Machine Learning
https://doi.org/10.1007/s10994-024-06533-y

Bounding the Rademacher complexity of Fourier neural operators

Taeyoung Kim[1] · **Myungjoo Kang**[1]

Received: 3 October 2022 / Revised: 25 December 2023 / Accepted: 16 February 2024

Abstract
Recently, several types of neural operators have been developed, including deep operator networks, graph neural operators, and Multiwavelet-based operators. Compared with these models, the Fourier neural operator (FNO), a physics-inspired machine learning method, is computationally efficient and can learn nonlinear operators between function spaces independent of a certain finite basis. This study investigated the bounding of the Rademacher complexity of the FNO based on specific group norms. Using capacity based on these norms, we bound the generalization error of the model. In addition, we investigate the correlation between the empirical generalization error and the proposed capacity of FNO. We infer that the type of group norm determines the information about the weights and architecture of the FNO model stored in capacity. The experimental results offer insight into the impact of the number of modes used in the FNO model on the generalization error. The results confirm that our capacity is an effective index for estimating generalization errors.

Keywords Rademacher complexity · Fourier neural operator · Generalization error · Physics-inspired machine learning · Neural operator

1 Introduction

Physics-inspired machine learning is an actively studied area with two approaches to learning. One approach focuses on determining solutions to partial differential equations (PDEs) for fixed PDE and boundary conditions and includes the deep Ritz method (Weinan & Yu, 2018), physics-informed neural networks (Raissi et al., 2019), and least-squares ReLU neural network (Cai et al., 2021). The other approach focuses on the operators between the function

Editor: Pradeep Ravikumar.

✉ Myungjoo Kang
mkang@snu.ac.kr

Taeyoung Kim
legend@snu.ac.kr

[1] Department of Mathematical Science, Seoul National University, Gwanak-ro, Seoul 08826, South Korea

spaces and includes DeepONets (Lu et al., 2021), multiwavelet-based operator (Gupta et al., 2021), graph neural operator (Li et al., 2020) and Fourier neural operator (FNO) Li et al. (2021). This study focuses on FNO, which uses a Fourier transform to manage the convolution operator between two functions quickly and practically. One advantage of FNO is its computational efficiency; unlike DeepONet, its representation is not limited to a finite-dimensional space spanned by a few basis functions. Previous studies (Li et al. (2021) and Pathak et al. (2022)) have confirmed that the FNO successfully approximates numerical solvers and real-world data, verifying its computational efficiency and potential applicability. Unlike real-world machine-learning problems, approximating the solver operator of a PDE is deterministic and concrete. The universal approximation property of the FNO and its approximation error for certain PDE problems (Kovachki et al., 2021) have been verified; however, there are no results estimating the generalization error of the FNO. Moreover, although approximating the solver operator of the PDE is a deterministic problem, we can only provide a finite number of samples to the FNO. Therefore, the accurate inference of hidden data is another problem to consider. Several approaches have been proposed regarding the bounding generalization error of deep neural networks, such as the group norm of weights (Neyshabur et al., 2015), spectral norm (Bartlett et al., 2021), path norm (Neyshabur et al., 2015), Fisher-Rao norm (Liang et al., 2019), and relative flatness (Petzka et al., 2021). In this study, we investigated the bounding of generalization errors within the probably approximately correct (PAC) learning theory framework. In particular, we bound the Rademacher complexity of the FNO.

1.1 Overview of FNOs

Figure 1 illustrates the FNO architecture. The network input is $\mathbb{R}^{d_a}$-valued function in the domain $\tilde{D} \subset \mathbb{R}^d$. We denote the input function space of FNO by $\mathcal{A}(\tilde{D};\mathbb{R}^{d_a})$. The vector value of the input function is lifted to a d_v-dimensional vector using a layer defined as $\mathcal{N}_P$; while passing through the Fourier layers (denoted as $\mathcal{A}_i$ in the diagram) iteratively, it is processed as a $\mathbb{R}^{d_v}$-valued function. Each Fourier layer comprises an activation function, which is the sum of a neural network with a convolution of the input function with a kernel parameterized by weight R_i. After passing through the Fourier layers, the vector value of the $\mathbb{R}^{d_v}$-valued function v_D is projected onto the d_u-dimensional vector using $\mathcal{N}_Q$. We denote the output function space of FNO by $\mathcal{U}(\tilde{D};\mathbb{R}^{d_u})$. The neural network A_i in the

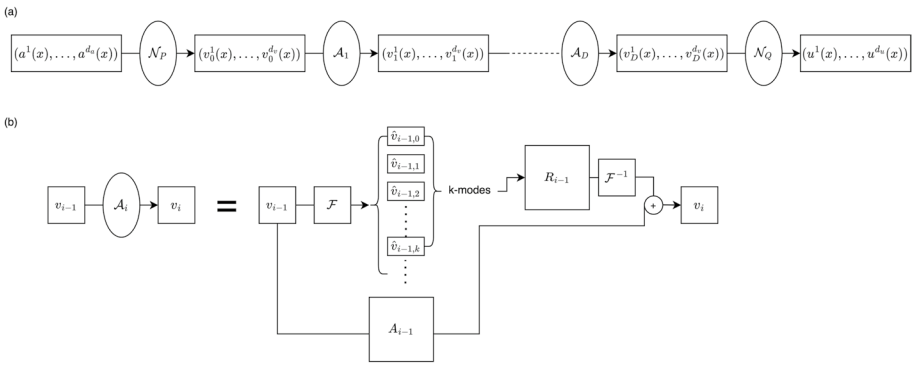

Fig. 1 **a** Sketch of the overall architecture of Fourier neural operator (FNO) **b** Detailed diagram of the Fourier layers

Fourier layers can be chosen arbitrarily. In our results, we chose A_i as a fully connected network (FCN) or convolutional neural network (CNN). Because computational machines cannot handle infinite-dimensional data, we constructed an FNO model using finite parameters based on the above concept, considering real-world implementation.

1.2 Probably approximately correct learning

PAC learning is a framework of statistical learning theory proposed by Valiant (1984). One of the main concepts of PAC learning theory is the no-free-lunch (NFL) theorem, which states that it is not possible to achieve low approximation and estimation errors simultaneously. The trade-off between the two errors is closely related to the complexity of the hypothesis class. Various quantities related to the complexity of the hypothesis class, such as the VC dimension, Rademacher complexity, and Gaussian complexity, determine the learnability and decay of estimation errors. All the complexities are related; however, there are several differences. For example, the VC dimension is independent of the training set, whereas the others are not. Neural networks and deep learning can be applied to PAC learning theory as a subcategory of machine learning. Recently, various studies have investigated bounding the Rademacher complexity and the VC dimensions of the hypothesis class of neural networks. For instance, results regarding the bounding of Rademacher complexities for FCN (Neyshabur et al., 2015), RNN (Minshuo et al., 2020), and GCN (Lv, 2021), and analysis of the VC dimension of neural networks (Sontag, 1998) have been obtained. In addition, there is information about the bounding Rademacher complexity of DeepONet (Gopalani et al., 2022), a kind of neural operator. Reference (Weinan et al., 2020) estimated the generalization error of ResNet in prior and posterior estimates.

1.3 Our contributions

In this study, we defined the capacities of FNO models based on certain group norms. We bound the Rademacher complexity of the hypothesis class based on these capacities for two types of FNOs (Fourier layers with FCN and CNN) and induced the bounding of the posterior generalization error of the FNO models. In Sect. 4, we experiment with data generated from the Burgers equation problem and verify the correlation between our bounding process and empirical generalization errors. Through experiments, we gain insights into the model architecture and weights contained in various capacities. We also qualitatively confirmed that the empirical generalization errors depend on the number of modes used in the FNO model. Furthermore, we confirmed the strong correlation between our capacity factored by dataset size and empirical generalization error on experiments with varying dataset size. We replicate the experiments using other PDE problems. Finally, we compared our capacity with the Hessian trace, Fisher-Rao norm, and relative flatness, showing time, memory efficiency, and effectiveness of our capacity.

2 Preliminary

Notation Several indices are considered in the discussion. Therefore, to simplify the formulas, we denote $x_1 \dots x_d$ as $\mathbf{x}$ and $k_1 \dots k_d$ as $\mathbf{k}$. In addition, for the multi-index tensor in the norm, indices denoted by $\cdot$ are used to calculate the norm; for example,

$$\|A_{xy.}\|_p = \sqrt[p]{\sum_i \left(A_{xyi}\right)^p}.$$

Discretization of data Because the function space is infinite-dimensional, to treat the data and operator numerically, we discretize the function domain and consider the function to be a finite-dimensional vector. Let $\tilde{D}_N = \{x_1, ..., x_N\}$ be the discretization of the domain $\tilde{D} \in \mathbb{R}^d$. Then, the $\mathbb{R}^m$-valued function f is discretized into $(f(x_1), ..., f(x_N))) \in \mathbb{R}^{N\times m}$: Subsequently, we discretize $\mathcal{A}(\tilde{D};\mathbb{R}^{d_a})$ and $\mathcal{U}(\tilde{D};\mathbb{R}^{d_u})$ as $\mathbb{R}^{N\times d_a}$ and $\mathbb{R}^{N\times d_u}$, respectively. Then, sample data are defined as follows: element $((a_{jk}), (u_{jk})) \in \mathbb{R}^{N\times d_a} \times \mathbb{R}^{N\times d_u}$.

Fourier transform Based on the Fourier analysis, we know that the Fourier transform transforms the convolution operation to pointwise multiplication. For the function of domain $\tilde{D} \subset \mathbb{R}^d$, let $\mathcal{F}$ and $\mathcal{F}^{-1}$ be the Fourier and inverse Fourier transforms over $\tilde{D}$, respectively. Thus, we obtain the following relationship:

$$f * k = \mathcal{F}^{-1}(\mathcal{F}(k) \cdot \mathcal{F}(f)).$$

For our analysis, we select $\tilde{D}$ as $[0, 2\pi]^d$. Because we treat functions as discretized vectors, we can treat the Fourier transform as a discrete Fourier transform. If the discretization of $\tilde{D}$ is uniform, it can be replaced by a fast Fourier transform. Consider that $\tilde{D}$ is discretized uniformly by resolution $N_1 \times \cdots \times N_d = N$; then, for discretized function $f \in \mathbb{R}^N$, its FFT $\mathcal{F}(f)(k)$ and IFFT $\mathcal{F}^{-1}(f)(k)$ are defined as follows:

$$\mathcal{F}(f)(k) = \frac{1}{\sqrt{N_1 \dots N_d}} \sum_{x_1}^{N_1} \cdots \sum_{x_d}^{N_d} f(x_1, ..., x_d) e^{-2i\pi \sum_{j=1}^{d} \frac{x_j k_j}{N_j}}$$

$$\mathcal{F}^{-1}(f)(k) = \frac{1}{\sqrt{N_1 \dots N_d}} \sum_{x_1}^{N_1} \cdots \sum_{x_d}^{N_d} f(x_1, ..., x_d) e^{2i\pi \sum_{j=1}^{d} \frac{x_j k_j}{N_j}}.$$

In our analysis, we denote the components of FFT and IFFT tensors as follows: $F_{\mathbf{kx}} = \frac{1}{\sqrt{N_1 \dots N_d}} e^{-2i\pi \sum_{j=1}^{d} \frac{x_j k_j}{N_j}}$, $F^{\dagger}_{\mathbf{xk}} = \frac{1}{\sqrt{N_1 \dots N_d}} e^{2i\pi \sum_{j=1}^{d} \frac{x_j k_j}{N_j}}$, respectively.

Definition 1 (General FNO) Let $\tilde{D}_N$ be the discretized domain in $\mathbb{R}^d$; then, **FNO** : $\mathbb{R}^{N\times d_a} \to \mathbb{R}^{N\times d_u}$ is defined as follows:

$$\mathbf{FNO} = \mathcal{N}_Q \circ \mathcal{A}_D \circ \mathcal{A}_{D-1} \cdots \circ \mathcal{A}_1 \circ \mathcal{N}_P,$$

where $\mathcal{N}_P$ and $\mathcal{N}_Q$ denote the neural networks used for lifting and projection, respectively. Each $\mathcal{A}_i$ is a Fourier layer. For simplicity, we assume that $\mathcal{N}_Q$ and $\mathcal{N}_P$ are linear maps. Each Fourier layer is a composition of the activation function with a sum of convolutions based on a parameterized function and linear map. Only partial frequencies were used in the Fourier layers, expressed as an index set $K = \{(k_1, ..., k_d) \in \mathbb{Z}^d : 0 \le k_j \le k_{max,j}, j = 1, ..., d\}$. The formula for the FNO is

$$
\begin{aligned}
v_0 &:= \mathcal{N}_P(a) = \sum_k P_{jk} a_{\mathbf{x}k} \\
v_{t+1} &:= \mathcal{A}_{t+1}(v_t) = \sigma\left(A_{t+1} v_t + \mathcal{F}^{-1}\left(R_{t+1} \cdot (\mathcal{F}(v_t))\right)\right). \\
&= \sigma\Big(\sum_{\mathbf{z},k} A_{t+1,\mathbf{x}\mathbf{z}jk} v_{t,\mathbf{z}k} + \sum_{\mathbf{z},\mathbf{k}\in K,k} F^{\dagger}_{\mathbf{x}\mathbf{k}} R_{t+1,\mathbf{k},jk} F_{\mathbf{k}\mathbf{z}} v_{t,\mathbf{z}k}\Big) \quad (t = 0, ..., D-1) \\
u &:= \sum_k v_{D,\mathbf{x}k} Q_{kj}.
\end{aligned}
$$

CNN layer For each Fourier layer, a general linear map can be replaced by a CNN layer. A schematic of the convolution with 2D data and a kernel is shown in Fig. 2.

A certain kernel size swipes the input tensors so that each index of outputs has an inner product with the kernel and local components of the input tensor centering the index. For example, for a d-rank input tensor of size $N_1 \times \cdots \times N_d$, we consider a d-rank tensor kernel K of size $c_1 \times \cdots \times c_d$, with c_i less than N_i. Let us denote this CNN layer by the kernel $C(c_1 \times \cdots \times c_d)$; then, the tensor that passes through the CNN layer with K is defined as follows:

$$
C(c_1, \ldots, c_d)(x_{x_1 \cdots x_d})_{z_1 \cdots z_d} = \sum_{j_1=0}^{c_1-1} \cdots \sum_{j_d=0}^{c_d-1} K_{j_1,\ldots,j_d} x_{z_1+j_1,\ldots,z_d+j_d}.
$$

The CNN layers were restricted to kernels of odd sizes to maintain the positional dimension of the tensor. Padding was applied to the input tensor of the CNN layer to fit the dimensions. For example, for $N_1 \times \cdots \times N_d$-dimensional tensor $x_{x_1 \cdots x_d}$ and CNN layer $C(c_1, \ldots, c_d)$, we pad $\frac{c_i-1}{2}$ zeros for each side of the input tensor. We denote this padded

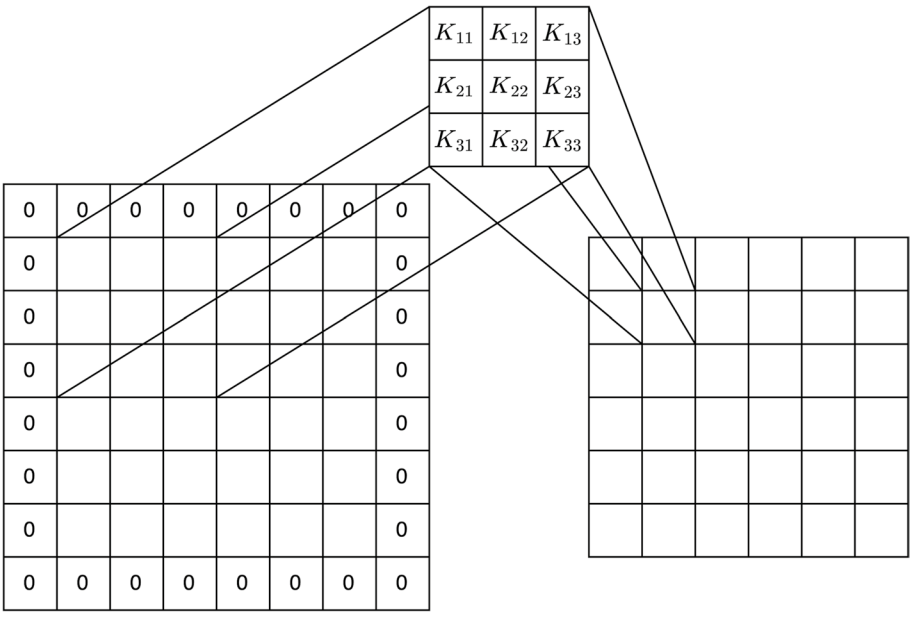

Fig. 2 Schematic of 2D-CNN layer

tensor by $\tilde{x}$. Subsequently, $C(c_1, \ldots, c_d)(\tilde{x}_{x_1 \cdots x_d})$ have the same dimensions as the input tensor. As the number of channels in the Fourier layers is fixed, for a CNN layer with multiple channels, we use the same notation, that is, $C(c_1, \ldots, c_d)$. The formula for multi-channel CNN layer is as follows:

$$C(c_1, \ldots, c_d)(x_{x_1 \cdots x_d})_{z_1 \cdots z_d j} = \sum_{k=1}^{d_u} \sum_{j_1=0}^{c_1-1} \cdots \sum_{j_d=0}^{c_d-1} K_{j_1, \ldots, j_d, k, j} x_{z_1+j_1, \ldots, z_d+j_d, k}.$$

Definition 2 (FNO with CNN layer) Consider the settings of the above FNO; the only difference is that the Fourier layer is the sum of the CNN layer and convolution with parameterized functions.

$$\begin{aligned} v_{t+1} &:= \mathcal{A}_{t+1}(v_t) = \sigma\left(C_{t+1}(c_1, \ldots, c_d)(\tilde{v}_t) + \mathcal{F}^{-1}\left(R_{t+1} \cdot (\mathcal{F}(v_t)) \right) \right) \\ &= \sigma\Big(\sum_{k=1}^{d_u} \sum_{j_1=0}^{c_1-1} \cdots \sum_{j_d=0}^{c_d-1} K_{t+1,jk,j_1,\ldots j_d} \tilde{v}_{t x_1+j_1, \ldots, x_d+j_d, k} \\ &\quad + \sum_{\mathbf{z},\mathbf{k} \in K,k} F_{\mathbf{xk}}{}^{\dagger} R_{t+1,\mathbf{k},jk} F_{\mathbf{kz}} v_{t,\mathbf{z}k} \Big). \end{aligned}$$

An ideal operator should infer a solution from all the functions in the input function space. However, for practical and implementation ease, finite training samples were selected from vector space distributions, which is a discretized function space. Suppose $\mathcal{D}$ is a distribution on $\mathbb{R}^{N \times d_a} \times \mathbb{R}^{N \times d_u}$. Then, we define the loss function as follows:

Definition 3 (Loss for FNO) Suppose that the training dataset is given by

$$S := \{((a_{i,jk}), (u_{i,jk})) \in \mathbb{R}^{N \times d_a} \times \mathbb{R}^{N \times d_u} : i = 1, ..., m\},$$

where each sample is chosen independent of the distribution $\mathcal{D}$. The training loss is defined as follows:

$$\mathcal{L}_S := \frac{1}{m} \sum_{i=1}^{m} \| u_{i,..} - \mathbf{FNO}(a_{i,..}) \|^2.$$

Let p be the probability distribution of $\mathcal{D}$, defined as $\mathbb{R}^{N \times d_a} \times \mathbb{R}^{N \times d_u}$. Then, the loss of the entire distribution $\mathcal{D}$ is defined as follows:

$$\mathcal{L}_{\mathcal{D}} := \int_{\mathbb{R}^{N \times d_a} \times \mathbb{R}^{N \times d_u}} \| u_{i,..} - \mathbf{FNO}(a_{i,..}) \|^2 dp(a, u).$$

3 Generalization bound for FNOs

In this section, we calculate the upper bound of the Rademacher complexity of the FNO, and estimate the generalization bound. In addition, we present several lemmas regarding the main results. The proof of the main theorems comprises two main lemmas: inequality

for the Rademacher complexity and sup-norm of FNO models. Using these lemmas, we prove the main results.

3.1 Mathematical setup

Definition 4 (Rademacher complexity) Let $\mathcal{F}$ represent mapping from $\mathcal{X}$ to $\mathbb{R}$. Suppose $\{x_i \in \mathcal{X} : i = 1, ..., m\}$. ϵ_i are independent and uniform and $\{+1, -1\}$-valued random variables. The empirical Rademacher complexity of $\mathcal{F}$ for a given sample set is defined as follows:

$$\mathcal{R}_m(\mathcal{F}) = \mathbb{E}_\epsilon \left[\frac{1}{m} \sup_{f \in \mathcal{F}} \sum_{i=1}^{m} \epsilon_i f(x_i) \right].$$

The main components of our results are as follows:

Definition 5 (weight norms and capacity) For the multi-rank tensor $M_{i_1,...,i_m,j_1,...,j_k}$, we define the following weight norm:

$$\|M_{i_1,...,i_m,j_1,...,j_k}\|_{p:\{i_1,...,i_m\},q:\{j_1,...,j_k\}} := \sqrt[q]{\sum_{j_1 \ldots j_m} \left(\sqrt[p]{\sum_{i_1 \ldots i_k} M_{i_1,...,i_m,j_1,...,j_k}{}^p} \right)^q}.$$

For $p = \infty$ or $q = \infty$, we consider the sup-norm instead of the definition above. Now, suppose that for an FNO with a Fourier layer of depth D, we denote Q and P as the projection and lifting weight matrices, respectively, and A_i and R_i as the weight tensors of the Fourier layers. We then define $\|\cdot\|_{p,q}$, where p is the index for positions, frequencies, and inputs, and q is the output index. The following norm is defined for the Fourier layer:

$$\|(A_i, R_i)\|_{p,q} := \|A_i\|_{p,q} + \|R_i\|_{p,q} \frac{\sqrt[p*]{k_{max,1} \ldots k_{max,d}}}{N^{\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+}}.$$

The capacity of the FNO model h as a product of the weights of its layers is defined as follows:

$$\gamma_{p,q}(h) := \|P\|_{p,q} \|Q\|_{p,q} \prod_{i=1}^{D} \|(A_i, R_i)\|_{p,q}.$$

Next, for the kernel tensor K of the CNN layer, we define the following norms for the weights and capacities of the entire neural network: In the $\|\cdot\|_{p,q}$ norm for the kernel tensor of the CNN layer, p is the index of the kernels and input, and q is the output index.

$$\|(K_i, R_i)\|_{p,q} := \|K\|_{p,q} \sqrt[p*]{c_1 \ldots c_d} + \sqrt[p*]{k_{max,1} \ldots k_{max,d}} \|R\|_{p,q}$$

$$\gamma_{CNN\,p,q}(h_{CNN}) := \|P\|_{p,q} \|Q\|_{p,q} \| \prod_{i=1}^{D} \|(K_i, R_i)\|_{p,q}.$$

Next, we define hypothesis classes in which the Rademacher complexity is bounded in our results. A hypothesis class is a collection of functions from which a learning algorithm selects a function.

Definition 6 (Hypothesis classes of FNO) Suppose that the function classes of the FNO with D depth and maximal modes of the Fourier layers are $k_{max,1}, ..., k_{max,d}$. The width and size of the input vector, size of the output vector, and activation function are fixed. Then, the hypothesis class for a general FNO is defined as follows:

$$\mathcal{H}^{d_{in}}_{C_P,C_0,...,C_D,C_Q} := \{\mathbf{FNO} : \|P\|_{p,q} \leq C_P, \\ \|(A_i, R_i)\|_{p,q} \leq C_i (i = 1, ..., D), \|Q\|_{p,\infty} \leq C_Q\}.$$

Finally, we define the hypothesis class of the FNO with CNN layers as follows:

$$\mathcal{H}_{CNN}{}^{d_{in}}_{C_P,C_0,...,C_L,C_Q} := \{\mathbf{FNO} : \|P\|_{p,q} \leq C_P, \\ \|(K_i, R_i)\|_{p,q} \leq C_i (i = 1, ..., D), \|Q\|_{p,\infty} \leq C_Q\}.$$

We also define the following auxiliary definition for the hypothesis class of sub-neural networks of FNO models, where the terminal layer is the Fourier layer (denoted as $\mathbf{FNO}_{sub:i}$).

$$\mathcal{H}^{d_{in}}_{C_P,C_0,...,C_i} := \{\mathbf{FNO}_{sub:i} : \|P\|_{p,q} \leq C_P, \|(A_t, R_t)\|_{p,q} \leq C_t, (t = 1, ..., i)\}.$$

Similarly, we define $\mathcal{H}_{CNN}{}^{d_{in}}_{C_P,C_0,...,C_i}$.

3.2 Main results

The notations in each lemma and theorem are based on the definitions in Sect. 3.1. The activation function is Lipschitz continuous and passes through the origin ($\sigma(0) = 0$). Moreover, we set our notations as follows: for a given sample $S = \{a_i\}_{i=1,...,m}$ (with input data a_i) and hypothesis class $\mathcal{H}^{d_{in}}_{C_P,C_0,...,C_t}$, we denote $h(a_i)$ by $v_{t,i}$, where $h \in \mathcal{H}^{d_{in}}_{C_P,C_0,...,C_t}$. The components are denoted by $v_{t,i,\mathbf{xj}}$.

The following lemma regarding l_p norms is frequently used in our proofs.

Lemma 1 (Norm inequality) *If* $1 \leq p \leq q \leq \infty$, *then for* $v \in \mathbb{R}^N$ *we obtain the following inequality*:

$$\|v\|_q \leq \|v\|_p \leq \|v\|_q N^{\frac{1}{p}-\frac{1}{q}}.$$

Let $\lfloor \cdot \rfloor_+$ *denote a ReLU function. Then, for an arbitrary* $1 \leq p, q$, *the inequality can be defined as*

$$\|v\|_p \leq \|v\|_q N^{\lfloor \frac{1}{p}-\frac{1}{q} \rfloor_+}.$$

The following lemma handles nonlinear loss in our proof (the proof can be found in Maurer (2016)).

Lemma 2 (Vector-contraction inequality for Rademacher complexity) *Assume that* σ *is a Lipschitz continuous function with Lipschitz constant L and* $\mathcal{F}$ *is a hypothesis class of* $\mathbb{R}^N$ *-valued functions. Thus, we obtain the following inequality*:

$$\mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{f\in\mathcal{F}}\sum_{i=1}^{m}\epsilon_i\sigma(f(x_i))\right]\le\sqrt{2}L\mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{f\in\mathcal{F}}\sum_{i,k}\epsilon_{ik}f_k(x_i)\right].$$

Here, we present the main results. The proof has two parts. First, we obtain the upper bound of $p*$-norm of the output of FNO models. Second, we bound the Rademacher complexity of the FNO model on samples based on the obtained upper bound. We assume that the projection and lifting layers are linear maps. However, we can easily generalize this to a general FCN.

Lemmas 3 and 3' are the main factors in our results; the Fourier layers are peeled inductively.

Lemma 3 *Suppose* $\mathcal{H}=\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}$ *is the FNO hypothesis class with constants* $C_P,C_1,\ldots,C_D,C_Q$. *Then, for the sample* $a\in\mathbb{R}^{N\times d_a}$, *we obtain the following inequality*:

$$\begin{aligned}&\sup_{h\in\mathcal{H}}\|h(a)_{\cdot\cdot}\|_{p*,\infty}\\&\quad\le L^D(NH)^{D\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}H^{\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}C_QC_D\ldots C_1C_P\|a\|_{p*}\end{aligned}$$

$$\begin{aligned}h(a)_{\mathbf{x}j}&=\sum_k v_{D,\mathbf{x}k}Q_{kj}\\&\le\|v_{D,\mathbf{x}\cdot}\|_{p*}\|Q_{\cdot j}\|_p\end{aligned}\tag{1}$$

Proof

Then, we have the following:

$$\begin{aligned}&\|h(a)_{\cdot\cdot}\|_{p*,\infty}\\&\le\sup_j\sqrt[p*]{\sum_{\mathbf{x}}\|v_{D,\mathbf{x}\cdot}\|_{p*}^{p*}\|Q_{\cdot j}\|_p^{p*}}\\&\le\|v_{D,\cdot\cdot}\|_{p*}C_Q\end{aligned}$$

Subsequently, we peel off the Fourier layers.

$$\begin{aligned}&\sigma\Big(A_D(a)+\mathcal{F}^{-1}(R_D\cdot(\mathcal{F}(a)))\Big)_{\mathbf{x}j}\\&=\sigma\Big(\sum_{\mathbf{z},k}A_{D,\mathbf{xz}kj}v_{D-1,\mathbf{z}k}+\sum_{\mathbf{k},\mathbf{z},k}F^\dagger_{\mathbf{xk}}R_{D,\mathbf{k},jk}F_{\mathbf{kz}}v_{D-1\mathbf{z}k}\Big)\\&\le L\Big|\sum_{\mathbf{z},k}A_{D,\mathbf{xz}kj}v_{D-1,\mathbf{z}k}+\sum_{\mathbf{k},\mathbf{z},k}F^\dagger_{\mathbf{xk}}R_{D,\mathbf{k},jk}F_{\mathbf{kz}}v_{D-1,\mathbf{z}k}\Big|\\&\le L\Big(\|A_{D,\mathbf{x}\cdot j}\|_p+\Big\|\sum F^\dagger_{\mathbf{xk}}R_{D,\mathbf{k},j\cdot}F_{\mathbf{k}\cdot}\Big\|_p\Big)\Big\|v_{D-1,\cdot\cdot}\Big\|_{p*},\end{aligned}\tag{2}$$

For $\Big\|\sum F^\dagger_{\mathbf{xk}}R_{D,\mathbf{k},j\cdot}F_{\mathbf{k}\cdot}\Big\|_p$ in (2),

$$\Big\|\sum_{\mathbf{k}}F^\dagger_{\mathbf{xk}}R_{D,\mathbf{k},j\cdot}F_{\mathbf{k}\cdot}\Big\|_p=\sqrt[p]{\sum_{\mathbf{z},k}\Big(\sum_{\mathbf{k}}F^\dagger_{\mathbf{xk}}R_{D,\mathbf{k},jk}F_{\mathbf{kz}}\Big)^p}.$$

For fixed $\mathbf{x}, \mathbf{z}, k$, $\left(F^{\dagger}_{\mathbf{xk}} F_{\mathbf{kz}}\right)_{\mathbf{k}}$ is a $k_{max,1}, \ldots, k_{max,d}$-dimensional vector, where each component exhibits the $\frac{e^{ib}}{N}$ form. Thus, by applying Hölder's inequality, we obtain the following inequality:

$$\begin{aligned}
&\Big\| \sum_{\mathbf{k}} F^{\dagger}_{\mathbf{xk}} R_{D,\mathbf{k},j} . F_{\mathbf{k}\cdot} \Big\|_p \\
&\leq \sqrt[p]{\sum_{\mathbf{z},k} \left(\frac{\sqrt[p*]{k_{max,1} \ldots k_{max,d}}}{N} \|R_{D,\cdot,jk}\|_p \right)^p} \\
&= \frac{\sqrt[p*]{k_{max,1} \ldots k_{max,d}}}{N} \sqrt[p]{N \sum_{\mathbf{k},k} R^p_{D,\mathbf{k},jk}} \\
&= \sqrt[p*]{\frac{k_{max,1} \ldots k_{max,d}}{N}} \|R_{D,\cdot,j,\cdot}\|_p .
\end{aligned}$$

Subsequently, the following bound is obtained:

$$\begin{aligned}
&\sigma\Big(A_D(a) + \mathcal{F}^{-1}(R_D \cdot (\mathcal{F}(a))) \Big)_{\mathbf{x}j} \\
&\leq L\Big(\|A_{D,\mathbf{x}\cdot,j}\|_p + \sqrt[p*]{\frac{k_{max,1} \ldots k_{max,d}}{N}} \|R_{D,\cdot,j,\cdot}\|_p \Big) \Big\| v_{D-1,\cdot\cdot} \Big\|_{p*} .
\end{aligned}$$

We iteratively apply the above bound to obtain the following inequality:

$$\begin{aligned}
&\sup_{h \in \mathcal{H}_{C_P, C_0, \ldots, C_D}} \|v_{D,\cdot\cdot}\|_{p*} \\
&\leq \sup_{h \in \mathcal{H}_{C_P, C_1, \ldots, C_D}} L\Big(\|A_D\|_{p,p*} + \sqrt[p*]{k_{max,1} \ldots k_{max,d}} \|R_D\|_{p,p*} \Big) \Big\| v_{D-1,\cdot\cdot} \Big\|_{p*} \\
&\leq (NH)^{\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} \sup_{h \in \mathcal{H}_{C_P, C_1, \ldots, C_D}} \\
&L\Big(\|A_D\|_{p,q} + \frac{\sqrt[p*]{k_{max,1} \ldots k_{max,d}}}{N^{\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+}} \|R_D\|_{p,q} \Big) \Big\| v_{D-1,\cdot\cdot} \Big\|_{p*}
\end{aligned} \tag{3}$$

$$\begin{aligned}
&\leq L(NH)^{\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} C_D \sup_{h \in \mathcal{H}_{C_P, C_1, \ldots, C_{D-1}}} \Big\| v_{D-1,\cdot\cdot} \Big\|_{p*} \\
&\leq \ldots \\
&\leq L^D (NH)^{D \lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} C_D \ldots C_1 \sup_{h \in \mathcal{H}_{C_P}} \Big\| v_{1,\cdot\cdot} \Big\|_{p*} \\
&\leq L^D (NH)^{D \lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} H^{\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} C_D \ldots C_1 C_P \|a\|_{p*} .
\end{aligned} \tag{4}$$

By combining the two inequalities, we obtain the following inequality.

$$
\begin{aligned}
&\|h(a)_{..}\|_{p*,\infty} \\
&\leq \|v_{D,..}\|_{p*} C_Q \\
&\leq L^D (NH)^{D\lfloor \frac{1}{p*}-\frac{1}{q} \rfloor_+} H^{\lfloor \frac{1}{p*}-\frac{1}{q} \rfloor_+} C_Q C_D \ldots C_1 C_P \|a\|_{p*}
\end{aligned}
$$

We use Hölder's inequality in (1) and (2) and norm inequalities in (3) and (4). □

The proof of the following lemma is similar to that of Lemma 3. However, in this case, the hypothesis class is FNO with CNN layers.

Lemma 3′ *Suppose $\mathcal{H} = \mathcal{H}_{CNN\,C_P,C_1,\ldots,C_D,C_Q}^{d_{in}}$ is the hypothesis class of an FNO with CNN layer and constants $C_P, C_1, \ldots, C_D, C_Q$. Then, for a sample $a \in \mathbb{R}^{N \times d_a}$, we obtain the following inequality*:

$$
\begin{aligned}
&\sup_{h \in \mathcal{H}} \|h(a)_{..}\|_{p*,\infty} \\
&\quad \leq L^D H^{(D+1)\lfloor \frac{1}{p*}-\frac{1}{q} \rfloor_+} C_Q C_D \ldots C_1 C_P \|a\|_{p*}
\end{aligned}
$$

Proof We modify the induction parts of the Fourier layers in the proof of Lemma 3.

$$
\begin{aligned}
&\sigma\Big(C_D(a) + \mathcal{F}^{-1}(R_D \cdot (\mathcal{F}(a)))\Big)_{\mathbf{x}j} \\
&= \sigma\Big(\sum_{j_1=0}^{c_1-1} \cdots \sum_{j_d=0}^{c_d-1} \sum_{k=1}^{d_v} K_{D,jk,j_1,\ldots,j_d} v_{D-1,x_1+j_1,\ldots,x_d+j_d k} \\
&\quad + \sum F^{\dagger}_{\mathbf{xk}} R_{D,\mathbf{k},jk} F_{\mathbf{k} z_1 \ldots z_d} v_{D-1,i,z_1 \ldots z_d k} \Big) \\
&\leq L\Bigg(\|K_{D,j,\ldots}\|_p \Big\| v_{D-1,x_1+\cdot,\ldots,x_d+\cdot,\cdot} \Big\|_{p*} \\
&\quad + \sqrt[p*]{\frac{k_{max,1} \ldots k_{max,d}}{N}} \|R_{D,\cdot,j,\cdot}\|_p \Big\| v_{D-1,..} \Big\|_{p*} \Bigg).
\end{aligned} \tag{5}
$$

where we use Hölder's inequality in (5). Subsequently, by applying $p*$ norm to the inequality above over $\mathbf{x}, j$ and the norm inequality, we obtain the following inequality:

$$
\begin{aligned}
&\Big\| \sigma\Big(C_D(a) + \mathcal{F}^{-1}(R_D \cdot (\mathcal{F}(a)))\Big)_{..} \Big\|_{p*} \\
&\leq L\Bigg(\sqrt[p*]{\sum_j \|K_{D,j,\ldots}\|_p^{p*} \sum_{\mathbf{x}} \sum_{j_1=0}^{c_1-1} \cdots \sum_{j_d=0}^{c_d-1} \sum_{k=1}^{d_v} \|v_{D-1,x_1+\cdot,\ldots,x_d+\cdot,\cdot}\|^{p*}} \\
&\quad + \sqrt[p*]{k_{max,1} \ldots k_{max,d}} \|R_D\|_{p,q} H^{\lfloor \frac{1}{p*}-\frac{1}{q} \rfloor_+} \Big\| v_{D-1,..} \Big\|_{p*} \Bigg) \\
&\leq L H^{\lfloor \frac{1}{p*}-\frac{1}{q} \rfloor_+} \Big(\sqrt[p*]{c_1 \ldots c_d} \|K_D\|_{p,q} + \sqrt[p*]{k_{max,1} \ldots k_{max,d}} \|R_D\|_{p,q} \Big) \Big\| v_{D-1,..} \Big\|_{p*}.
\end{aligned}
$$

The remainder of this proof is similar to that of Lemma 3. □

Lemma 4 *Suppose* $\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}$ *is the hypothesis class of the FNO with given constants* $C_P, C_1, \ldots, C_D, C_Q$. *Then, for samples* $S = \{a_i\}_{i=1,\ldots,m}$, *we obtain the following inequality*:

$$\mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{h\in\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right] \le \frac{N^{\frac{1}{p}}d_u}{m}\sum_i \sup_{h\in\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}}\|h(a_i)_{..}\|_{p*,\infty}.$$

Proof

$$\begin{aligned}
&\mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{h\in\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right]\\
&\le \mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{h\in\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}}\sum_{i,\mathbf{x},j}\left|h(a_i)_{\mathbf{x}j}\right|\right]\\
&\le \frac{N^{\frac{1}{p}}d_u}{m}\mathbb{E}_\epsilon\left[\sup_{h\in\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}}\sum_i\|h(a_i)_{..}\|_{p*,\infty}\right]\\
&\le \frac{N^{\frac{1}{p}}d_u}{m}\sum_i\sup_{h\in\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}}\|h(a_i)_{..}\|_{p*,\infty}
\end{aligned} \tag{6}$$

where we used norm inequality in (6). □

Theorem 1 *Suppose* $\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}$ *is a hypothesis class with constants* $C_P, C_1, \ldots, C_D, C_Q$. *Then, for samples* $S = \{a_i\}_{i=1,\ldots,m}$, *we obtain the following inequality*:

$$\begin{aligned}
&\mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{h\in\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right]\\
&\le L^D(NH)^{D\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}H^{\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_uC_QC_D\ldots C_1C_P\frac{1}{m}\sum_{i=1}^m\|a_i\|_{p*}.
\end{aligned}$$

Proof

$$\begin{aligned}
&\mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{h\in\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right]\\
&\le N^{\frac{1}{p}}d_u\frac{1}{m}\sum_{i=1}^m\sup_{h\in\mathcal{H}_{C_P,C_1,\ldots,C_D,C_Q}}\|h(a_i)_{..}\|_{p*,\infty} && \text{(Lemma 4)}\\
&\le L^D(NH)^{D\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}H^{\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_uC_QC_D\ldots C_1C_P\frac{1}{m}\sum_{i=1}^m\|a_i\|_{p*}. && \text{(Lemma 3)}
\end{aligned}$$

□

Theorem 2 (FNO with CNN layer) *Suppose* $\mathcal{H}_{CNN}{}^{d_{in}}_{C_P,C_1,\ldots,C_D,C_Q}$ *is a hypothesis class with constants* $C_P, C_1, \ldots, C_D, C_Q$. *Then, for samples* $S = \{a_i\}_{i=1,\ldots,m}$, *we obtain the following inequality*:

$$\mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{h\in\mathcal{H}^{d_{in}}_{CNN\,C_P,C_1,\ldots,C_D,C_Q}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right]$$
$$\leq L^D H^{(D+1)\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_u C_Q C_D\ldots C_1 C_P\frac{1}{m}\sum_{i=1}^{m}\|a_i\|_{p*}.$$

Proof

$$\mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{h\in\mathcal{H}^{d_{in}}_{CNN\,C_P,C_1,\ldots,C_D,C_Q}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right]$$
$$\leq N^{\frac{1}{p}}d_u\frac{1}{m}\sum_{i=1}^{m}\sup_{h\in\mathcal{H}_{CNN\,C_P,C_1,\ldots,C_D,C_Q}}\|h(a_i)_{..}\|_{p*,\infty}\qquad\text{(Lemma4)}$$
$$\leq L^D H^{(D+1)\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_u C_Q C_D\ldots C_1 C_P\frac{1}{m}\sum_{i=1}^{m}\|a_i\|_{p*}.\qquad\text{(Lemma 3')}$$

□

Corollary 1 *For a constant $\gamma>0$, consider the hypothesis class $\mathcal{H}_{\gamma_{p,q}\leq\gamma}$, which is a collection of FNOs with $\gamma_{p,q}\leq\gamma$. For samples $S=\{a_i\}_{i=1,\ldots,m}$, we obtain the following inequality*:

$$\mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{h\in\mathcal{H}_{\gamma_{p,q}\leq\gamma}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right]$$
$$\leq\gamma L^D(NH)^{D\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}H^{\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_u\frac{1}{m}\sum_{i=1}^{m}\|a_i\|_{p*}.$$

For a given hypothesis class $\mathcal{H}_{CNN\gamma_{p,q}\leq\gamma}$, similar to $\mathcal{H}_{\gamma_{p,q}\leq\gamma}$, and training samples $S=\{a_i\}_{i=1,\ldots,m}$, we obtain the following inequality:

$$\mathbb{E}_\epsilon\left[\frac{1}{m}\sup_{h\in\mathcal{H}_{CNN\gamma_{p,q}\leq\gamma}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right]$$
$$\leq\gamma_{CNN}L^D H^{(D+1)\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_u\frac{1}{m}\sum_{i=1}^{m}\|a_i\|_{p*}.$$

Proof As

$$\mathcal{H}_{\gamma_{p,q}\leq\gamma}\subset\bigcup_{0\leq C_P C_1\ldots C_D C_Q<\gamma}\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_Q}.$$

We obtain the following inequalities.

$$\mathbb{E}_{\epsilon}\left[\frac{1}{m}\sup_{h\in\mathcal{H}_{\gamma_{p,q}\leq\gamma}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right]$$
$$\leq \mathbb{E}_{\epsilon}\left[\frac{1}{m}\sup_{h\in\bigcup_{0\leq C_P C_1\ldots C_D C_Q<\gamma}\mathcal{H}^{d_{in}}_{C_P,C_1,\ldots,C_Q}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right]$$

Because the upper bound of $p*$-norm of the models of the hypothesis class in the above equation is the same as that in Lemma 3, we apply the same logic as in Theorem 1. Thus, we obtain the following inequality:

$$\mathbb{E}_{\epsilon}\left[\frac{1}{m}\sup_{h\in\mathcal{H}_{\gamma_{p,q}\leq\gamma}}\sum_{i,\mathbf{x},j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right]$$
$$\leq \gamma L^D (NH)^{D\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+} H^{\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+} N^{\frac{1}{p}} d_u \frac{1}{m}\sum_{i=1}^{m}\|a_i\|_{p*}.$$

Similarly, based on the above proof, we obtain the inequality for FNO with CNN layers. □

Recall the following fundamental theorem (for details, see Shalev-Shwartz and Ben-David (2014)) that states the statistical estimation of the generalization error bound of a given hypothesis class in terms of Rademacher complexity.

Theorem 3 (Generalization error bounding based on Rademacher complexity) *Given hypothesis class $\mathcal{H}$ and loss function $l : \mathcal{H}\times Z$ that satisfy the following case: for all $h\in\mathcal{H}$ and $z\in Z$, we obtain $|l(h,z)|\leq c$. Then, with a probability of at least $1-\delta$, for all $h\in\mathcal{H}$, we obtain*

$$\mathbb{E}_{\mathcal{D}}[l(h,z)] - \mathbb{E}_S[l(h,z)] \leq 2\mathcal{R}_m(l\circ\mathcal{H}) + c\sqrt{\frac{2\log 4/\delta}{m}}.$$

where $\mathcal{D}$ is the probability distribution on Z and S is a training dataset sampled from $\mathcal{D}$ i.i.d.

Before considering the generalization bound of FNO, we select the distribution $\mathcal{D}$ on $\mathbb{R}^{N\times d_a}\times\mathbb{R}^{N\times d_u}$ to have a compact support. Thus, $|l(h,z)|\leq c$ condition in Theorem 3 holds. Then, using Theorem 3 and Corollary 1, we obtain the following estimation of the generalization error bound:

Theorem 4 (Generalization error bound for FNO) *For the training dataset $S=\{(a_i,u_i)\}_{i=1,\ldots,m}$, sampled from probability distribution $\mathcal{D}$ i.i.d., and for hypothesis class $\mathcal{H}_{\gamma_{p,q}\leq\gamma}$, let $h^\star$ be the ERM minimizer of L_S and $\|h(a)-u\|_2\leq\epsilon^2$ for all $(a,u)\sim\mathcal{D}$, $h\in\mathcal{H}_{\gamma_{p,q}\leq\gamma}$. Subsequently, with a probability of at least $1-\delta$, we obtain the following inequality*:

$$L_{\mathcal{D}}(h^\star) - L_S(h^\star)$$
$$\leq 4\sqrt{2}\epsilon\gamma L^D (NH)^{D\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+} H^{\lfloor\frac{1}{p*}-\frac{1}{q}\rfloor_+} N^{\frac{1}{p}} d_u \frac{1}{m}\sum_{i=1}^{m}\|a_i\|_{p*} + \epsilon^2\sqrt{\frac{2\log 4/\delta}{m}}.$$

Similarly for hypothesis class of FNOs with CNN layers, dataset S, and hypothesis class $\mathcal{H}_{CNN\gamma_{p,q}\leq\gamma}$, *let* $h^{\star}_{CNN}$ *be the ERM minimizer of* L_S *and* $\|h(a)-u\|_2 \leq \epsilon^2$ *for all* $(a,u)\sim\mathcal{D}$, $h\in\mathcal{H}_{CNN\gamma_{p,q}\leq\gamma}$. *Subsequently, with a probability of at least* $1-\delta$, *we obtain the following inequality*:

$$
\begin{aligned}
&L_{\mathcal{D}}(h^{\star}_{CNN}) - L_S(h^{\star}_{CNN}) \\
&\quad\leq 4\sqrt{2}\epsilon\gamma_{CNN}L^D H^{(D+1)\lfloor\frac{1}{p^*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_u\frac{1}{m}\sum_{i=1}^{m}\|a_i\|_{p^*} + \epsilon^2\sqrt{\frac{2\log 4/\delta}{m}}.
\end{aligned}
$$

Proof We just need to calculate $\mathcal{R}_m(l\circ\mathcal{H})$ term in Theorem 3.

$$
\begin{aligned}
\mathcal{R}_m(l\circ\mathcal{H}_{\gamma_{p,q}\leq\gamma(\mathcal{N}_{FNO})}) &\leq 2\sqrt{2}\epsilon\mathbb{E}_{\epsilon}\left[\frac{1}{m}\sup_{h\in\mathcal{H}_{\gamma_{p,q}\leq\gamma}}\sum_{i,\mathbf{x}_j}\epsilon_{i\mathbf{x}j}h(a_i)_{\mathbf{x}j}\right] && \text{(Lemma 2)}\\
&\leq 2\sqrt{2}\epsilon\gamma L^D(NH)^{D\lfloor\frac{1}{p^*}-\frac{1}{q}\rfloor_+}H^{\lfloor\frac{1}{p^*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_u\frac{1}{m}\sum_{i=1}^{m}\|a_i\|_{p^*}. && \text{(Corollary 1)}
\end{aligned}
$$

Similarly, based on the above proof, we obtain the inequality for the FNO with CNN layers. □

If the capacity of FNO model h is γ, it is included in the hypothesis class $\mathcal{H}_{\gamma_{p,q}\leq\gamma}$. Because the inequalities in Theorem 4 hold for all hypotheses in class, we have the following posterior estimate of FNO:

Corollary 2 (Posterior estimation of generalization and expected errors) *Given architecture parameters* N, H, d_u, d_a, L, *and training samples* $\{(a_i,u_i)\}_{i=1,\ldots,m}$ *with* $\|a_i\|_{p^*}\leq B$ *for all* i. *Suppose* h *is a trained FNO (Fourier layer with FCN or CNN) such that* $\|h(a)-u\|_2\leq\epsilon^2$ *for all training samples. Then, with a confidence level of at least* $1-\delta$, *we obtain the following estimates*:

$$
\begin{aligned}
&L_{\mathcal{D}}(h) - L_S(h) \\
&\quad\leq 4\sqrt{2}\epsilon L^D(NH)^{D\lfloor\frac{1}{p^*}-\frac{1}{q}\rfloor_+}H^{\lfloor\frac{1}{p^*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_u\gamma_{p,q}(h)B + \epsilon^2\sqrt{\frac{2\log 4/\delta}{m}}.\\
&\Longrightarrow L_{\mathcal{D}}(h) \leq 4\sqrt{2}\epsilon L^D(NH)^{D\lfloor\frac{1}{p^*}-\frac{1}{q}\rfloor_+}H^{\lfloor\frac{1}{p^*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_u\gamma_{p,q}(h)B\\
&\qquad+\epsilon^2\left(1+\sqrt{\frac{2\log 4/\delta}{m}}\right).
\end{aligned}
$$

for FNOs with CNN,

$$
\begin{aligned}
&L_{\mathcal{D}}(h_{CNN}) - L_S(h_{CNN}) \\
&\quad\leq 4\sqrt{2}\epsilon L^D H^{(D+1)\lfloor\frac{1}{p^*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_u\gamma_{CNNp,q}(h_{CNN})B + \epsilon^2\sqrt{\frac{2\log 4/\delta}{m}}.\\
&\Longrightarrow L_{\mathcal{D}}(h_{CNN}) \leq 4\sqrt{2}\epsilon L^D H^{(D+1)\lfloor\frac{1}{p^*}-\frac{1}{q}\rfloor_+}N^{\frac{1}{p}}d_u\gamma_{CNNp,q}(h_{CNN})B\\
&\qquad+\epsilon^2\left(1+\sqrt{\frac{2\log 4/\delta}{m}}\right).
\end{aligned}
$$

Our definitions of capacity and results are motivated by the group-norm capacity of an FCN (Neyshabur et al., 2015). The results in Neyshabur et al. (2015) imposed the upper bound of the generalization error on the capacity and inverse factor of $\sqrt{m}$. The proof of this upper bound relies on the homogeneity of the ReLU activation function. Our results apply only to the Lipschitzness of the activation function. Although our results only have the O(1) bound, if we focus on the ReLU activation case, we may have the following bound based on our capacity derivation and theorems in Neyshabur et al. (2015).

Corollary 3 (Posterior estimation of generalization error and expected error in the RELU activation case) *Given architecture parameters* N, H, d_u, d_a, L, *and training samples* $\{(a_i, u_i)\}_{i=1,\ldots,m}$ *with* $\|a_i\|_{p*} \leq B$ *for all* i. *Suppose* h *is a trained FNO* (*Fourier layer with FCN or CNN*) *such that* $\|h(a) - u\|_2 \leq \epsilon^2$ *for all training samples and* $1 \leq p \leq 2$, $1 \leq q \leq p*$. *Then, with a confidence level of at least* $1 - \delta$, *we obtain the following estimates*:

$$
\begin{aligned}
&L_{\mathcal{D}}(h) - L_S(h) \\
&\quad \leq 4\sqrt{2}\epsilon L^D (NH)^{D\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} H^{\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} N^{\frac{1}{p}} d_u \gamma_{p,q}(h) \frac{min\{p*, 4log(2d_a)\}B}{\sqrt{m}} \\
&\qquad + \epsilon^2 \sqrt{\frac{2\log 4/\delta}{m}}. \\
&\quad \Longrightarrow L_{\mathcal{D}}(h) \leq 4\sqrt{2}\epsilon L^D (NH)^{D\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} H^{\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} N^{\frac{1}{p}} d_u \gamma_{p,q}(h) \frac{min\{p*, 4log(2d_a)\}B}{\sqrt{m}} \\
&\qquad + \epsilon^2 \left(1 + \sqrt{\frac{2\log 4/\delta}{m}}\right).
\end{aligned}
$$

For FNOs with CNN,

$$
\begin{aligned}
&L_{\mathcal{D}}(h_{CNN}) - L_S(h_{CNN}) \\
&\quad \leq 4\sqrt{2}\epsilon L^D H^{(D+1)\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} N^{\frac{1}{p}} d_u \gamma_{CNN_{p,q}}(h_{CNN}) \frac{min\{p*, 4log(2d_a)\}B}{\sqrt{m}} \\
&\qquad + \epsilon^2 \sqrt{\frac{2\log 4/\delta}{m}}. \\
&\quad \Longrightarrow L_{\mathcal{D}}(h_{CNN}) \leq 4\sqrt{2}\epsilon L^D H^{(D+1)\lfloor \frac{1}{p*} - \frac{1}{q} \rfloor_+} N^{\frac{1}{p}} d_u \gamma_{CNN_{p,q}}(h_{CNN}) \frac{min\{p*, 4log(2d_a)\}B}{\sqrt{m}} \\
&\qquad + \epsilon^2 \left(1 + \sqrt{\frac{2\log 4/\delta}{m}}\right).
\end{aligned}
$$

Therefore, for the ReLU activation case, we can guarantee convergence of the generalization error with increasing training dataset size. However, the actual convergence rate of the generalization error is higher than the theoretical bound, as observed in Sect. 4.

4 Experiments

This section validates the experimental results. In addition, we show that the capacity we defined is an effective index for estimating empirical generalization errors in various respects.

4.1 Overall correlation over various p and q

First, we investigate the correlation between our capacity and the empirical generalization errors for various capacities of p and q.

Data specification For the experiment, we synthesized a dataset based on the following Burgers equation:

$$u_t = -uu_x + 0.01uu_{xx}$$

The domain of the problem is a circle; we uniformly discretize the domain by $N = 1024$. As described in Sect. 2, each data point represents a pair of functions. In our experiment, the input function was an initial condition, and the target function was a solution to the above equation at $t = 0.1$. Each input function was generated from Gaussian random fields with covariance $k(x, y) = e^{-\frac{(x-y)^2}{(0.05)^2}}$. The training and test datasets comprise 800 and 200 pairs of functions, respectively (both generated independently).

Correlation for various capacities of p and q We investigated the correlation between the generalization error and capacities. Each point in Fig. 3 represents a trained model for the randomly chosen hyperparameters. The architecture of the models used in our experiment was organized as follows: 2-depth Fourier layers, linear layers without projection activation, and lifting layers. The width is fixed at 64. The weight decay for each training session was randomly chosen from 0, 2*1e−2, 4*1e−2, 6*1e−2, and 8*1e−2; k_{max} was randomly chosen from 8, 12, 16, and 20; the kernel size was randomly chosen from 1, 3, 5, and 7 for 100 iterations.

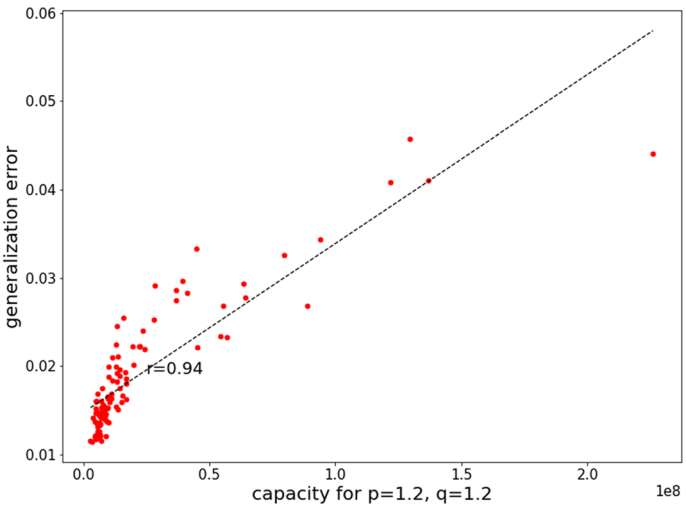

Fig. 3 Scatter plot of generalization error versus capacity for $p = 1.2, q = 1.2$

Table 1 Correlation between empirical generalization error and capacities of various p and q for trained models with randomly chosen hyperparameters

	$p=1$	$p=1.2$	$p=1.6$	$p=2$	$p=4$	$p=\infty$
$q=1$	0.8757	0.9137	0.7794	0.7595	0.7542	0.7285
$q=1.2$	0.8395	**0.9358**	0.8007	0.7635	0.7526	0.7265
$q=1.6$	0.8127	0.9007	0.8476	0.7750	0.7495	0.7231
$q=2$	0.8037	0.8720	0.8815	0.7860	0.7466	0.7204
$q=4$	0.7919	0.8417	0.9084	0.7938	0.7322	0.7112
$q=\infty$	0.7555	0.8229	0.8859	0.7765	0.7235	0.7219

Bold value indicates the highest one

Table 1 lists the correlations for the various values of p and q. The correlation decreases with increasing p and q. This is because as p increases, the p-norm loses information about elements other than the highest norm. Thus, the information of the model is lost in a capacity defined by high values of p and q. However, as p goes to ∞, $p*$ reaches 1; thus, kernel size and k_{max} have a greater effect on capacity as p increases. Therefore, we assume that the capacity of a high p contains more information regarding model architecture. To prove our arguments, we conducted experiments in which k_{max} was varied and the other hyperparameters were fixed. First, to show that the capacities of low p and q contain more information about the model weights than its architecture, we trained three models with negligible differences in k_{max}. Second, to demonstrate that the capacities of high p and q are more closely related to the model architecture, we trained three types of models with considerable differences in k_{max}. For each experiment, we trained the models 30 times for each k_{max} setting, that is, 14, 16, and 18 in the left column of Fig. 4 and 10, 30, and 50 in the right column. Hyperparameters other than k_{max} were fixed: the kernel size of the CNN layer was 1, the width was 64, and the depth of the Fourier layers was 2. As revealed by the left column of Fig. 4, models with small gaps in k_{max} lose the correlation between the generalization gap and capacity with increasing p and q. However, in the right column, the highest correlation between the capacity and generalization error is obtained for higher p and q values compared to those in the left column. The correlation is maintained at 0.89 for the $p, q = \infty$ case.

4.2 Dependency of generalization errors on architectures and datasets

4.2.1 Dependency on k_{max}

Next, we examined the dependency of the generalization error on the model architecture. In the experiments, hyperparameters other than k_{max} are fixed. We consider two cases: Fourier layers at depths of 1 and 2.

A low k_{max} implies that the dynamics of learning are unpredictable and chaotic (Seleznova & Kutyniok, 2021); therefore, we did not consider models with extremely small values of k_{max}. We varied k_{max} from 13 to 39 in two intervals. For a detailed analysis, we removed the CNN layer parts from the Fourier layers, such that the generalization error is proportional to the weight norm R_i. To verify the influence of k_{max} size, we divided the generalization error by R_i. As the defined capacity is correlated with the generalization error, it is expected that the divided generalization error is correlated with $\sqrt[p*]{k_{max}}$ at a depth of 1 and $\sqrt[p*]{\sqrt{k_{max}}}$ at a depth of 2. As listed in Tables 2 and 3, the generalization error divided by the norms is correlated to $\sqrt[p*]{k_{max}}$ and $\sqrt[p*]{\sqrt{k_{max}}}$. Hence, we can verify that the correlation

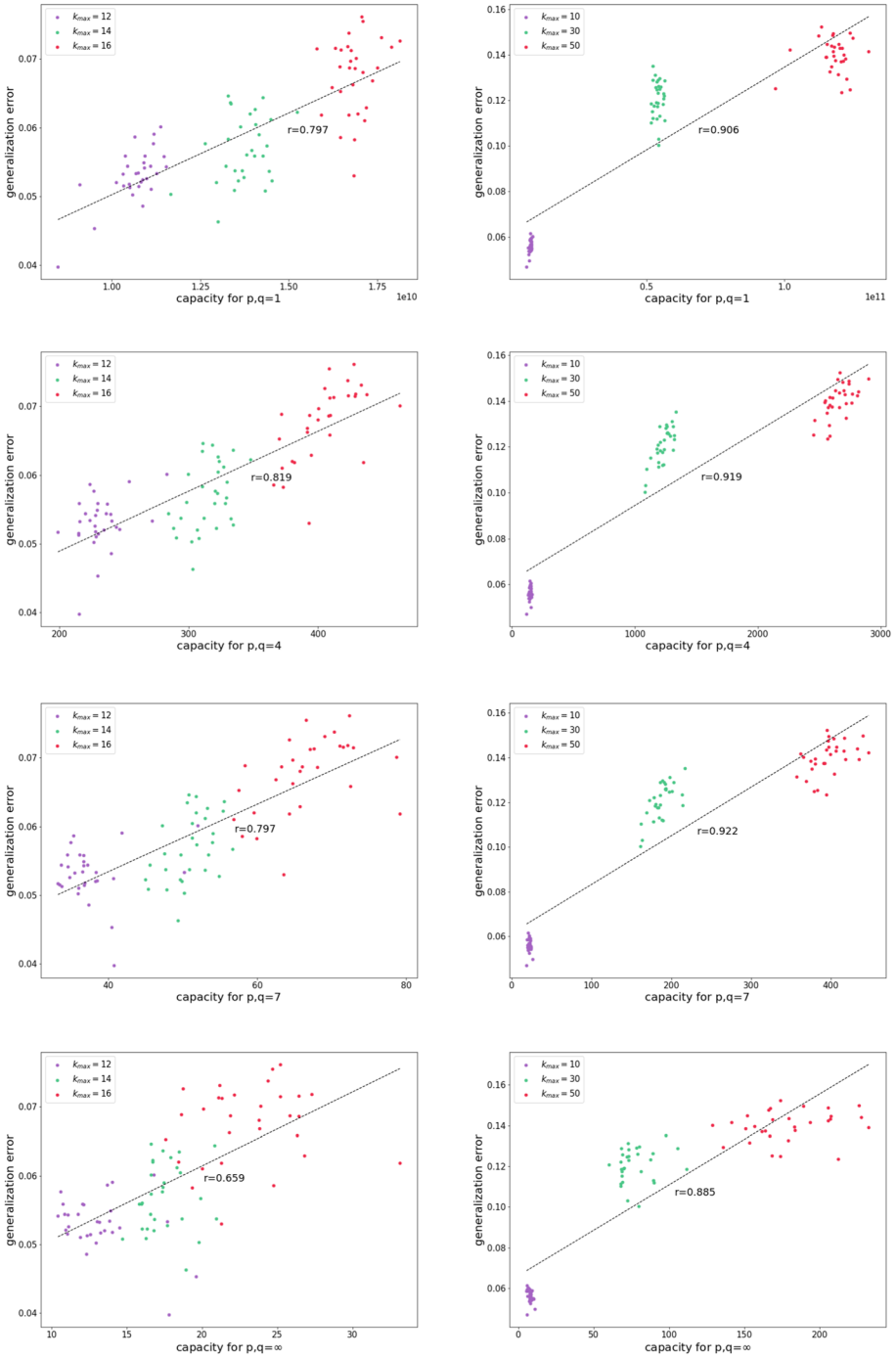

Fig. 4 *Left:* Scatter plot, correlation, and linear regression between generalization error and capacities of various p and q for 30 trained models for $k_{max} = 14, 16, 18$; *Right:* Scatter plot, correlation, and linear regression between generalization error and capacities of various p and q for 30 trained models for $k_{max} = 10, 30, 50$

Table 2 Correlation between empirical generalization error divided by weight norm of Fourier layers and $\sqrt[p*]{k_{max}}$ for FNO with depth 1 of Fourier layers

	$p=2$	$p=2.5$	$p=4$	$p=8$	$p=20$	$p=\infty$
$q=1$	0.6913	0.8199	0.8928	0.9062	0.8855	0.8647
$q=2$	0.7210	0.8386	0.9029	**0.9129**	0.8921	0.8699
$q=4$	0.7302	0.8389	0.8990	0.9064	0.8868	0.8629
$q=8$	0.7041	0.8133	0.8797	0.8872	0.8649	0.8328
$q=\infty$	0.6561	0.7620	0.8454	0.8573	0.8231	0.7741

Bold value indicates the highest one

Table 3 Correlation between empirical generalization error divided by weight norm of Fourier layers and $\sqrt[p*]{k_{max}}$ for FNO with depth 2 of Fourier layers

	$p=2$	$p=4$	$p=8$	$p=12$	$p=20$	$p=\infty$
$q=1$	−0.4145	0.8722	0.9319	0.9387	0.9385	0.9322
$q=2$	−0.4027	0.8882	0.9396	0.9439	0.9436	0.9365
$q=4$	−0.3386	0.9063	0.9484	0.9506	0.9485	0.9397
$q=8$	−0.1041	0.9129	**0.9508**	0.9504	0.9448	0.9319
$q=\infty$	0.3099	0.8821	0.9207	0.9162	0.9045	0.8834

Bold value indicates the highest one

is low for low capacities of p and q, and the desired dependency on k_{max} is unclear. As p and q increase, this correlation first increases and then decreases slightly. Based on these data, we can conclude that capacities with higher p and q contain more information about the model architecture(k_{max}), and capacities with very high p and q may cause a loss of specific information about each model; thus, the correlation decreases. Figure 5 shows the scatter plot and regression for a few experimental cases. The generalization error dependence on k_{max} was more convex at a depth of two. Based on our definition of capacity, the exponent of k_{max} is proportional to the depth of the Fourier layers. Therefore, the increased convexity illustrated on the right side of the figures qualitatively validates the results.

4.2.2 Empirical dependency on the size of training samples

As in Corollary 3 and other results on bounding the generalization error of neural nets by capacity norms, generalization error should converge as the training data set grows. The convergence rate is $O(n^{-l})$ where $0 \leq l \leq 0.5$. However, we found that the real convergence rate is much faster than the theoretical bounding rate. Moreover, our defined capacity is a suitable indicator by observing that the combination of our capacity with factor by training dataset size is highly correlated to the empirical generalization error. First, we show that we also obtain a high correlation for nonlinear activation other than ReLU. The architecture of the models is as follows: all the hyperparameters are the same as in Sect. 4.1, the weight decay, k_{max}, is randomly chosen from [0,e−3,...,4e−3], [10,12,...,20] respectively. The results are shown in Fig. 6.

Next, we demonstrate a high correlation between our empirically determined formula and the generalization errors of our experimental results. The architecture of this experiment was the same as that described above. However, the architecture of the model was fixed at $k_{max} = 14$ and the other hyperparameters were equal to those in the above experiments. The only varying parameter was the training dataset size [200, 400,..., 10,000].

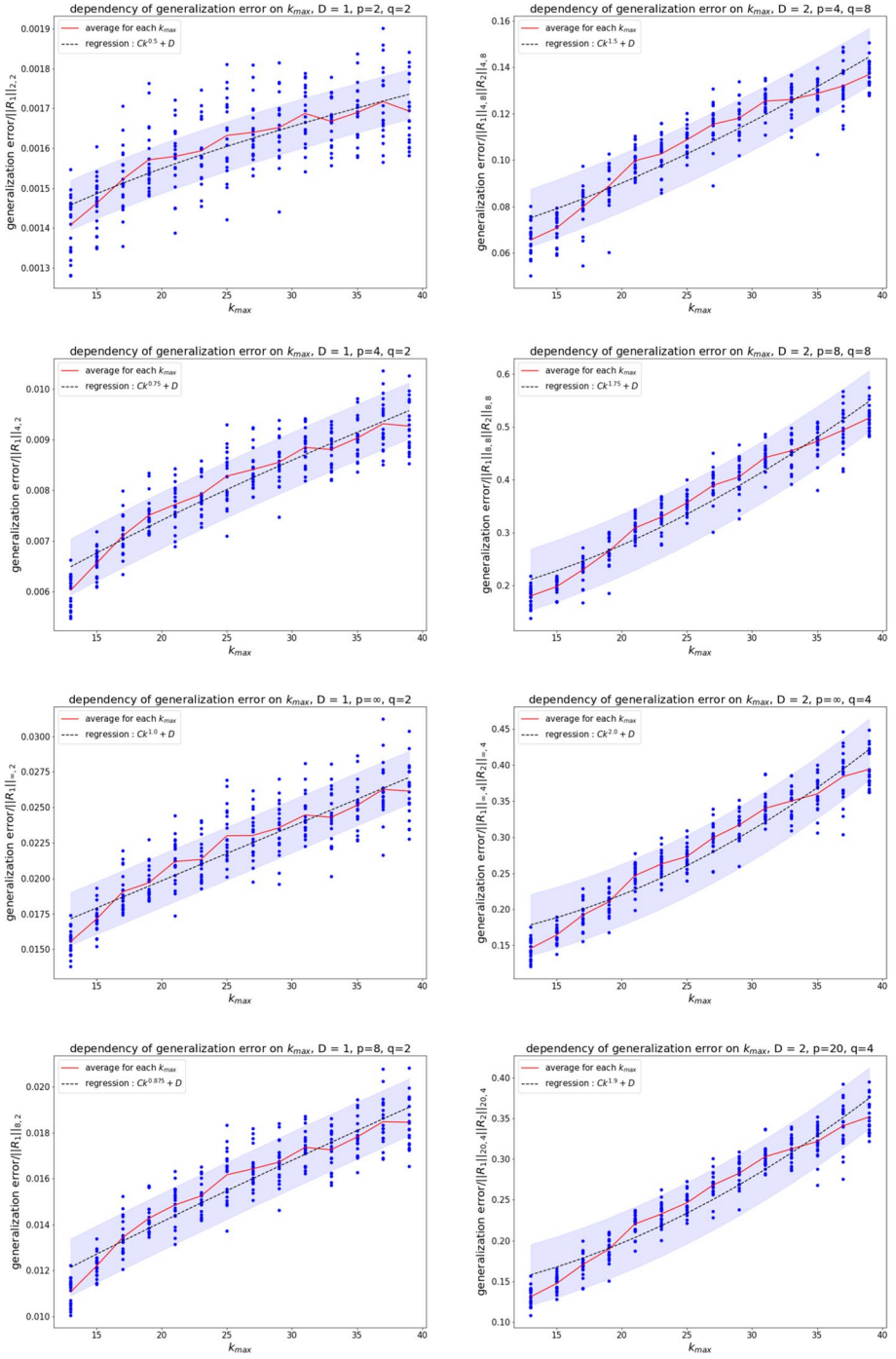

Fig. 5 *Left:* Scatter plot and regression between generalization error divided by norms of Fourier layers and $\sqrt[pq]{k_{max}}$ for various p, q where the depth of Fourier layer is 1; *Right:* Scatter plot and regression between generalization error divided by norms of Fourier layers and $\sqrt[\frac{pq}{2}]{k_{max}}$ for various p, q where the depth of Fourier layer is 2

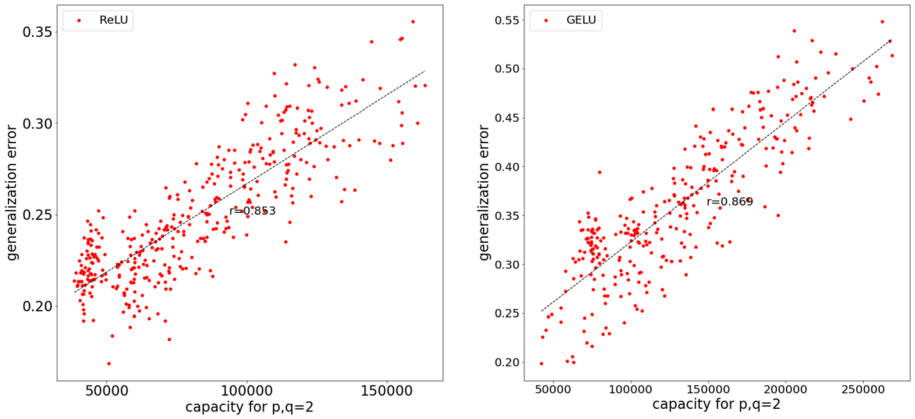

Fig. 6 *Left:* Scatter plot of generalization error for ReLU case *Right:* Scatter plot of generalization error for GELU case

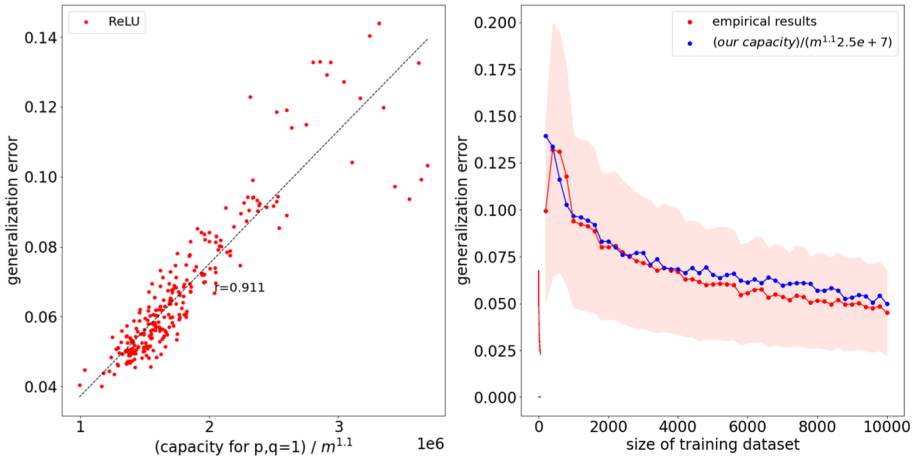

Fig. 7 *Left:* Scatter plot of generalization error for ReLU *Right:* Graph of empirical generalization error and normalized capacity factored by size of training dataset

We found that the actual convergence rates are much faster than the results of Corollary 3 (which implies that the upper bound of the generalization error is $O(m^{-0.5})$) and the convergence rates are dependent on the activation functions of the models; $O(m^{-1.1})$ for ReLU, and $O(m^{-1.55})$ for GELU. The results are presented in Figs. 7 and 8. The shaded areas in the right figures indicate the variance in the empirical generalization error. We also calculated the variance of our estimation; however, when normalized to the scale of empirical generalization error, it is significantly small compared to empirical generalization error. Therefore, it is a stable index for estimating generalization errors.

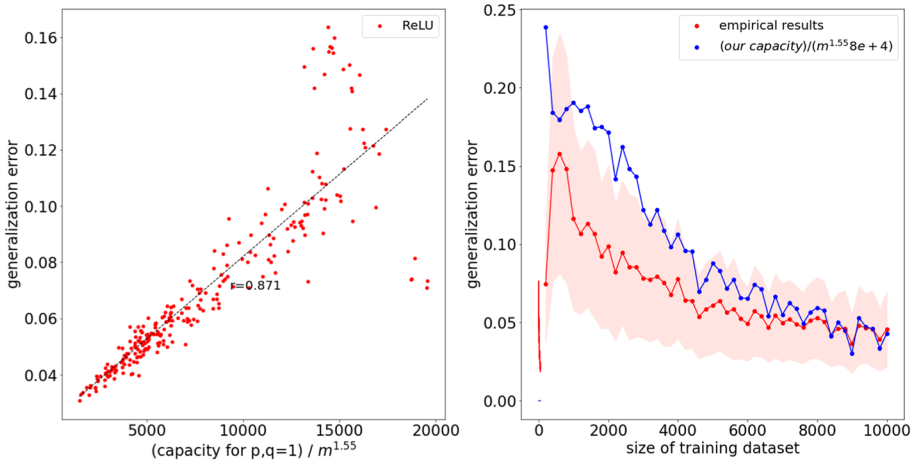

Fig. 8 *Left:* Scatter plot of generalization error for GELU case *Right:* Graph of empirical generalization error and normalized capacity factored by size of training dataset

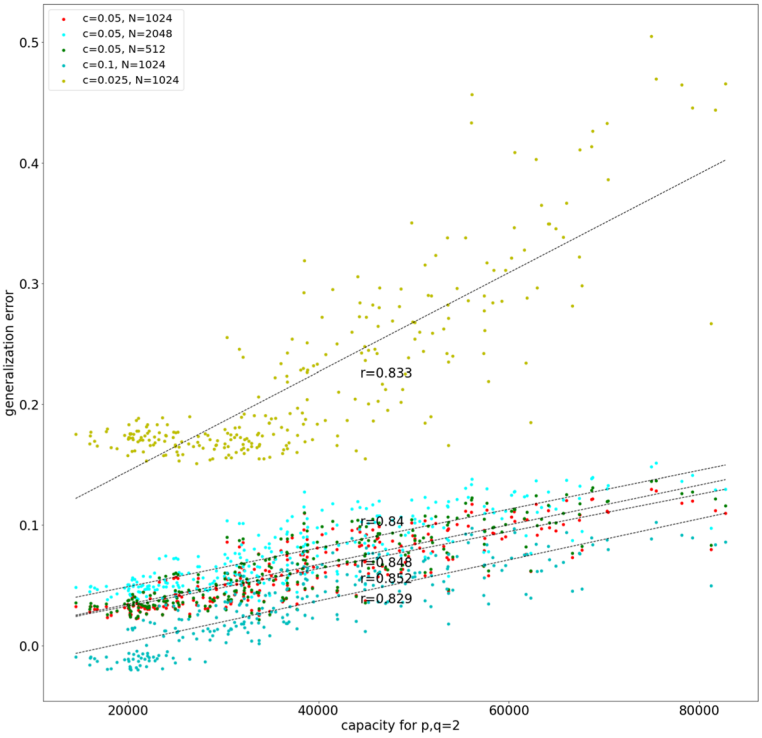

Fig. 9 Scatter plot and Correlation on the various test datasets

Table 4 Correlation between empirical generalization error and our capacity for various test datasets where c = 0.05, N = 1024 case is i.i.d to training dataset

(c, N)	(0.05, 1024)	(0.05, 2048)	(0.05, 512)	(0.1, 1024)	(0.025, 1024)
Correlation	0.852	0.840	0.848	0.829	0.833

4.3 Correlation for different resolution test samples and out of distribution samples

The upper bound of the generalization errors in Sect. 3 depends on discretization and dataset structure. To investigate how our capacity correlates with the generalization error of the test samples generated from different data distributions of the original training dataset, we constructed several test datasets other than the original one. For the experiment, we trained our models based on the same settings as those in Sect. 4.2.2, and inferenced on various test datasets. Recall that covariance of the GRF is $k(x, y) = e^{-\frac{(x-y)^2}{c^2}}$ where c is a coefficient: Our test datasets were all four cases: $c = 0.05$ with discretization $N = 512$ and $N = 2048$. and $N = 1024$ with $c = 0.025$ and $c = 0.1$. The capacity is calculated as $p = 2, q = 2$. The experimental results are shown in Fig. 9 and Table 4. It is interesting that although in Corollary 2 and 3, the upper bound has explicit dependence on discretization size N, in the empirical experiment, our capacity itself has a resolution-invariant property, showing almost the same tendency in the N=2048 and N=512 cases as in the original case. When $c = 0.1$, the tendencies of regression line and data are similar as in $c = 0.05$ case, and even empirical generalization errors are lower. However, when $c = 0.025$, the tendency of the data points was frustrated and had a higher generalization error. We assume that the main reason for this phenomenon is that the information on the frequency data distribution is different. For $c = 0.025$ case, each function pair has more high-frequency components. The $c = 0.1$ case has even fewer high-frequency components than $c = 0.05$ case.

4.4 Additional experiments on other PDEs

To show that our capacity is an effective indicator for estimating the generalization error, we experimented with more cases, which are problems of the governing equations 1-d integration, 1-d heat equation, 2-d heat equation and 2-d Navier–stokes equation. We verified the correlation between the empirical generalization error and the defined capacity. Similar to the experiments for the Burgers equation described in Sect. 4.2, We checked the correlation between empirical generalization error and capacity factored by the sizes of dataset for the fixed model architecture and varying sizes of dataset (Fig. 10). As discussed in Sect. 4.2, the GELU performs better than ReLU activation, and we fixed our activation as GELU throughout all experiments. We omitted the CNN layer to simplify the experiment.

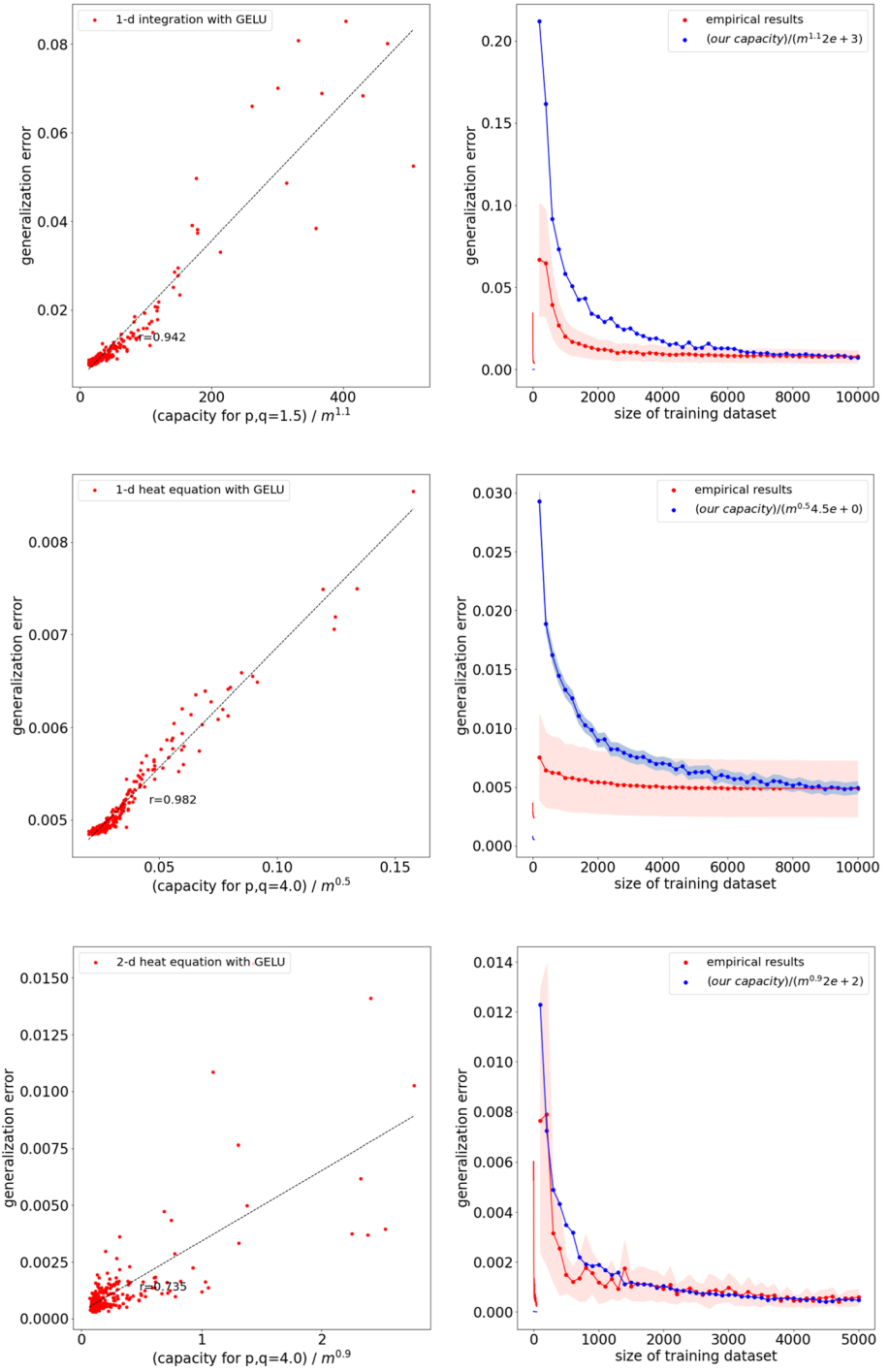

Fig. 10 *Left:* Scatter plot of generalization error for various PDE problems *Right:* Graph of empirical generalization error and normalized capacity factored by size of training dataset

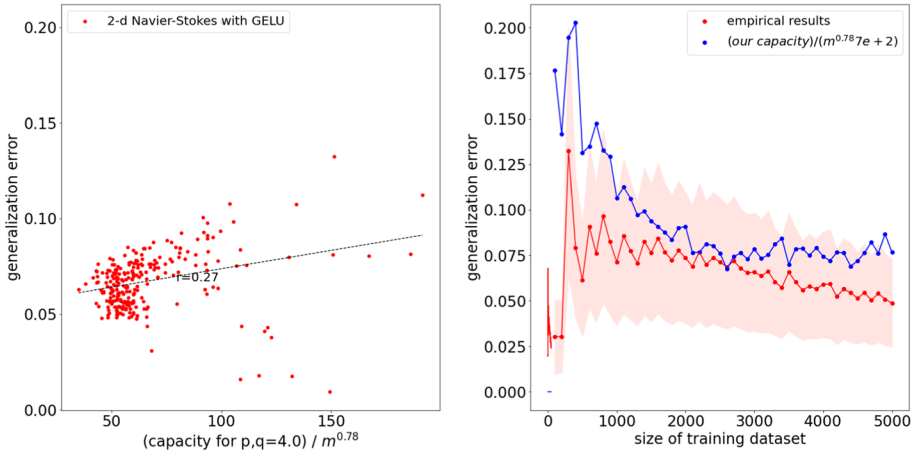

Fig. 10 (continued)

4.4.1 1-d integration

1-d integration is one of the simplest function operators. The experimental settings were as follows:

$$u_x = 10u_0,$$
$$u(0) = 0$$

where u_0 is an initial condition, and u is a scaled integration of this function. As in the Burgers equation setting in Sects. 4.1 and 4.2, our domain is a circle uniformly discretized by $N = 1024$. The function was scaled by multiplying it by 10 for normalization. Each initial function is generated from Gaussian random fields with covariance $k(x, y) = e^{-\frac{(x-y)^2}{(0.05)^2}}$, similar to the Burgers equation. The total training dataset comprises 10,000 pairs of functions, and test dataset comprises 200 pairs of functions, respectively.

Training settings For fixed hyperparameters and varying sizes of the training dataset, we fixed the model architecture as $k_{max} = 14$ and the width as 64, the depth of the Fourier layers as 2, and the weight decay as e−3. The size of training dataset is [200, 400,..., 10,000]. For each training dataset size, the training was repeated five times.

4.4.2 1-d heat equation

We experimented with another 1-d time dependent PDE problem known as 1-d heat equation. 1-d heat equation describes the heat distribution on a physical object such as a steel rod. The governing equations are as follows.

$$u_t = u_{xx}$$

The domain of the problem is a circle, and we uniformly discretize the domain by N = 1024. The initial condition was generated by the same Gaussian random field as that in the Burgers equation and 1-d integration settings. The target function is a section of the

solution of the heat equation for a given initial condition at T=0.1. The training and test datasets have 10,000 and 200 pairs of functions, respectively.

Training settings The training settings are the same as in 1-d integration problem.

4.4.3 2-d heat equation

Time-dependent heat equations are parabolic PDEs that describe heat diffusion. We experimented based on a 2-d time-dependent heat equation to verify that our capacity is effective for a linear 2-dimensional PDE. The governing equations are as follows:

$$u_t = \nabla^2 u$$

The domain of the problem is a torus, which means that the problem is periodic along the x- and y-axes. The domain was uniformly discretized with N=64 in both coordinates. The initial condition is generated by 2-d Gaussian random field with the distribution $\mu = \mathcal{N}(0, 7^{3/2}(-\Delta + 49I)^{-2.5})$ having the same GRF as in A.3.3 of Li et al. (2021). The target function is a section of the solution of the heat equation for a given initial condition at T = 0.005: The training and test datasets have 5000 and 200 pairs of functions, respectively.

Training settings For fixed hyperparameters and varying sizes of the training dataset, we fixed the model architecture as $(k_{max,1}, k_{max,2}) = (14, 14)$, with a width of 32, a depth of 2, and a weight decay of e−3. The training dataset size is [100, 200,..., 5000]. For each training dataset size, the training was repeated five times.

4.4.4 2-d Navier–Stokes equation

We consider the vicious, incompressible 2-d Navier–Stokes equation in vorticity form. The governing equations are as follows.

$$\begin{aligned} u_t + v \cdot \nabla u &= \nu \Delta u + f, \\ \Delta \cdot v &= 0, \\ u(x, 0) &= u_0 \end{aligned}$$

where u denotes the vorticity of velocity field v ($u = \nabla \times v$) defined in the torus-product time interval $(T \times [0, T])$. $0 \leq \nu$ is the viscosity coefficient, and f is a forcing function fixed as $f(x) = 0.1(sin(2\pi(x_1 + x_2)) + cos(2\pi(x_1 + x_2)))$. We selected one of the 2-d Navier–Stokes equation training dataset samples constructed in Li et al. (2021). Therefore, the initial vorticity function is generated from $\mu = \mathcal{N}(0, 7^{3/2}(-\Delta + 49I)^{-2.5})$ as in the heat equation. The viscosity coefficient ν is 1e−4. We set the target of our model as vorticity at time $T = 5$ (=$u(0, 5)$) for a given initial vorticity.

Training settings The training settings are the same as in the 2-d heat equation problem.

4.5 Comparison with other capacities

In this subsection, we compare our capacity with recently developed capacity norms: the Fisher-Rao norm (Liang et al., 2019), the Hessian trace norm (Petzka et al., 2021) and relative flatness (Petzka et al., 2021). For the experiment, we trained models with hyperparameters of width 16, 2-depth Fourier layers; k_{max} was chosen from [10,12,...,20] and weight decay from [0,1e−3,...,4e−3] on training dataset with 800 pairs of functions, which is the same as in the previous experiment settings. From Table 6 and Fig. 11, we infer that our capacity has the

Table 5 Time and memory cost for calculation of each norms

	Our capacity	Fisher–Rao norm	Hessian trace norm	Relative flatness
Time cost (s)	1e−3	0.15	175.12	188.86
Memory cost	80.1875 (KB)	80.1875 (KB)	200 (MB)	200 (MB)

Table 6 Correlation between empirical generalization error and various norms for different test datasets

	Our capacity	Fisher-Rao norm	Hessian trace norm	Relative flatness
c = 0.05, N = 1024	0.921	0.195	0.046	0.182
c = 0.05, N = 2048	0.956	0.257	0.085	0.243
c = 0.05, N = 512	0.936	0.135	0.065	0.222
c = 0.1, N = 1024	0.842	0.168	0.164	0.281
c = 0.025, N = 1024	0.698	0.197	0.017	0.141

highest correlation among the norms because our capacity does contain information about the hyperparameter concerning the model's architecture, whereas other norms do not. In addition, complicated derivatives or second-derivative information are not required to calculate our capacity. Therefore, as shown in Table 5, the calculation time of our capacity is much shorter than that of the other norms, enhancing memory efficiency.

5 Conclusion

We investigated the bounding Rademacher complexity of an FNO and defined its capacity, which depends on the model architecture and the group norms of the weights. Although several results exist regarding the bounding Rademacher complexity of various types of neural networks, the FNO possesses tensor weights of rank higher than two. Therefore, our study may be useful for other NNs containing higher-rank tensors. Our results are experimentally validated. Based on these experiments, we gained insights into the impact of p and q values and information about the model weights and architecture stored in terms of capacities. Through experiments on other PDEs and their dependency on the architecture and size of datasets, we validated that our capacity norm is effective for estimating the empirical generalization error. By comparing it with other sophisticated capacity norms, we empirically prove that our capacity norm is an efficient and effective index among these norms. Moreover, although various neural operators have been developed, including FNO and DeepONet, the analysis of PAC learning for these neural operators has not been performed in detail. Thus, this study may serve as a guide for such analyses. In this study, we assume that the activation function is fixed. For a general model containing parameterized activation, such as PReLU, we need to modify our analysis. Although the Rademacher complexity contains information about datasets, the bounding of our results lacks a specific dependency on each problem. Because we experimented with various PDE problems, the performance of the FNO varied for each problem. Therefore, we must extend the complexities to include information about datasets. In addition, although we empirically verified faster convergence compared to the theoretical

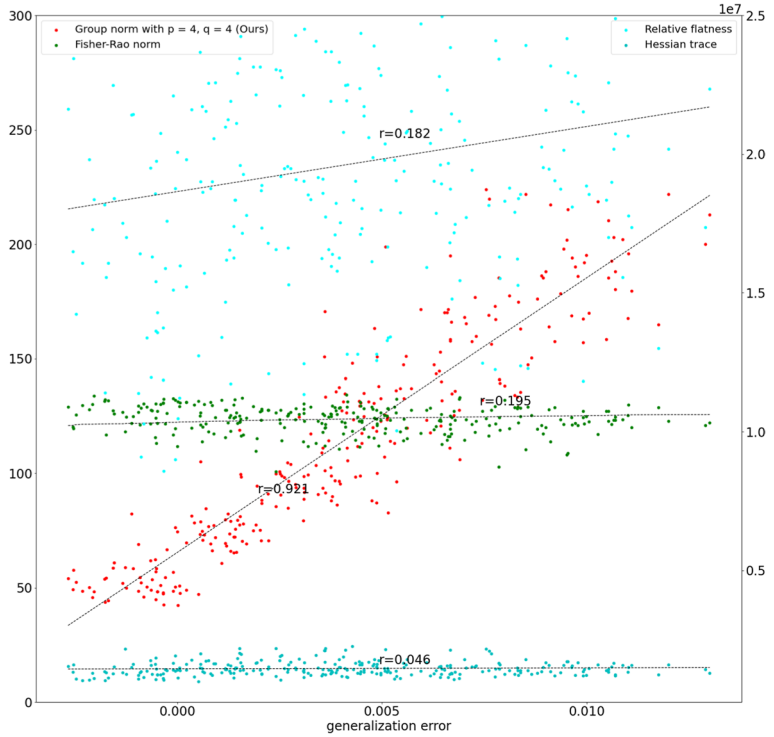

Fig. 11 Scatter plot and correlation for between empirical generalization error and various norms

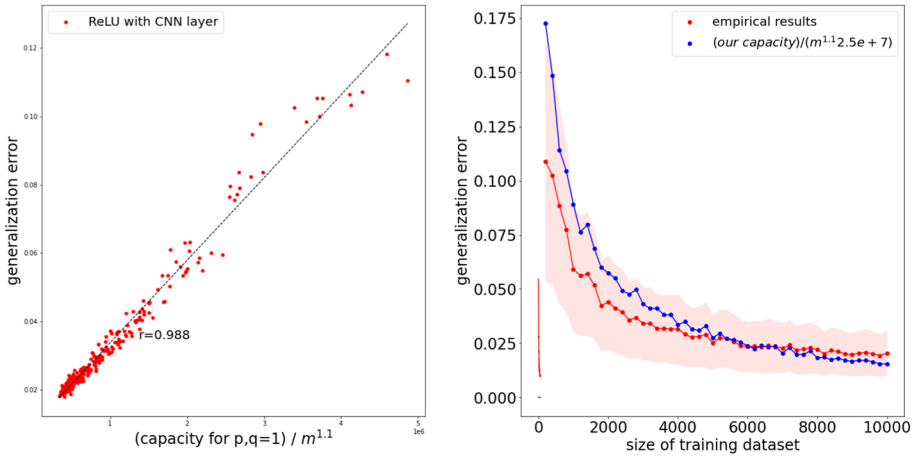

Fig. 12 *Left:* Scatter plot of generalization error for models with CNN layers and ReLU activation. *Right:* Graph of empirical generalization error and normalized capacity factored by the size of training dataset

bound, we need to justify the empirical fast convergence of the generalization error through careful theoretical analysis.

Appendix A: Experimental details

A.1 Environment of experiments

All the experiments were conducted using Pytorch 2.0.1, with Python 3.9.6. The specifications of the hardware environment are as follows (Table 7):

A.2 Numerical schemes used in experiments

We used a forward difference scheme with a time step of 1e−4 and 1e−5 for 1-d and 2-d heat equations, respectively. For the 1-d integration experiment, we used the most basic method (right-hand rule) by adding all terms and multiplying by the interval spacing. For 2-d Navier–Stokes equation, we used the dataset in Li et al. (2021); the detailed numerical scheme is specified in Li et al. (2021). The equation was solved using a stream function formulation and a pseudospectral method. For time marching, the Crank-Nicolson update was used with a time step of 1e−4.

A.3 Mathematical formulas for norms

We calculated the Fisher-Rao norm using the following formula with the finite difference method derived in Liang et al. (2019):

$$\|\theta\|_{fr}^2 = \mathbb{E}\left\langle \frac{\partial l(f_\theta(X), Y)}{\partial f_\theta(X)}, f_\theta(X) \right\rangle$$

For the Hessian trace, we apply the following formula to the first Fourier layer:

$$H_{(k,s',s),(\tilde{k},\tilde{s}',\tilde{s})} = \frac{\partial^2 l}{\partial w_{(k,s',s)} \partial w_{(\tilde{k},\tilde{s}',\tilde{s})}}$$
$$tr(H) = \sum_{all(k,s',s)} H_{(k,s',s),(k,s',s)}$$

Finally, for relative flatness, because the weight is a multi-rank tensor, we slightly modified the formula in Petzka et al. (2021). This formula is also applied to the first Fourier layer. The formula is as follows:

$$H_{(k,s',s),(\tilde{k},\tilde{s}',\tilde{s})} = \frac{\partial^2 l}{\partial w_{(k,s',s)} \partial w_{(\tilde{k},\tilde{s}',\tilde{s})}}$$
$$\tilde{tr}(H_{(s,s')}) = \sum_{all(k,k',\tilde{s},\tilde{s}')} H_{(k,\tilde{s},s),(k',\tilde{s}',s')}$$
$$\kappa_{Tr} := \sum_{all(s,s')} \langle R_{\cdot,\cdot,s}, R_{\cdot,\cdot,s'} \rangle \tilde{tr}(H_{(s,s')})$$

Table 7 Specification of computer hardware

CPU	GPU	RAM
Intel i9-10900	Nvidia RTX3080	64GB

Appendix B: FNO with CNN layer versus without CNN layer

In this section, we present the results of the CNN-layer case. Although, for simplicity of analysis, we dropped the CNN layers in the experiments in Sect. 4, as shown in Fig. 12, the correlation is even higher than the cases with CNN layers dropped.

Acknowledgements This work was supported by the Challengeable Future Defense Technology Research and Development Program through the Agency for Defense DevelopmentADD, funded by the Defense Acquisition Program Administration in 2021 (No. 915020201), and NRF Grant [2021R1A2C3010887].

Author contributions TK devised the approach, developed the mathematical proofs, coded, and conducted the experiments. MK corrected the overall manuscript and acquired funding for the project.

Funding Open Access funding enabled and organized by Seoul National University.

Availability of data and materials and codes The codes used to generate dataset used in this study and conduct experiments, including modeling models, are available from TK upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Awasthi, P., Frank, N., & Mohri, M. (2020). On the rademacher complexity of linear hypothesis sets. arXiv arXiv:2007.11045

Bartlett, P. L., Foster, D. J. & Telgarsky, M. (2021). Spectrally-normalized margin bounds for neural networks. *Advances in Neural Information Processing Systems*

Cai, Z., Chen, J., & Liu, M. (2021). Least-squares ReLU neural network (LSNN) method for linear advection-reaction equation. *Journal of Computational Physics, 443*, 686–707.

Gopalani, P., Karmakar, S. & Mukherjee, A. (2022). Capacity bounds for the deeponet method of solving differential equations. arXiv arXiv:2205.11359

Gupta, G., Xiao, X., & Bogdan, P. (2021). Multiwavelet-based operator learning for differential equations. *Advances in Neural Information Processing Systems, 34*, 24048–24062.

Hao, C., Zhanfeng, M., Zhouwang, Y. & Xiao, W. (2019). Theoretical investigation of generalization bound for residual networks. In *International joint conferences on artificial intelligence organization*, pp. 2081–2087

Jakubovitz, D., Giryes, R., Rodrigues, M. R. D. (2019). Generalization Error in Deep Learning. Birkhuser Cham, GEWERBESTRASSE 11, Cham, Switzerland.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A. & Anandkumar, A. (2021). Neural operator: Learning maps between function spaces. arXiv arXiv:2108.08481

Kovachki, N., Lanthaler, S., & Mishra, S. (2021). On universal approximation and error bounds for Fourier neural operators. *Journal of Machine Learning Research, 22*(290), 1–76.
Lei, Y., Dogan, U., Zhou, D., & Kloft, M. (2019). Data-dependent generalization bounds for multi-class classification. *IEEE Transactions on Information Theory, 65*(5), 2995–3021.
Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. & Anandkumar, A. (2020). Neural operator: Graph kernel network for partial differential equations. ICLR 2020 Workshop ODE/PDE+DL
Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A. & Anandkumar, A. (2021) Fourier neural operator for parametric partial differential equations. ICLR 2021.
Liang, T., Poggio, T., Rakhlin, A., & Stokes, J. (2019). Fisher-rao metric, geometry, and complexity of neural networks. *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics, 89*, 888–896.
Long, P. M. & Sedghi, H. (2020). Generalization bounds for deep convolutional neural networks. ICLR 2020.
Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence, 3*(3), 218–229.
Lv, S. (2021). Generalization bounds for graph convolutional neural networks via rademacher complexity. arXiv arXiv:2102.10234
Maurer, A. (2016). A vector-contraction inequality for rademacher complexities. arXiv arXiv:1605.00251
Minshuo, C., Xingguo, L., & Tuo, Z. (2020). On generalization bounds of a family of recurrent neural networks. *Proceedings of Machine Learning Research, 108*, 1233–1243.
Neyshabur, B., Tomioka, R., & Srebro, N. (2015). Norm-based capacity control in neural networks. *Proceedings of Machine Learning Research, 40*, 1376–1401.
Pathak, J., subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., Hassanzadeh, P., Kashinath, K. & Anandkumar, A. (2022). Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. arXiv arXiv:2202.11214
Petzka, H., Kamp, M., Adilova, L., Sminchisescu, C., & Boley, M. (2021). Relative flatness and generalization. *Advances in Neural Information Processing Systems, 34*, 18420–18432.
Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics, 378*, 686–707.
Seleznova, M., & Kutyniok, G. (2021). Analyzing finite neural networks: Can we trust neural tangent kernel theory? *Proceedings of Machine Learning Research, 145*, 847–867.
Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.
Sontag, E. D. (1998). Vc dimension of neural networks. *NATO ASI Series F Computer and Systems Sciences, 168*, 69–96.
Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM, 27*(11), 1134–1142.
Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks, 10*(5), 988–999.
Weinan, E., Chao, M., & Qingcan, W. (2020). Rademacher complexity and the generalization error of residual networks. *Communications in Mathematical Sciences, 18*(6), 1755–1774.
Weinan, E., & Yu, B. (2018). The deep ritz method: A deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics, 6*(1), 1–12.
Wen, G., Li, Z., Azizzadenesheli, K., Anandkumar, A., & Benson, S. M. (2022). U-fno-an enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources, 163*, 104180.