal{N}_1\left(\frac{\beta\epsilon}{20B}, \mathcal{F}, x_1^m\right) \le \left(\frac{20M}{\beta\epsilon}\right)^{c_2^* L^2 \log L}$$

for the constant c_2^* depending only on c^*. Then Lemma 13 with $\mathcal{F} = \pi_M \Phi_{L,s}^{pool}$, $\varepsilon = 1/2$, $\beta = 1/n$ yields that with confidence at least

$$1 - 14\,(40Mm)^{c_2^* L^2 \log L} \exp\{-c_3^* \alpha m\},$$

there holds

$$\mathcal{U}_2 \le \frac{1}{2}(\alpha + \frac{1}{m} + \mathcal{E}(\pi_M f_{D,L,s}^{pool}) - \mathcal{E}(f_\rho)),$$

for some c_3^* depending only on M. Let

$$14\,(40Mm)^{c_2^* L^2 \log L} \exp\{-c_3^* \alpha m\} = \delta.$$

We have

$$\alpha = \frac{c_3^*}{m} \log \frac{14}{\delta} + \frac{c_3^* c_2^* L^2 \log L \log(40Mm)}{m}.$$

Therefore, with confidence $1 - \delta$, there holds

$$\mathcal{U}_2 \le \frac{c_4^* L^2 \log L \log m}{m} \log \frac{2}{\delta} + \frac{1}{2}(\mathcal{E}(\pi_M f_{D,L,s}^{pool}) - \mathcal{E}(f_\rho)), \tag{33}$$

where c_4^* is a constant independent of m, δ or L. Inserting (33) and (32) into (31), we get from $2ab \le a^2 + b^2$ for $a, b > 0$ that

$$\begin{aligned}\mathcal{E}(\pi_M f_{D,L,s}^{pool}) - \mathcal{E}(f_\rho) &\le 2\tilde{C}_2 L^{-\frac{4r}{d}} \log^{\frac{2r}{d}} L + 2c_1^* \frac{L^{-\frac{2r}{d}} \log^{\frac{2r}{d}} L}{\sqrt{m}} \log\frac{2}{\delta} + \frac{2c_4^* L^2 \log L \log m}{m} \log\frac{2}{\delta} \\ &\le c_5^* \left(L^{-\frac{4r}{d}} \log^{\frac{2r}{d}} L + \frac{L^2 \log L \log m}{m} \right) \log\frac{2}{\delta}\end{aligned}$$

for c_5^* a constant independent of m, L or δ. Recalling $L \sim m^{\frac{d}{4r+2d}}$, we obtain that with confidence $1-\delta$,

$$\mathcal{E}(\pi_M f_{D,L,s}^{pool}) - \mathcal{E}(f_\rho) \le c_6^* m^{-\frac{2r}{2r+d}} (\log m)^{\max\{2r/d,2\}} \log\frac{2}{\delta},$$

where c_6^* is a constant independent of m, L or δ. This completes the proof of Theorem 2. ■

To prove Corollary 3, we need the following well known probability to expectation formula. We present a simple proof for the sake of completeness.

Lemma 14 *Let $0 < \delta < 1$, and $\xi \in \mathbb{R}_+$ be a random variable. If $\xi \le \mathcal{A} \log^b \frac{c}{\delta}$ holds with confidence $1-\delta$ for some $\mathcal{A}, b, c > 0$, then*

$$E[\xi] \le c\Gamma(b+1)\mathcal{A},$$

where $\Gamma(\cdot)$ is the Gamma function.

Proof Since $\xi \le \mathcal{A} \log^b \frac{c}{\delta}$ holds with confidence $1-\delta$, we have

$$P[\xi > t] \le c \exp\{\mathcal{A}^{-1/b} t^{1/b}\}.$$

Using the probability to expectation formula

$$E[\xi] = \int_0^\infty P\left[\xi > t\right] dt \tag{34}$$

to the positive random variable ξ, we have

$$E[\xi] \le c \int_0^\infty \exp\{\mathcal{A}^{-1/b} t^{1/b}\} \le c\mathcal{A}\Gamma(b+1).$$

This completes the proof of Lemma 14. ■

We then prove Corollary 3 as follows.

Proof [Proof of Corollary 3] The lower bound of (14) is well known and we refer readers to (Györfi et al., 2002, Chap.3) for a detailed proof. The upper bound of (14) follows from (13) and Lemma 14 with $\mathcal{A} = c_6^* m^{-\frac{2r}{2r+d}} (\log m)^{\max\{2r/d,2\}}$, $b = 1$ and $c = 2$ directly. This completes the proof of Corollary 3. ■