MULTISCALE MODEL. SIMUL.
Vol. 21, No. 4, pp. 1457–1485

NEURAL NETWORK APPROXIMATION OF COARSE-SCALE SURROGATES IN NUMERICAL HOMOGENIZATION*

FABIAN KRÖPFL†, ROLAND MAIER‡, AND DANIEL PETERSEIM§

Abstract. Coarse-scale surrogate models in the context of numerical homogenization of linear elliptic problems with arbitrary rough diffusion coefficients rely on the efficient solution of fine-scale subproblems on local subdomains whose solutions are then employed to deduce appropriate coarse contributions to the surrogate model. However, in the absence of periodicity and scale separation, the reliability of such models requires the local subdomains to cover the whole domain which may result in high offline costs, in particular for parameter-dependent and stochastic problems. This paper justifies the use of neural networks for the approximation of coarse-scale surrogate models by analyzing their approximation properties. For a prototypical and representative numerical homogenization technique, the Localized Orthogonal Decomposition method, we show that one single neural network is sufficient to approximate the coarse contributions of all occurring coefficient-dependent local subproblems for a nontrivial class of diffusion coefficients up to arbitrary accuracy. We present rigorous upper bounds on the depth and number of nonzero parameters for such a network to achieve a given accuracy. Further, we analyze the overall error of the resulting neural network enhanced numerical homogenization surrogate model.

Key words. deep learning, neural networks, surrogate models, numerical homogenization, approximation properties

MSC codes. 68T07, 65N30, 35J15, 35A35

DOI. 10.1137/22M1524278

1. Introduction.

Surrogate models play an important role in the context of partial differential equations (PDEs) with multiscale features. After the offline adaptation to a particular coefficient, these models provide a reliable and online efficient approximation of the solution operator on some coarse scale of interest. In the context of modern numerical homogenization, the adaptation of the models is typically realized through coefficient-specific approximation spaces with provably optimal approximation on the coarse target scale. Prominent examples are methods such as the Localized Orthogonal Decomposition (LOD) [MP14, HP13, MP20], gamblets [Owh17], rough polyharmonic splines [OZB14] or generalized (multiscale) finite element methods [BL11, EGW11]. For a more detailed overview of such methods, we refer to [AHP21] and the references therein. Such numerical homogenization methods have been successful for a wide range of PDEs and their applications because they do not rely on structural assumptions on the coefficient such as periodicity or scale separation. Another benefit of numerical homogenization approaches is their construction, which is similar to classical finite element methods. In particular, they are based on

*Received by the editors September 23, 2022; accepted for publication (in revised form) July 26, 2023; published electronically October 20, 2023.
https://doi.org/10.1137/22M1524278
Funding: The work of the first and third authors was supported by the Deutsche Forschungsgemeinschaft within the research project Autonomous Research for Exploring Structure-property Linkages and Optimizing Microstructures (PE 2143/7-1).
†Institute of Mathematics, University of Augsburg, 86159, Augsburg, Germany (fabian.kroepfl@uni-a.de).
‡Institute for Applied and Numerical Mathematics, Karlsruhe Institute of Technology, 76131, Karlsruhe, Germany (roland.maier@kit.edu).
§Institute of Mathematics & Centre for Advanced Analytics and Predictive Sciences (CAAPS), University of Augsburg, 86159, Augsburg, Germany (daniel.peterseim@uni-a.de).

coarse-scale submatrices that are combined to a coarse global system matrix by appropriate local-to-global mappings. However, the local submatrices are based on the sufficiently resolved solution of fine-scale subproblems on local subdomains (which are also known as *cell problems* or *corrector problems* in the homogenization literature) that resolve all features of the local PDE response. In the absence of periodicity and scale separation, the reliability of the resulting surrogate relies on the consideration of the coefficient in the full domain, i.e., the union of all local subdomains needs to cover the whole domain, which may result in high offline cost, in particular for parameter-dependent or stochastic coefficients.

To reduce the complexity bottleneck in the process of building reliable surrogate models, there has recently been a growing interest in utilizing data-driven approaches such as deep learning in the homogenization community; see, e.g., [ABS+20, PZ21, CLPZ21, WHG+21, CE18, SSM23, HL22, LLZ22]. The previous work [KMP22] proposes to replace the computation of all local system matrices—that can each be seen as the output of a mapping from a local extract of the underlying PDE coefficient to the corresponding coarse local matrix—by one single neural network in the offline phase of numerical homogenization. The approximation by a local and thus relatively small (trained) neural network allows one to reduce the high complexity of the corrector computations to a number of forward passes through the network, which is of particular value in nonstationary, parametric, or nondeterministic problems, since the computational savings during the computation of the local matrices eventually outweigh the initial effort required to train the network. Numerical experiments in [KMP22] showed that a reasonably sized network leads to good approximation properties compared to the surrogate based on the classical computation of local subproblems on a fine scale.

In this work, we aim at a mathematical foundation of this neural network enhanced numerical homogenization approach in the context of a prototypical elliptic model problem with an underlying (possibly fine-scale) coefficient. We restrict ourselves to the LOD which will be reviewed briefly in section 2 below. Due to its representative nature outlined in [AHP21], we believe that the derivations can be generalized to other variants of classical and modern numerical homogenization methods. From a practical point of view, the coarse-scale surrogate model computed by the LOD can be represented by a global coarse-scale system matrix that is assembled from local matrices that depend on local instances of the underlying coefficient. We investigate the approximation properties of neural networks as a replacement of the mappings from local coefficient to local matrix in section 3.

Our analysis builds upon the formal framework for studying the approximation properties of deep neural networks developed in [PV18]. The approach is based on combining smaller neural networks as building blocks to approximate complicated nonlinear functions and has so far been used in a multitude of settings and applications. Examples include the approximation of higher-order finite elements [OPS20], Kolmogorov PDEs in the context of option pricing [EGJS21], and reduced basis methods in the context of parametric PDEs [KPRS21]. For an overview on the expressivity of neural networks, we refer to the survey article [GRK20].

Based on these recent findings, the main result of this paper in section 3 provides an upper bound on depth and number of parameters of an appropriate neural network which is able to achieve a certain tolerance error when approximating the local coefficient-to-matrix mappings that are the basis of the LOD. In section 4, we then investigate the error between discrete solutions obtained with the original LOD approach and discrete solutions that rely on a surrogate model that is built from the

local neural network approximations. In particular, we show that there exists a network such that the error of the two discrete solutions is at most of size $\mathcal{O}(H)$, where H denotes the coarse scale of interest. Note that this bound is of the same error as the discretization error of the LOD with respect to the exact solution to the elliptic problem. Finally, we comment on the choice of fine-scale parameters for the computation and approximation of local contributions (section 5) and draw conclusions.

Notation. Throughout this work, $C, c > 0$ denote generic constants that are independent of the scales H, ε, h, but might depend on the physical dimension $d \leq 3$. Further, we write $\theta \lesssim \eta$ if $\theta \leq C\eta$ and $\theta \approx \eta$ if additionally also $\theta \geq c\eta$. Matrices and vectors are denoted in boldface notation throughout this paper. In particular, we write $\mathbf{Id}_n$ for the $n \times n$ identity matrix and $\mathbf{0}_n$ for the zero vector in $\mathbb{R}^n$.

2. Operator compression with deep neural networks.

In this section, we briefly explain the LOD methodology and summarize the main ideas and results of [KMP22], which provides the basis for the theoretical analysis in the subsequent sections. The aim of this section is not to provide a detailed analysis of the method, but to give an overview of the underlying concepts.

2.1. Setting.

Let $d \in \{1, 2, 3\}$, and let $D \subseteq \mathbb{R}^d$ be a bounded, polyhedral, convex Lipschitz domain. We denote with $H_0^1(D)$ the standard Sobolev space of L^2-functions whose traces vanish on ∂D and that have weak first derivatives in $L^2(D)$. Given a function $f \in L^2(D)$ and a scalar coefficient $A \in L^\infty(D)$, consider the prototypical variational problem of finding $u \in H_0^1(D)$ such that

$$a(u, v) := \int_D A\nabla u \cdot \nabla v \,\mathrm{d}x = \int_D fv \,\mathrm{d}x \quad \text{for all } v \in H_0^1(D). \tag{2.1}$$

Note that we do not pose any requirements on the coefficient A apart from the existence of uniform bounds $\alpha, \beta \in \mathbb{R}$ such that

$$0 < \alpha \leq A(x) \leq \beta < \infty \quad \text{for a.e. } x \in D. \tag{2.2}$$

In particular, we do not assume periodicity or scale separation and allow for arbitrarily rough coefficients that may vary on a continuum of scales up to some fine microscale ε. In this setting, the Lax–Milgram theorem guarantees the existence of a unique solution $u \in H_0^1(D)$ to problem (2.1). Note that a and u implicitly depend on A.

2.2. Coarse-scale discretization by localized orthogonal decomposition.

Although the coefficient A encodes microscopic oscillations on a scale ε, in practical simulations one is often interested in the effective behavior of the solution u on some macroscopic target scale of interest $H \geq \varepsilon$. It is well known, however, that the discretization of (2.1) with standard approaches such as conforming finite elements leads to acceptable results only if the problem is discretized on a scale smaller than ε that fully resolves all fine-scale oscillations of the coefficient; see, e.g., the illustrations in [MP20, Chap. 2].

In order to overcome this problem, numerous methods have been proposed that compress the fine-scale information contained in the coefficient A to a surrogate model $\mathfrak{S}_A$ (represented by a system matrix $\mathbf{S}_A$) capable of reproducing the effective behavior of u on the target scale H. One such method known as Localized Orthogonal Decomposition (LOD) achieves this by explicitly constructing a coarse approximation space spanned by coefficient-adapted basis functions, which can then be used to approximate (2.1) in a Galerkin fashion, resulting in a sparse system matrix $\mathbf{S}_A$ that is composed of multiple local contributions.

The LOD has originally been introduced in an elliptic setting [MP14, HP13], but has been successfully applied in connection with, e.g., wave propagation problems [GP15, Pet17, AH17, PS17, GHV18, MP19, RHB19, GM21, LMP21, MV22, HW22] and parabolic PDEs [MP18, LMM22]. The idea has also been extended to higher-order [Mai21] and multiresolution [HP22] variants based on the hierarchical approach known as gamblets [Owh17]. Further, the possibility to superlocalize the local subproblems has been investigated [HP21].

In the following, we briefly describe the method and an adapted version based on local neural network approximations as presented in [KMP22]. Note that the method in [KMP22] may be applied to general linear divergence form partial differential equations in a straightforward way and works beyond the LOD framework, but we restrict ourselves to its application in connection with a prototypical elliptic setting and the LOD method within this work.

Let $\mathcal{T}_H$ be a uniform Cartesian mesh of D with characteristic mesh size H, let $Q^1(\mathcal{T}_H)$ be the standard first-order finite element space of piecewise polynomials with coordinate degree at most 1 and consider the conforming finite-dimensional space $V_H := Q^1(\mathcal{T}_H) \cap H^1_0(D)$. In our setting, H denotes the length of an edge of an element of $\mathcal{T}_H$. Besides the coarse mesh $\mathcal{T}_H$, we will also need a fine mesh $\mathcal{T}_h$ with $h < \varepsilon$ and the corresponding discrete space $V_h \supset V_H$, as well as an intermediate mesh $\mathcal{T}_\varepsilon$ with $h < \varepsilon < H$. Moreover, we assume these meshes to be nested and uniform refinements of each other, i.e., that $\mathcal{T}_\varepsilon$ is a uniform refinement of $\mathcal{T}_H$ and $\mathcal{T}_h$ a uniform refinement of $\mathcal{T}_\varepsilon$. An important ingredient of the method is a projective quasi-interpolation operator $\mathcal{I}_H \colon H^1_0(D) \to V_H$, which fulfills

$$\|H^{-1}(v - \mathcal{I}_H v)\|_{L^2(T)} + \|\nabla \mathcal{I}_H v\|_{L^2(T)} \leq C \|\nabla v\|_{L^2(\mathtt{N}(T))} \tag{2.3}$$

for all $v \in H^1_0(D)$ and any element $T \in \mathcal{T}_H$, where the constant C is independent of H, and $\mathtt{N}(S) := \mathtt{N}^1(S)$ is an element patch around $S \subseteq D$ defined by

$$\mathtt{N}^1(S) := \bigcup \big\{ \overline{K} \in \mathcal{T}_H \,|\, \overline{S} \cap \overline{K} \neq \emptyset \big\}.$$

A prominent example of an operator $\mathcal{I}_H$ that fulfills (2.3) is the one considered in [EG17], which is also used for the numerical experiments in [KMP22]. For that reason we restrict our analysis in section 3 to this particular choice. It is defined by $\mathcal{I}_H := E_H \circ \Pi_H$, where Π_H denotes the piecewise L^2-projection onto $Q_1(\mathcal{T}_H)$. For any $v_H \in Q_1(\mathcal{T}_H)$, the operator E_H averages in any inner node z of $\mathcal{T}_H$ the values of the neighboring elements, i.e.,

$$\big(E_H(v_H)\big)(z) := \sum_{\substack{K \in \mathcal{T}_H : \\ z \in \overline{K}}} \big(v_H|_{\overline{K}}\big)(z) \cdot \frac{1}{\operatorname{card}\{K' \in \mathcal{T}_H : \, z \in \overline{K'}\}}. \tag{2.4}$$

Further, $\big(E_H(v_H)\big)(z) = 0$ if $z \in \partial D$. Given $\mathcal{I}_H$, we define the so-called *fine-scale space* as

$$\mathcal{W} := \ker \mathcal{I}_H|_{V_h},$$

which contains functions that cannot be captured by the space V_H. For any $S \subseteq D$, we also define a local version of $\mathcal{W}$,

$$\mathcal{W}(S) := \{ w \in \mathcal{W} \,|\, \operatorname{supp}(w) \subseteq S \}.$$

Appropriate local versions of the fine-scale space are now used to *correct* coarse-scale functions in a coefficient-dependent manner. Therefore, we iteratively define an enlarged element patch $\mathtt{N}^\ell(S) := \mathtt{N}(\mathtt{N}^{\ell-1}(S))$, $\ell \geq 2$. Let a fixed $A \in L^\infty(D)$ which fulfills (2.2) be given. For an $\ell \in \mathbb{N}$ (that will later be quantified) as well as a function $v_H \in V_H$ and $T \in \mathcal{T}_H$, we now introduce the *element corrector* $\mathcal{Q}_T^\ell v_H \in \mathcal{W}(\mathtt{N}^\ell(T))$ as the solution to

$$a(\mathcal{Q}_T^\ell v_H, w) = \int_T A\nabla v_H \cdot \nabla w \,\mathrm{d}x \quad \text{for all } w \in \mathcal{W}(\mathtt{N}^\ell(T)). \tag{2.5}$$

Note that this is a discrete subproblem, which is locally computed on the fine scale h. The corresponding algebraic realization of this subproblem will be discussed in section 2.3 below and assumptions on the fine mesh are discussed in section 5. Note that the correctors $\mathcal{Q}_T^\ell v_H$ for different T are independent of each other and their respective supports are limited to $\mathtt{N}^\ell(T)$. That is, global computations on a fine scale can be avoided. Further, the correctors are only computed for a basis of V_H due to linearity.

The last step towards the final multiscale method consists of defining an appropriate space based on the element correctors. Therefore, we first define the *global correction operator* $\mathcal{Q}^\ell \colon V_H \to \mathcal{W}$ by

$$\mathcal{Q}^\ell := \sum_{T \in \mathcal{T}_H} \mathcal{Q}_T^\ell, \tag{2.6}$$

which will be used to correct classical finite element functions. In particular, the *classical LOD approximation (C-LOD)* now seeks $u_H^{\mathrm{c}} \in V_H$ such that

$$a((\mathsf{id} - \mathcal{Q}^\ell)u_H^{\mathrm{c}}, (\mathsf{id} - \mathcal{Q}^\ell)v_H) = \int_D f\, v_H \,\mathrm{d}x \quad \text{for all } v_H \in V_H. \tag{2.7}$$

Alternatively, u can be approximated with a Petrov–Galerkin variant of the classical LOD. That is, the correction is only used for the trial functions but not for the test functions. The *Petrov–Galerkin LOD approximation (PG-LOD)* $u_H^{\mathrm{pg}} \in V_H$ solves

$$a((\mathsf{id} - \mathcal{Q}^\ell)u_H^{\mathrm{pg}}, v_H) = \int_D f\, v_H \,\mathrm{d}x \quad \text{for all } v_H \in V_H. \tag{2.8}$$

The Petrov–Galerkin variant has several computational advantages compared to the classical method and was therefore used for the experiments in [KMP22].

From a theoretical point of view, the approximations u_H^{c} and u_H^{pg} are both first-order accurate in $L^2(D)$ if the oversampling parameter ℓ is chosen logarithmically in the target mesh size H, i.e., $\ell \approx |\log(H)|$, and the fine mesh parameter h is chosen small enough (cf. section 5 below for details). More precisely, it holds that

$$\|u - u_H^{\mathrm{c}}\|_{L^2(D)} \lesssim \|u - u_h\|_{L^2(D)} + (H + e^{-c_{\mathrm{dec}}\ell})\,\|f\|_{L^2(D)}, \tag{2.9}$$

as well as

$$\|u - u_H^{\mathrm{pg}}\|_{L^2(D)} \lesssim \|u - u_h\|_{L^2(D)} + (H + e^{-c_{\mathrm{dec}}\ell})\,\|f\|_{L^2(D)}, \tag{2.10}$$

where $u_h \in V_h$ is the classical Galerkin finite element approximation of u in the space V_h. These results follow from the triangle inequality and the results presented, e.g., in [GP17, CMP20, MP20]. The last two estimates arise from a localization estimate of the form

$$\|\nabla(\mathcal{Q} - \mathcal{Q}^\ell)v_H\|_{L^2(D)} \leq Ce^{-c_{\mathrm{loc}}\ell}\,\|\nabla v_H\|_{L^2(D)}, \quad v_H \in V_H \tag{2.11}$$

with a constant c_{loc} which is independent of h, H, and ℓ. Note that $\mathcal{Q} := \mathcal{Q}^\infty$ is computed based on nonlocalized subproblems (2.5). This result is, for instance, shown in [HP13, Pet16] (based on [MP14]) and will later on be required in section 4.

2.3. Algebraic realization of the LOD. As explained above, the correction operator is constructed based on a number of local element correctors on given patches. In the following, we elaborate on the algebraic realization of these subproblems. Since the corrector computations all have a similar structure, we restrict ourselves to one particular problem. Let $K \in \mathcal{T}_H$ be fixed, and let ω be the patch with ℓ additional element layers around K, i.e., $\omega = \mathtt{N}^\ell(K)$. The construction of the local matrix requires the solution of a corrector problem on some fine scale h as introduced above. First, we define $V_h(\omega) := \{v_h \in V_h \mid \operatorname{supp}(v_h) \subseteq \omega\}$ and $V_H|_\omega := \{v_H|_\omega \mid v_H \in V_H\}$ with nodal basis functions $\{\lambda_j\}_{j=1}^{n_\omega}$ and $\{\Lambda_j\}_{j=1}^{N_\omega}$, respectively. Here, $n_\omega = \dim V_h(\omega)$ and $N_\omega = \dim V_H|_\omega$. For a given basis function Λ_i, the corrector problem seeks the solution $\mathcal{Q}_K^\ell(\Lambda_i|_K) \in V_h(\omega)$ to

$$a(\mathcal{Q}_K^\ell(\Lambda_i|_K), w_h) = a(\Lambda_i|_K, w_h) \tag{2.12}$$

for all $w_h \in V_h(\omega) \cap \ker \mathcal{I}_H$. Note that we need only compute the corrections for the 2^d basis functions for which $\operatorname{supp}\Lambda_i \cap K \neq \emptyset$. For simplicity and without loss of generality, we assume in the following that this is the case for the first indices $1, \dots, 2^d$.

To avoid defining a basis of $V_h(\omega) \cap \ker \mathcal{I}_H$, it is favorable to write (2.12) as an equivalent saddle point problem, which reads

$$\begin{array}{rcccl} a(\mathcal{Q}_K^\ell(\Lambda_i|_K), v_h) & + & (\varphi_i, \mathcal{I}_H v_h)_{L^2(\omega)} & = & a(\Lambda_i|_K, v_h), \\ (\mathcal{I}_H \mathcal{Q}_K^\ell(\Lambda_i|_K), \mu_H)_{L^2(\omega)} & & & = & 0 \end{array} \tag{2.13}$$

for all $v_h \in V_h(\omega)$ and $\mu_H \in V_H|_\omega$, where $\varphi_i \in V_H|_\omega$ is the associated Lagrange multiplier. Note that (2.13) has a unique solution pair $(\mathcal{Q}_T^\ell \Lambda_i, \varphi_i)$ and the problem is equivalent to solving (2.12); see, for instance, [Mai20, Chap. 2].

To solve problem (2.13) numerically, it is reformulated as a system of linear equations. We use the linear combinations

$$\mathcal{Q}_K^\ell(\Lambda_i|_K) = \sum_{j=1}^{n_\omega} \xi_j^i \lambda_j, \qquad \varphi_i = \sum_{j=1}^{N_\omega} \phi_j^i \Lambda_j,$$

and write $\boldsymbol{\Xi}^i = [\xi_1^i, \dots, \xi_{n_\omega}^i]^T$ and $\boldsymbol{\Phi}^i = [\phi_1^i, \dots, \phi_{N_\omega}^i]^T$ for the corresponding vectors.

Let $\mathbf{S} := \mathbf{S}_{A,\omega}$ be the finite element stiffness matrix with respect to $V_h(\omega)$ (weighted by the coefficient A and with built-in boundary conditions) and $\mathbf{P}_\omega \in \mathbb{R}^{n_\omega \times N_\omega}$ (resp., $\mathbf{P}_{\omega,K} \in \mathbb{R}^{n_\omega \times 2^d}$) the prolongation matrix between functions in $V_H|_\omega$ and $V_h(\omega)$ (resp., $V_H|_K$ and $V_h(\omega)$). Further, $\mathbf{I}_\omega \in \mathbb{R}^{N_\omega \times n_\omega}$ denotes the realization of the operator $\mathcal{I}_H$ on ω (without explicit boundary conditions). With these matrices, (2.13) can be expressed as

$$\begin{array}{rcl} \mathbf{S}\boldsymbol{\Xi} + \mathbf{I}_\omega^T \boldsymbol{\Phi} & = & \mathbf{S}\mathbf{P}_{\omega,K}, \\ \mathbf{I}_\omega \boldsymbol{\Xi} & = & \mathbf{0}_{N_\omega \times 2^d}, \end{array} \tag{2.14}$$

where $\boldsymbol{\Xi} = [\boldsymbol{\Xi}^1 | \dots | \boldsymbol{\Xi}^{2^d}]$ and $\boldsymbol{\Phi} = [\boldsymbol{\Phi}^1 | \dots | \boldsymbol{\Phi}^{2^d}]$ are matrices that comprise the vectors $\{\boldsymbol{\Xi}^i\}$ and $\{\boldsymbol{\Phi}^i\}$, respectively. Some calculations show that $\boldsymbol{\Phi} = (\mathbf{I}_\omega \mathbf{S}^{-1} \mathbf{I}_\omega^T)^{-1} \mathbf{I}_\omega \mathbf{P}_{\omega,K}$. Inserting this in the first equation of (2.14) yields

$$\boldsymbol{\Xi} = \mathbf{P}_{\omega,K} - \mathbf{S}^{-1} \mathbf{I}_\omega^T (\mathbf{I}_\omega \mathbf{S}^{-1} \mathbf{I}_\omega^T)^{-1} \mathbf{I}_\omega \mathbf{P}_{\omega,K}.$$

The local PG-LOD matrix with respect to the element K and the patch ω around K is defined by

$$\mathbf{S}_{\omega}^{\mathrm{pg}}[i,j] = a(\Lambda_i, (\mathsf{id} - \mathcal{Q}_K^{\ell})(\Lambda_j|_K)), \quad (i,j) \in \{1,\dots,N_\omega\} \times \{1,\dots,2^d\}. \tag{2.15}$$

In terms of the matrix system (2.14), $\mathbf{S}_{\omega}^{\mathrm{pg}}$ can be equivalently defined by

$$\begin{aligned} \mathbf{S}_{\omega}^{\mathrm{pg}} &= \mathbf{P}_{\omega}^T \mathbf{S}(\mathbf{P}_{\omega,K} - \mathbf{\Xi}) = \mathbf{P}_{\omega}^T \mathbf{S}\mathbf{S}^{-1}\mathbf{I}_{\omega}^T(\mathbf{I}_{\omega}\mathbf{S}^{-1}\mathbf{I}_{\omega}^T)^{-1}\mathbf{I}_{\omega}\mathbf{P}_{\omega,K} \\ &= \mathbf{P}_{\omega}^T \mathbf{I}_{\omega}^T(\mathbf{I}_{\omega}\mathbf{S}^{-1}\mathbf{I}_{\omega}^T)^{-1}\mathbf{I}_{\omega}\mathbf{P}_{\omega,K}, \end{aligned} \tag{2.16}$$

which will be an important representation in order to analyze the approximability by a suitable neural network in the following. Note that $\mathbf{S}_{\omega}^{\mathrm{pg}}$ implicitly depends on the coefficient A through the bilinear form a and the correction operator $\mathcal{Q}_K^{\ell}$.

2.4. Neural network approximation of the local subproblems. The method presented in [KMP22] can be used to approximate the local contributions to the LOD stiffness matrix, i.e., the matrices $\mathbf{S}_{\omega}^{\mathrm{pg}}$ of the previous subsections, by one single neural network that maps from a local instance of the coefficient on $\omega := \mathtt{N}^{\ell}(K)$ to an approximation of the matrix $\mathbf{S}_{\omega}^{\mathrm{pg}}$. For completeness, we briefly discuss the main ideas and refer to [KMP22, sect. 4] for numerical experiments that show the feasibility of the approach. In the subsequent section, we also rigorously investigate the strategy from the viewpoint of approximation theory.

The matrices $\mathbf{S}_{\omega}^{\mathrm{pg}}$, as defined in (2.15), are rectangular matrices that are characterized by the local degrees of freedom $i = 1,\dots,N_\omega$ and $j = 1,\dots,2^d$. For the final Petrov–Galerkin stiffness matrix corresponding to the discretized problem (2.8), we require the sum of the local corrections $\mathcal{Q}_K^{\ell}$; cf. (2.6). That is, to build the stiffness matrix, we need the sum of the local matrices $\mathbf{S}_{\omega}^{\mathrm{pg}}$ as well. This is done with appropriate local-to-global mappings Φ_K that transform the local matrix for an element $K \in \mathcal{T}_H$ to an inflated matrix with respect to the global degrees of freedom. The global stiffness matrix $\mathbf{S}^{\mathrm{pg}}$ thus has the form

$$\mathbf{S}^{\mathrm{pg}} = \sum_{K \in \mathcal{T}_H} \Phi_K\big(\mathbf{S}_{\mathtt{N}^{\ell}(K)}^{\mathrm{pg}}\big); \tag{2.17}$$

see [KMP22, sect. 3.4] for further details. As mentioned above, the local matrices $\mathbf{S}_{\mathtt{N}^{\ell}(K)}^{\mathrm{pg}}$ implicitly depend on A, but the mappings Φ_K are independent of the coefficient. The method for approximating the matrix $\mathbf{S}^{\mathrm{pg}}$ now keeps this coefficient-independent decomposition and only replaces the local matrices $\mathbf{S}_{\mathtt{N}^{\ell}(K)}^{\mathrm{pg}}$ by suitable approximations based on a single neural network. This is a reasonable approach due to the fact that the local corrector problems (2.12) all have a similar structure and are (up to the actual values of the coefficient) independent of their position within the domain.

In the following, we theoretically justify this strategy in the sense that we show that there exists a suitable network that provides a (local) surrogate such that the solution computed with the corresponding global stiffness matrix is reasonably close to the actual PG-LOD solution. In particular, we are interested in the dimensions (i.e., depth and number of nonzero parameters) of such a neural network. In section 3, we first focus on the approximability of the local LOD subproblems by a neural network, before we investigate the difference between the corresponding discrete global coarse-scale solutions in section 4.

3. Neural network approximation of local LOD matrices. In this section, we investigate the complexity of a neural network to approximate the mapping from a coefficient to the local PG-LOD matrix as given in (2.16) and give rigorous upper bounds for depth and number of nonzero parameters of such a network. To start with, we have to establish a framework for neural network approximation theory that we will then apply to our specific setting.

3.1. Neural network calculus. We adopt the main ideas and definitions of the formal framework developed in [PV18]. To familiarize the reader with the central concepts and the notation that we will subsequently use, we quickly recap the most important cornerstones of this framework as well as some results from [KPRS21] regarding the approximation of matrix inversion by deep ReLU-neural networks.

DEFINITION 3.1 (neural network). *A neural network Ψ of depth $L \in \mathbb{N}$ is a sequence of matrix-vector tuples*

$$\Psi = ((\mathbf{W}_l, \mathbf{b}_l))_{l=1}^{L},$$

where each layer $(\mathbf{W}_l, \mathbf{b}_l)$ consists of a weight matrix $\mathbf{W}_l \in \mathbb{R}^{N_l \times N_{l-1}}$ and a bias vector $\mathbf{b}_l \in \mathbb{R}^{N_l}$ for $N_0, \dots, N_L \in \mathbb{N}$.

We will refer to $L(\Psi) := L$ as the number of layers, to $\dim_{\mathrm{in}}(\Psi) := N_0$ as the input dimension and to $\dim_{\mathrm{out}}(\Psi) := N_L$ as the output dimension of Ψ. We call $M_l(\Psi) := \|\mathbf{W}_l\|_0 + \|\mathbf{b}_l\|_0$ the number of parameters in the lth layer, where $\|\cdot\|_0$ counts the number of nonzero entries in a given matrix or vector. The total number of parameters is then given by $M(\Psi) := \sum_{l=1}^{L} M_l$.

DEFINITION 3.2 (realization of a neural network). *Let Ψ be a neural network of depth L. For a set $S \subset \mathbb{R}^{N_0}$ and an activation function $\rho\colon \mathbb{R} \to \mathbb{R}$ that is assumed to act componentwise on vectors by convention, the realization of Ψ with activation function ρ over S is the mapping $\mathcal{R}_\rho^S(\Psi)\colon S \to \mathbb{R}^{N_L}$ implemented by the neural network Ψ. It is given by*

$$\mathcal{R}_\rho^S(\Psi)(\mathbf{x}) := \mathbf{x}_L,$$

where $\mathbf{x}_L$ results from

$$\begin{aligned} \mathbf{x}_0 &:= \mathbf{x}, \\ \mathbf{x}_l &:= \rho(\mathbf{W}_l\, \mathbf{x}_{l-1} + \mathbf{b}_l), \quad l = 1, \dots, L-1, \\ \mathbf{x}_L &:= \mathbf{W}_L\, \mathbf{x}_L + \mathbf{b}_L, \end{aligned}$$

i.e.,

$$\mathcal{R}_\rho^S(\Psi)(x) = \mathbf{W}_L\, \rho(\mathbf{W}_{L-1}(\dots \rho(\mathbf{W}_2\, \rho(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \dots) + \mathbf{b}_{L-1}) + \mathbf{b}_L.$$

Although many different activation functions ρ are conceivable and used in practice, we restrict ourselves to one of the most popular choices, the so-called *Rectified Linear Unit (ReLU)* activation function given by $\rho(x) := \max\{0, x\}$ in this work.

DEFINITION 3.3 (concatenation of neural networks). *Let $\Psi^1 := ((\mathbf{W}_l^1, \mathbf{b}_l^1))_{l=1}^{L_1}$ and $\Psi^2 := ((\mathbf{W}_l^2, \mathbf{b}_l^2))_{l=1}^{L_2}$ be two neural networks of depth L_1, L_2, respectively, such that $\dim_{\mathrm{in}}(\Psi^1) = \dim_{\mathrm{out}}(\Psi^2)$. Then the concatenation of Ψ^1 and Ψ^2 is denoted by $\Psi^1 \bullet \Psi^2$ and reads*

$$\begin{aligned} \Psi^1 \bullet \Psi^2 := ((\mathbf{W}_1^2, \mathbf{b}_1^2), \dots, (\mathbf{W}_{L_2-1}^2, \mathbf{b}_{L_2-1}^2), (\mathbf{W}_1^1 \mathbf{W}_{L_2}^2, \mathbf{W}_1^1 \mathbf{b}_{L_2}^2 + \mathbf{b}_1^1), \\ (\mathbf{W}_2^1, \mathbf{b}_2^1), \dots, (\mathbf{W}_{L_1}^1, \mathbf{b}_{L_1}^1)). \end{aligned}$$

It is easy to check that the realization of this concatenation implements the composition of the realizations of Ψ^1 and Ψ^2, i.e.,

$$\mathcal{R}_\rho^{\mathbb{R}^{N_0^2}}(\Psi^1 \bullet \Psi^2) = \mathcal{R}_\rho^{\mathbb{R}^{N_0^1}}(\Psi^1) \circ \mathcal{R}_\rho^{\mathbb{R}^{N_0^2}}(\Psi^2).$$

Thus, the concatenation of two neural networks can be interpreted as connecting them in series. The problem with this approach is that, in general, $M(\Psi^1 \bullet \Psi^2)$ cannot be bounded linearly with respect to $M(\Psi^1)$ and $M(\Psi^2)$. The next lemma shows one possible exception to this.

LEMMA 3.4. *Let Ψ^1 be a neural network of the form*

$$\Psi^1 = \big((\mu\mathbf{Q}, \mathbf{0}_n)\big), \quad \mu \in \mathbb{R},$$

where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is a permutation matrix and Ψ^2 a neural network with output dimension n. Then it holds that

(i) $L(\Psi^1 \bullet \Psi^2) = L(\Psi^2)$,
(i) $M(\Psi^1 \bullet \Psi^2) = M(\Psi^2)$.

Proof. This is a special case of [KPRS21, Lemma A.1.]. ∎

In the general case of more complicated networks, however, we need to introduce another type of concatenation, which is built on the construction of a two-layer neural network that implements the identity function.

LEMMA 3.5 (see [PV18, Lem. 2.3]). *Let $n \in \mathbb{N}$ and define the two-layer neural network*

$$\Psi_n^{\mathrm{Id}} := \left(\left(\begin{bmatrix} \mathbf{Id}_n \\ -\mathbf{Id}_n \end{bmatrix}, \mathbf{0}_{2n} \right), \left(\begin{bmatrix} \mathbf{Id}_n, -\mathbf{Id}_n \end{bmatrix}, \mathbf{0}_n \right) \right)$$

with input and output dimension n. Then it holds that

$$R_\rho^{\mathbb{R}^n}(\Psi_n^{\mathrm{Id}}) = \mathrm{Id}_{\mathbb{R}^n},$$

i.e., Ψ_n^{Id} implements the identity function on $\mathbb{R}^n$.

With this definition, we can introduce the so-called sparse concatenation that allows us to precisely control the number of nonzero parameters when two or more networks are connected in series.

DEFINITION 3.6 (sparse concatenation of neural networks). *Given two networks Ψ^1 and Ψ^2 of depth $L_1, L_2 \in \mathbb{N}$, respectively, with $\dim_{\mathrm{in}}(\Psi^1) = \dim_{\mathrm{out}}(\Psi^2) =: n$ as above, we define the sparse concatenation of Ψ^1 and Ψ^2 as*

$$\Psi^1 \odot \Psi^2 := \Psi^1 \bullet \Psi_n^{\mathrm{Id}} \bullet \Psi^2,$$

where Ψ_n^{Id} is the identity network from Lemma 3.5.

Obviously, the sparse concatenation does not change the function realized by the network compared to the regular concatenation and therefore also implements the composition of the individual realizations. Moreover, the following lemma shows the desired linear bound on $M(\Psi^1 \odot \Psi^2)$.

LEMMA 3.7 (cf. [EGJS21, Lem. 5.3]). *Given two neural networks Ψ^1 and Ψ^2, their sparse concatenation fulfills*

(i) $L(\Psi^1 \odot \Psi^2) \le L(\Psi^1) + L(\Psi^2)$,
(ii) $M(\Psi^1 \odot \Psi^2) \le M(\Psi^1) + M(\Psi^2) + M_1(\Psi^1) + M_{L(\Psi^2)}(\Psi^2) \le 2M(\Psi^1) + 2M(\Psi^2)$
$\lesssim M(\Psi^1) + M(\Psi^2)$.

Moreover, given $k>2$ neural networks $\Psi^1,\dots,\Psi^k$, their sparse concatenation satisfies

(i) $L(\Psi^1 \odot \cdots \odot \Psi^k) \le \sum_{i=1}^k L(\Psi^i)$,

(ii) $M(\Psi^1 \odot \cdots \odot \Psi^k) \lesssim \sum_{i=1}^k M(\Psi^i)$.

In order to construct more complex networks out of smaller building blocks, we also need to be able to connect them in parallel. The definition below describes how this can be done for networks with identical depths. In principle, network parallelization can also be defined for networks of different depths. However, since we do not require such a construction for our proofs in the next subsection, we restrict ourselves to the simpler case.

DEFINITION 3.8 (parallelization of neural networks). *Let $\Psi^i = ((\mathbf{W}_l^i, \mathbf{b}_l^i))_{l=1}^L$, $i = 1,\dots,k$, be a family of k neural networks with identical input dimension n and identical depth L. Then the parallelization of the Ψ^i is given by*

$$P(\Psi^1,\dots,\Psi^k) = \left(\left(\begin{bmatrix} \mathbf{W}_1^1 & & \\ & \ddots & \\ & & \mathbf{W}_1^k \end{bmatrix}, \begin{bmatrix} \mathbf{b}_1^1 \\ \vdots \\ \mathbf{b}_1^k \end{bmatrix}\right), \dots, \left(\begin{bmatrix} \mathbf{W}_L^1 & & \\ & \ddots & \\ & & \mathbf{W}_L^k \end{bmatrix}, \begin{bmatrix} \mathbf{b}_L^1 \\ \vdots \\ \mathbf{b}_L^k \end{bmatrix}\right)\right).$$

For the realizations of parallelized networks of identical input dimension n, it holds that

$$R_\rho^{\mathbb{R}^n}(\Psi^1,\dots,\Psi^k)(\mathbf{x_1},\dots,\mathbf{x_k}) = [R_\rho^{\mathbb{R}^n}(\Psi^1)(\mathbf{x_1}),\dots,R_\rho^{\mathbb{R}^n}(\Psi^k)(\mathbf{x_k})]$$

for $\mathbf{x_1},\dots,\mathbf{x_k} \in \mathbb{R}^n$.

LEMMA 3.9. *Let $\Psi^1,\dots,\Psi^k$ be a family of k neural networks with identical input dimension and identical depth L. Then the parallelization $P(\Psi^1,\dots,\Psi^k)$ fulfills*

(i) $L(P(\Psi^1,\dots,\Psi^k)) = L$,

(ii) $M(P(\Psi^1,\dots,\Psi^k)) = \sum_{i=1}^k M(\Psi^i)$.

Proof. The result follows by construction. ∎

Now that we have laid down the foundations of neural network calculus, we turn to the question of how to approximate matrix inversion with a neural network. Since in the standard formulation of feedforward networks the input and output are one-dimensional arrays, we choose to consider columnwise "flattened" (or vectorized) matrices. For $\mathbf{A} \in \mathbb{R}^{k\times l}$ we thus write

$$\operatorname{vec}(\mathbf{A}) := [A_{1,1},\dots,A_{k,1},\dots,A_{1,l},\dots,A_{k,l}]^T \in \mathbb{R}^{k\cdot l}$$

and, conversely, for $\mathbf{v} = [v_{1,1},\dots,v_{k,1},\dots,v_{1,l},\dots,v_{k,l}]^T \in \mathbb{R}^{k\cdot l}$,

$$\operatorname{mat}(\mathbf{v}) := (v_{i,j})_{i,j=1}^{k,l} \in \mathbb{R}^{k\times l}.$$

Furthermore, we define for $n \in \mathbb{N}$ and $Z > 0$ the set

$$K_n^Z := \left\{\operatorname{vec}(\mathbf{A}) \mid \mathbf{A} \in \mathbb{R}^{n\times n}, \|\mathbf{A}\|_2 \le Z\right\},$$

where $\|\cdot\|_2$ denotes the spectral norm induced by the Euclidean vector norm. The idea behind the approximation of matrix inversion with a deep neural network is based on approximating the Neumann series of matrices which can be suitably bounded in terms of their spectral norm. That is, we approximate the map

$$\operatorname{inv} : \{\mathbf{A} \in \mathbb{R}^{n\times n} \mid \|\mathbf{A}\|_2 \le 1-\delta\} \to \mathbb{R}^{n\times n}, \qquad \mathbf{A} \mapsto (\mathbf{Id}_n - \mathbf{A})^{-1} = \sum_{k=0}^{\infty} \mathbf{A}^k.$$

The following theorem makes this precise and shows that matrix inversion can be approximated up to arbitrary precision.

THEOREM 3.10 (see [KPRS21, Thm. 3.8]). *For $n \in \mathbb{N}, \vartheta \in (0, 1/4)$ and $\delta \in (0, 1)$, let*

$$m(\vartheta, \delta) := \left\lceil \frac{\log(0.5\vartheta\delta)}{\log(1-\delta)} \right\rceil.$$

Then there exists a neural network $\Psi_{\mathrm{inv},\vartheta}^{1-\delta,n}$ with input dimension n^2 and output dimension n^2 with

(i) $L(\Psi_{\mathrm{inv},\vartheta}^{1-\delta,n}) \lesssim \log(m(\vartheta,\delta)) \big(\log(1/\vartheta) + \log(m(\vartheta,\delta)) + \log(n)\big)$,

(ii) $M(\Psi_{\mathrm{inv},\vartheta}^{1-\delta,n}) \lesssim m(\vartheta,\delta)\, \log^2(m(\vartheta,\delta))\, n^3 \big(\log(1/\vartheta) + \log(m(\vartheta,\delta)) + \log(n)\big)$, *such that for all $\mathrm{vec}(\mathbf{A}) \in K_n^{1-\delta}$ it holds that*

(iii) $\sup_{\mathrm{vec}(\mathbf{A}) \in K_n^{1-\delta}} \| (\mathbf{Id}_n - \mathbf{A})^{-1} - \mathrm{mat}(R_\rho^{K_n^{1-\delta}}(\Psi_{\mathrm{inv},\vartheta}^{1-\delta,n})(\mathrm{vec}(\mathbf{A}))) \|_2 \leq \vartheta$,

(iv) $\|\mathrm{mat}(R_\rho^{K_n^{1-\delta}}(\Psi_{\mathrm{inv},\vartheta}^{1-\delta,n})(\mathrm{vec}(\mathbf{A})))\|_2 \leq \vartheta + \|(\mathbf{Id}_n - \mathbf{A})^{-1}\|_2 \leq \vartheta + \frac{1}{\delta}$.

Remark 3.11 (conservation of symmetry in neural network matrix inversion). In the original construction of the network $\Psi_{\mathrm{inv},\vartheta}^{1-\delta,n}$ given in [KPRS21], the network is not symmetry-preserving in the sense that if $\mathbf{A}$ is a symmetric matrix, the output matrix

$$\mathrm{mat}\Big(R_\rho^{K_n^{1-\delta}}\big(\Psi_{\mathrm{inv},\vartheta}^{1-\delta,n}\big)(\mathrm{vec}(\mathbf{A}))\Big)$$

might not be symmetric although $(\mathbf{Id}_n - \mathbf{A})^{-1}$ is. For our intended application below, this is not very convenient and makes the error analysis much more difficult. However, the construction can be slightly adapted to preserve the symmetry. Using our notation, we define in the induction step of the proof of [KPRS21, Proposition A.4] the modified network

$$\widetilde{\Psi}_{k,\vartheta}^{1/2,n} := \widetilde{\Psi}_{\mathrm{mult},\vartheta/4}^{1,n,n,n} \odot P\left(\Psi_{2^j,\vartheta}^{1/2,n}, \Psi_{t,\vartheta}^{1/2,n}\right) \bullet \left(\left(\begin{bmatrix}\mathbf{Id}_{n^2}\\ \mathbf{Id}_{n^2}\end{bmatrix}, \mathbf{0}_{2n^2}\right)\right),$$

where $\widetilde{\Psi}_{\mathrm{mult},\vartheta/4}^{1,n,n,n}$ is a network that is constructed analogously to the network implementing general matrix-matrix multiplication in [KPRS21, Proposition 3.7]. However, instead of the three-dimensional array $\mathbf{D}$ in the proof therein, we consider a modified array $\widetilde{\mathbf{D}}$, that for $i, j, k \in \{1, \ldots, n\}$ and $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ is defined by

$$\widetilde{\mathbf{D}}_{i,k,j}(\mathrm{vec}(\mathbf{A}), \mathrm{vec}(\mathbf{B})) := \begin{cases} (\mathbf{A}_{i,k}, \mathbf{B}_{k,j}) & \text{if } i \leq j, \\ (\mathbf{B}_{i,k}, \mathbf{A}_{k,j}) & \text{if } i > j, \end{cases}$$

i.e., it contains ordered tuples of the corresponding entries in $\mathbf{A}, \mathbf{B}$. Then the desired result follows analogously to [KPRS21].

Note that the result in Theorem 3.10 above does not utilize any structural properties of the matrix to be inverted such as sparsity, positive definiteness or symmetry. If one has additional structural knowledge about the matrix, as in our intended application in the sections below, we believe that by approximating more sophisticated solvers, for example multigrid approaches, with a neural network, one could improve upon the cubic complexity in the number of nonzero parameters.

The previous theorem is the key ingredient to proving that the approximation of the local LOD matrices can be realized by deep neural networks, to which we turn now.

3.2. Application to local PG-LOD matrices. Before we formulate the central result of this section, we have to introduce some more notation. We define

$$\mathfrak{A} := \{A \in L^\infty(D) \,|\, 0 < \alpha \leq A(x) \leq \beta < \infty \text{ for a.e. } x \in D\},$$

which is the set of admissible coefficients for problem (2.1) according to (2.2), where now α, β are fixed constants. Since neural networks take only discrete information as input, we restrict ourselves to the finite-dimensional subset

$$\mathfrak{A}_\varepsilon := \{A \in \mathfrak{A} \,|\, A|_T \text{ is constant for every } T \in \mathcal{T}_\varepsilon\} \subseteq \mathfrak{A}$$

of elementwise constant coefficients on $\mathcal{T}_\varepsilon$. Recall that the meshes $\mathcal{T}_\varepsilon$ and $\mathcal{T}_h$ are assumed to be uniform refinements of $\mathcal{T}_H$ and that for a given patch $\omega = \mathtt{N}^\ell(K)$, $K \in \mathcal{T}_H$, the local matrices $\mathbf{S}_\omega^{\mathrm{pg}}$ from (2.15) that we want to approximate solely depend on the values of A on ω.

Note that we implicitly assume that the patch ω has "full" size, i.e., that it corresponds to an element K that is at least $\ell+1$ layers of coarse elements away from the boundary of D. We emphasize, however, that the treatment of the boundary cases can as well be incorporated into the theory below by changing the overall complexity of the network only by a moderate factor. The first ingredient is that patches closer to the boundary can be artificially extended by elements where the coefficient has the value zero to keep a uniform size of all patches; we refer to [KMP22, sect. 3.4] for further details. Further, a slight adaptation of the matrices used in the algebraic realization of the method presented in section 2.3 is required for the boundary cases. This adaptation is conceptually straightforward but rather technical, since one has to account for artificial "degrees of freedom" outside the physical domain D. For better readability, we thus only focus on the inner patches in the proof of Theorem 3.12 below.

We enumerate the elements in the restricted mesh $\mathcal{T}_\varepsilon(\omega)$ by $1, \dots, m_\ell := ((2\ell+1)H/\varepsilon)^d$, which allows us to store the respective values of the coefficient in a vector $\mathbf{A}_{\varepsilon,\omega} \in \mathbb{R}^{m_\ell}$. Based on this, we consider the set

$$\mathfrak{A}_{\varepsilon,\omega} := \{\mathbf{A}_{\varepsilon,\omega} \,|\, A \in \mathfrak{A}_\varepsilon\} \subseteq \mathbb{R}^{m_\ell}$$

of all possible input vectors corresponding to the patch ω. Moreover, we enumerate the inner nodes of $\mathcal{T}_h(\omega)$ by $1, \dots, n_\ell := ((2\ell+1)H/h - 1)^d$ and inner and boundary nodes of $\mathcal{T}_H(\omega)$ by $1, \dots, N_\ell := (2\ell+2)^d$.

THEOREM 3.12 (neural network approximation of local PG-LOD matrices). *Let $\eta \in (0, 1/4)$ be a given error tolerance, and let $\ell \in \mathbb{N}$. Then there exists a neural network Ψ_η^{pg} with input dimension m_ℓ and output dimension $N_\ell 2^d$ such that for any $\omega = \mathtt{N}^\ell(K)$, $K \in \mathcal{T}_H$, and any $\mathbf{A}_{\varepsilon,\omega} \in \mathfrak{A}_{\varepsilon,\omega}$ it holds that*

$$\Big\| \mathbf{S}_\omega^{\mathrm{pg}} - \mathrm{mat}\Big(\mathcal{R}_\rho^{\mathfrak{A}_{\varepsilon,\omega}}(\Psi_\eta^{\mathrm{pg}})(\mathbf{A}_{\varepsilon,\omega}) \Big) \Big\|_2 \leq \eta.$$

The network Ψ_η^{pg} has the following properties;

(i) $L(\Psi_\eta^{\mathrm{pg}}) \lesssim \log(m(\theta,\delta)) \big(\log(1/\theta) + \log(m(\theta,\delta)) + \log(n_\ell) \big)$
$+ \log(m(\gamma,\widehat{\delta})) \big(\log(1/\gamma) + \log(m(\gamma,\widehat{\delta})) + \log(N_\ell) \big) + 1,$

(ii) $M(\Psi_\eta^{\mathrm{pg}}) \lesssim m(\theta,\delta) \log^2(m(\theta,\delta)) n_\ell^3 \big(\log(1/\theta) + \log(m(\theta,\delta)) + \log(n_\ell) \big)$
$+ m(\gamma,\widehat{\delta}) \log^2(m(\gamma,\widehat{\delta})) N_\ell^3 \big(\log(1/\gamma) + \log(m(\gamma,\widehat{\delta})) + \log(N_\ell) \big) + (\ell H/h)^{2d},$

where $\delta \approx \lambda_{\min}(\mathbf{S})\, \lambda_{\max}(\mathbf{S})^{-1}$ and $\widehat{\delta} \approx \lambda_{\min}(\mathbf{I}_\omega \mathbf{S}^{-1} \mathbf{I}_\omega^T)\, \lambda_{\max}(\mathbf{I}_\omega \mathbf{S}^{-1} \mathbf{I}_\omega^T)^{-1}$. Further, the values $\theta, \gamma \in (0, \eta)$ are chosen such that

$$\theta < \min \left\{ \lambda_{\min}(\mathbf{I}_\omega \mathbf{S}^{-1} \mathbf{I}_\omega^{\mathbf{T}}), \frac{\lambda_{\min}(\mathbf{I}_\omega \mathbf{S}^{-1} \mathbf{I}_\omega^T)}{2\, \mathfrak{v}_{\mathrm{resc}} \|\mathbf{I}_\omega^T\|_2^2} \right\}$$

and

$$\frac{\mathfrak{v}_{\mathrm{resc}}\,\|\mathbf{P}_{\omega,K}\|_2\,\|\mathbf{P}_\omega\|_2\,\|\mathbf{I}_\omega\|_2^4}{\widehat{\mathfrak{V}}_{\mathrm{inf}}^2}\,\theta + \|\mathbf{P}_{\omega,K}\|_2\,\|\mathbf{P}_\omega\|_2\,\|\mathbf{I}_\omega\|_2^2\,\gamma \le \eta$$

with $\mathfrak{v}_{\mathrm{resc}} \approx \lambda_{\max}(\mathbf{S})^{-1}$ *and* $\widehat{\mathfrak{V}}_{\mathrm{inf}} \approx \lambda_{\min}(\mathbf{I}_\omega \mathbf{S}^{-1} \mathbf{I}_\omega^T)$*; cf. also the precise definitions in* (A.2) *and* (A.6), (A.7) *in the proof.*

Proof. The main idea of the proof is to start from the representation (2.16) of the local PG-LOD stiffness matrix and subsequently implement or approximate all matrix multiplications or inversions of the mapping $\mathbf{A}_{\varepsilon,\omega} \mapsto \mathbf{S}_\omega^{\mathrm{pg}}$ by suitably constructed neural networks. The matrix multiplications can be implemented exactly by single-layer networks, the inversions, however, have to be approximated using Theorem 3.10. In the end, these individual building blocks can be connected in series to obtain a final network with the desired properties. A detailed proof can be found in section A.1 of the appendix. ∎

We close the section by bounding the size of a network in order for it to well-approximate any local PG-LOD stiffness matrix with an error of size $\mathcal{O}(H^k)$, $k \in \mathbb{N}$, which will be essential for the error estimation of the global errors between the two surrogate models that are based on deterministic local PG-LOD matrices and their network approximations, respectively.

Corollary 3.13. *Let* $H < 1/4$ *and* $\ell \approx |\log(H)|$*. For any* $k \in \mathbb{N}$*, there exists a neural network* $\Psi_{H^k}^{\mathrm{pg}}$ *with input dimension* m_ℓ*, and output dimension* $N_\ell 2^d$ *such that for any* $\omega = \mathtt{N}^\ell(K)$*,* $K \in \mathcal{T}_H$*, and any* $\mathbf{A}_{\varepsilon,\omega} \in \mathfrak{A}_{\varepsilon,\omega}$

$$\Big\| \mathbf{S}_\omega^{\mathrm{pg}} - \mathrm{mat}\Big(\mathcal{R}_\rho^{\mathfrak{A}_{\varepsilon,\omega}}(\Psi_{H^k}^{\mathrm{pg}})(\mathbf{A}_{\varepsilon,\omega}) \Big) \Big\|_2 \lesssim H^k.$$

Further, we have

(i) $L(\Psi_{H^k}^{\mathrm{pg}}) \lesssim |\log(h)|^2 + |\log(k)|^2$,

(ii) $M(\Psi_{H^k}^{\mathrm{pg}}) \lesssim h^{-2}(k|\log(H)| + |\log(h)|)(|\log(h)|^3 + |\log(k)|^3)(|\log(H)|H/h)^{3d}$.

Proof. The result is obtained by applying Theorem 3.12 above with $\eta = H^k$ and estimating all quantities in the resulting upper bounds on depth and number of nonzero parameters in terms of the mesh sizes H, h. For more details, see the full proof in section A.2 of the appendix. ∎

Remark 3.14. Note that the goal of the above results is to show that a suitable neural network exists that allows one to approximate the local PG-LOD matrices up to arbitrary accuracy. We emphasize that this does not guarantee that such a network is actually learned during the training phase in practice. Further, the bounds on the number of layers and nonzero parameters of the network have to be understood as worst-case estimates. In the numerical experiments section of [KMP22] it was shown that the estimates presented above indeed seem very pessimistic and one is able to train a network that produces satisfying results with significantly fewer parameters.

4. Error analysis of the neural network enhanced surrogate model. In the previous section, we have seen that the local PG-LOD matrices individually can be well-approximated by a neural network. However, in applications we are usually more concerned about how this translates to the resulting difference of the respective global solutions when applied to our test problem. This involves multiple error sources that will be discussed in this section.

In the following, we investigate the error in solutions between the solution $u_H^{\mathrm{pg}} \in V_H$ of (2.8), i.e., the PGLOD solution, and the solution $u_H^{\mathrm{nn}} \in V_H$ obtained from the stiffness matrix which is assembled based on the (optimal) local neural network according to Theorem 3.12 for a suitable tolerance η. Let $\mathbf{u}^{\mathrm{pg}}$ be the vector corresponding to u_H^{pg}, i.e.,

$$u_H^{\mathrm{pg}} = \sum_{j=1}^{N} \mathbf{u}_j^{\mathrm{pg}} \Lambda_j. \tag{4.1}$$

Here, $\{\Lambda_j\}_{j=1}^N$ denote the nodal basis functions of V_H with $N = \dim V_H$. As above, we write

$$\mathbf{S}^{\mathrm{pg}} = \sum_{K \in \mathcal{T}_H} \Phi_K\big(\mathbf{S}^{\mathrm{pg}}_{\mathtt{N}^\ell(K)}\big); \tag{4.2}$$

cf. (2.17). In the same manner, let $\mathbf{u}^{\mathrm{nn}}$ be the vector corresponding to u_H^{nn} and let

$$\mathbf{S}^{\mathrm{nn}} := \sum_{K \in \mathcal{T}_H} \Phi_K\big(\mathbf{\Theta}_K\big) \tag{4.3}$$

be the respective stiffness matrix. Note that the matrices

$$\mathbf{\Theta}_K := \mathrm{mat}\big(\mathcal{R}_\rho^{\mathfrak{A}_{\varepsilon,\mathtt{N}^\ell(K)}}(\Psi_\eta^{\mathrm{pg}})(\mathbf{A}_{\varepsilon,\mathtt{N}^\ell(K)})\big) \tag{4.4}$$

are the local matrices obtained by a forward pass of the local instance $\mathbf{A}_{\varepsilon,\mathtt{N}^\ell(K)}$ of a fixed coefficient A on $\mathtt{N}^\ell(K)$ through the neural network; cf. section 3.1, particularly Theorem 3.12 and Corollary 3.13. Our goal is to investigate under which conditions the error $\|u_H^{\mathrm{pg}} - u_H^{\mathrm{nn}}\|_{L^2(D)}$ is bounded by some given tolerance, ideally of order $\mathcal{O}(H)$. This would be optimal since u_H^{pg} already includes an error which scales at least like $\mathcal{O}(H)$ compared to the ideal solution to (2.1). Note that we have the following norm equivalence (see, e.g., [Fri73] and observe that the eigenvalues of the local mass matrix are bounded by $H^d 6^{-d}$ and $H^d 2^{-d}$ from below and above, respectively),

$$\big(H/6\big)^d \|\mathbf{v}\|_2^2 \le \mathbf{v}^T \mathbf{M} \mathbf{v} = \|v\|_{L^2(D)}^2 \le H^d \|\mathbf{v}\|_2^2, \tag{4.5}$$

where $v \in V_H$ and $\mathbf{v}$ is the corresponding vector. Here, $\mathbf{M}$ denotes the classical finite element mass matrix. Therefore, it holds that

$$\|u_H^{\mathrm{pg}} - u_H^{\mathrm{nn}}\|_{L^2(D)}^2 = (\mathbf{u}^{\mathrm{pg}} - \mathbf{u}^{\mathrm{nn}})^T \mathbf{M} (\mathbf{u}^{\mathrm{pg}} - \mathbf{u}^{\mathrm{nn}}) \approx H^d \|\mathbf{u}^{\mathrm{pg}} - \mathbf{u}^{\mathrm{nn}}\|_2^2.$$

To investigate the error between these two solutions, it is favorable to consider the symmetric version of the LOD for an intermediate step. This will be treated in the following subsection.

4.1. Difference between C-LOD and PG-LOD approximation. Let $u_H^{\mathrm{c}} \in V_H$ be the solution to (2.7) with corresponding vector $\mathbf{u}^{\mathrm{c}}$ as above. The stiffness matrix to (2.7) is denoted $\mathbf{S}^{\mathrm{c}}$. The next lemma shows that the difference $u_H^{\mathrm{c}} - u_H^{\mathrm{pg}}$, respectively $\mathbf{u}^{\mathrm{c}} - \mathbf{u}^{\mathrm{pg}}$, exponentially decays with increasing localization parameter ℓ introduced in section 2.2.

LEMMA 4.1. *Let* $u_H^{\mathrm{c}} \in V_H$ *and* $u_H^{\mathrm{pg}} \in V_H$ *be the solutions of* (2.7) *and* (2.8), *respectively, and let* $\mathbf{u}^{\mathrm{c}}$ *and* $\mathbf{u}^{\mathrm{pg}}$ *be the corresponding vectors; cf.* (4.1). *Then*

$$\|u_H^{\mathrm{c}} - u_H^{\mathrm{pg}}\|_{L^2(D)} \lesssim \exp(-c\ell)$$

and

$$\|\mathbf{u}^{\mathrm{c}} - \mathbf{u}^{\mathrm{pg}}\|_2 \lesssim H^{-d/2} \exp(-c\ell),$$

where the constant hidden in $\lesssim$ *depends on the right-hand side* f.

Proof. The proof is based on the observation that the error between the solutions to (2.7) and (2.8) can be reduced to a decay estimate of the form (2.11). The full proof is stated in section B.1 of the appendix. ∎

This result will allow us to switch from the PG-LOD approximation to the C-LOD approximation in the next subsection. Note that the C-LOD approximation is much more convenient to use in connection with spectral estimates. In particular, we have the following lemma.

LEMMA 4.2. *Let* $\mathbf{S}^{\mathrm{c}}$ *be the stiffness matrix corresponding to* (2.7). *Its minimal eigenvalue fulfills*

$$\lambda_{\min}(\mathbf{S}^{\mathrm{c}}) \geq cH^d$$

with a constant c *that does not depend on the mesh size* H.

Proof. The proof is given in section B.2 of the appendix. ∎

4.2. Error in solution. With the preliminary considerations of the previous subsection, we can now investigate the error between the solutions u_H^{pg} and u_H^{nn}.

THEOREM 4.3 (coarse-scale error). *Let* $H < 1/4$, $\ell \approx |\log(H)|$. *There exists a neural network* Ψ^{pg} *with*

$$L(\Psi^{\mathrm{pg}}) \lesssim |\log(h)|^2 \quad \text{and} \quad M(\Psi^{\mathrm{pg}}) \lesssim h^{-2} |\log(h)|^4 (|\log(H)|H/h)^{3d} \tag{4.6}$$

such that for any $A \in \mathfrak{A}_\varepsilon$ *the solutions* $u_H^{\mathrm{pg}} \in V_H$ *of* (2.8) *(and its vector representation* $\mathbf{u}^{\mathrm{pg}}$*) and the network solution* u_H^{nn} *(resp., the vector* $\mathbf{u}^{\mathrm{nn}}$*) fulfill*

$$\|\mathbf{u}^{\mathrm{pg}} - \mathbf{u}^{\mathrm{nn}}\|_2 \lesssim H^{1-d/2} \quad \text{and} \quad \|u_H^{\mathrm{pg}} - u_H^{\mathrm{nn}}\|_{L^2(D)} \lesssim H.$$

Proof. The proof reduces the stated error to local contributions that can be estimated using Corollary 3.13. To employ useful properties of the symmetric and positive definite matrix $\mathbf{S}^{\mathrm{c}}$, we take an intermediate step to first estimate the error compared to the symmetric solution $u_H^{\mathrm{c}} \in V_H$ of (2.7) (resp., the vector $\mathbf{u}^c$) and make use of Lemma 4.1 and the eigenvalue bound in Lemma 4.2. The detailed proof is presented in section B.3. ∎

Theorem 4.3 shows that there exists a neural network such that the approach of section 2 leads to a global coarse-scale error between the discrete PG-LOD solution and its neural network-based variant of order $\mathcal{O}(H)$. The size of such a network can be bounded dependent on the scales H and h. This leads to an overall error compared to the exact solution to the elliptic problem of the order $\mathcal{O}(H)$ as well, provided that h is reasonably small; see also (2.10). In certain cases, the necessary choice of the scale h (and thus the dependence of the size of the network) can be stated in terms of H and ε only as investigated in the following section.

5. How fine is enough? LOD on an appropriate fine scale. In the previous sections, we have investigated how PG-LOD stiffness matrices can be well-approximated by a local neural network in the sense that the error of the (global) coarse discretizations are reasonably close with respect to the mesh size H. Note that the dimensions of the network depend on the scale h on which the corrector problems (2.12) are computed. In this section, we want to investigate how fine this fine scale actually needs to be.

5.1. Finding an optimal fine computational scale. In general, the corrections for the LOD may be computed on an arbitrarily fine scale h. Choosing a finer h also means smaller discretization errors and we generally have

$$\|u - u_h\|_{L^2(D)} \to 0 \quad \text{as } h \to 0$$

for the first term on the right-hand side of (2.10). For the corrector problems, however, a finer h also increases the size of the linear systems that need to be solved. As investigated in section 3.1, this leads to more parameters in the neural network which approximates the local contributions. Since the network itself already introduces an error and the LOD also comes with an error of order $\mathcal{O}(H)$, we now seek the largest possible scale h with $0 < h < H$ such that the scale h is sufficient to still obtain an overall error of order H while minimizing the necessary parameters of the network. This particularly means that the term $\|u - u_h\|_{L^2(D)}$ should be of order $\mathcal{O}(H)$. We have the following result.

LEMMA 5.1 (fine-scale error). *There exists an $s > 0$ (depending on α, β) such that for any $A \in \mathfrak{A}_\varepsilon$, the solution u to* (2.1) *and its Galerkin approximation u_h in V_h satisfy $u \in H^{1+s}(D)$ and*

$$\|\nabla(u - u_h)\|_{L^2(D)} \lesssim h^s \|u\|_{H^{1+s}(D)}. \tag{5.1}$$

Proof. The regularity results presented in [Pet01, Chap 2] state that there exists some $s > 0$ such that $u \in H^{1+s}(D)$ for our class of coefficients $\mathfrak{A}_\varepsilon$. With an appropriate interpolation operator $\mathcal{I}_h$, we may thus derive (see, e.g., [EG17, Thm. 6.4])

$$\|\nabla(u - u_h)\|_{L^2(D)} \lesssim \|\nabla(1 - \mathcal{I}_h)u\|_{L^2(D)} \leq Ch^s \|u\|_{H^{1+s}(D)}. \qquad \square$$

Remark 5.2 (dependence on ε). If $A \in W^{1,\infty}(D)$ with oscillations on the scale ε, i.e., $\|A\|_{W^{1,\infty}(D)} \leq C\varepsilon^{-1}$, we have $s = 1$ in Lemma 5.1 with $\|u\|_{H^2(D)} \lesssim h/\varepsilon \|f\|_{L^2(D)}$ (see, e.g., the bounds in the proofs of [PS12, Lem. 4.3] or [MP19, Lem. 3.3]). That is, the choice $h \approx H\varepsilon$ leads to an error of size $\mathcal{O}(H)$.

If instead $A \in \mathfrak{A}_\varepsilon$, the condition on h reads $h \approx H^{1/s} \|u\|^{-1/s}_{H^{1+s}(D)}$. We emphasize that the norm of u does not depend on h and one can expect that $\|u\|_{H^{1+s}(D)} = \mathcal{O}(\varepsilon^{-s})$, leading to the optimal choice $h \approx H^{1/s}\varepsilon$.

With the above considerations and Theorem 4.3, we can finally state a result that quantifies the worst-case size and depth of a (local) neural network to achieve an overall error of the corresponding global surrogate of order $\mathcal{O}(H)$ independently of the scale h (which, in practice, could be chosen arbitrarily small).

COROLLARY 5.3. *Let the solution u of* (2.1) *fulfill $u \in H^{1+s}(D)$, and let $\|u\|_{H^{1+s}(D)} = \mathcal{O}(\varepsilon^{-s})$ for some $s > 0$. Then, there exists a local neural network Ψ^{pg} with depth*

$$L(\Psi^{\mathrm{pg}}) \lesssim s^{-2} |\log(H)|^2 + |\log(\varepsilon)|^2$$

and size

$$M(\Psi^{\mathrm{pg}}) \lesssim H^{-2/s}\varepsilon^{-2}\left(s^{-4}|\log(H)|^4 + |\log(\varepsilon)|^4\right)\left(|\log(H)|H^{1-1/s}\varepsilon^{-1}\right)^{3d}$$

such that the error of u and the solution $u_H^{\mathrm{nn}} \in V_H$ obtained from the stiffness matrix which is assembled based on Ψ^{pg} fulfills

$$\|u - u_H^{\mathrm{nn}}\|_{L^2(D)} \lesssim H.$$

6. Conclusion and outlook. In this paper, we have theoretically investigated the approximation properties of neural networks to approximate the coarse-scale contributions of local subproblems in numerical homogenization. The computation of these local problems that are solved on a fine scale is the bottleneck when computing reliable coarse-scale surrogates in numerical homogenization. Therefore, replacing this process by a single trained neural network allows for a significant speedup, because the computation of an appropriate surrogate is reduced to simple forward passes through the network. We have focused on the LOD method as a representative numerical homogenization method and presented upper bounds on the size of a network that leads to a total error between the deterministic discrete approximation and its variant based on the output surrogate of a trained neural network that scales linearly with respect to the coarse scale of interest.

We emphasize that the presented results focus on general approximation properties and do not provide an answer to the important question regarding whether and how optimal approximating networks can be found through training from data, which might not be possible at all for certain tasks as recently shown in [CAH21]. Nevertheless, our findings provide insight into how the dimensions of a suitable neural network for the approximation of the local subproblems in numerical homogenization should be chosen if the target scale of interest and the oscillation scale of the coefficient are given. This serves as a first step in developing mathematically rigorous guidelines on how to design suitable neural network architectures for numerical homogenization tasks. Further investigations into this direction are subject to future research.

Appendix A. Proofs in section 3.

A.1. Proof of Theorem 3.12. As mentioned at the beginning of section 3.2, we assume that $K \in \mathcal{T}_H$ is an element that is at least $\ell + 1$ layers of elements away from the boundary of D and set $\omega = \mathtt{N}^\ell(K)$. The idea of the proof is to start from the representation of the local PG-LOD matrix given in (2.16), namely

$$\mathbf{S}_\omega^{\mathrm{pg}} = \mathbf{P}_\omega^T \mathbf{I}_\omega^T (\mathbf{I}_\omega \mathbf{S}^{-1} \mathbf{I}_\omega^T)^{-1} \mathbf{I}_\omega \mathbf{P}_{\omega,K}, \tag{A.1}$$

and decomposing the approximation of the mapping $\mathbf{A}_{\varepsilon,\omega} \mapsto \mathrm{vec}(\mathbf{S}_\omega^{\mathrm{pg}})$ into the following seven consecutive steps:

(1) Linear transformation $\mathbf{A}_{\varepsilon,\omega} \mapsto \mathrm{vec}(\mathbf{S})$,
(2) Inversion $\mathrm{vec}(\mathbf{S}) \mapsto \mathrm{vec}(\mathbf{S}^{-1})$,
(3) Linear transformation $\mathrm{vec}(\mathbf{S}^{-1}) \mapsto \mathrm{vec}(\mathbf{S}^{-1}\mathbf{I}_\omega^T) =: \mathrm{vec}(\mathbf{X})$,
(4) Linear transformation $\mathrm{vec}(\mathbf{X}) \mapsto \mathrm{vec}(\mathbf{I}_\omega \mathbf{X}) =: \mathrm{vec}(\mathbf{Y})$,
(5) Inversion $\mathrm{vec}(\mathbf{Y}) \mapsto \mathrm{vec}(\mathbf{Y}^{-1})$,
(6) Linear transformation $\mathrm{vec}(\mathbf{Y}^{-1}) \mapsto \mathrm{vec}(\mathbf{Y}^{-1}\mathbf{I}_\omega \mathbf{P}_{\omega,K}) =: \mathrm{vec}(\mathbf{Z})$,
(7) Linear transformation $\mathrm{vec}(\mathbf{Z}) \mapsto \mathrm{vec}(\mathbf{P}_\omega^T \mathbf{I}_\omega^T \mathbf{Z})$.

Note that, in fact, steps 1, 3, 4, 6, and 7 can be implemented exactly by single-layer neural networks. The matrix inversions in steps 2 and 5, however, have to

be approximated up to some tolerance. Our goal is, therefore, to construct neural networks $\Psi^1, \dots, \Psi^7$, which implement and approximate these steps. Connecting those building blocks in series then yields a network Ψ_η^{pg} with the desired properties.

Step 1. The key insight in the first step is that the mapping $\mathbf{A}_{\varepsilon,\omega} \mapsto \mathrm{vec}(\mathbf{S})$ can be written as a linear transformation of the input vector $\mathbf{A}_{\varepsilon,\omega} \mapsto \mathbf{U}\mathbf{A}_{\varepsilon,\omega}$. Based on this observation, the single-layer network

$$\Psi^1 := \big((\mathbf{U}, \mathbf{0}_{n_\ell^2})\big)$$

with input dimension m_ℓ and output dimension n_ℓ^2 exactly implements the desired map. Writing λ_i, $i = 1, \dots, n_\ell$, for the classical nodal basis of $V_h(\omega)$, and identifying the index of entry $i \in \{1, \dots, n_\ell^2\}$ in $\mathrm{vec}(\mathbf{S})$ with the index pair $(k,l) \in \{1, \dots, n_\ell\} \times \{1, \dots, n_\ell\}$ of the corresponding entry in $\mathbf{S}$, the matrix $\mathbf{U} \in \mathbb{R}^{n_\ell^2 \times m_\ell}$ is given by

$$\mathbf{U}_{i,j} = \int_{T_j} \nabla\lambda_k \cdot \nabla\lambda_l \, \mathrm{d}x,$$

where T_j is the jth element in $\mathcal{T}_\varepsilon(\omega)$. The fact that $\|\mathbf{U}\|_0 \le 2^{3d-1} m_\ell (\varepsilon/h)^d$ by a rough estimation then yields

(i) $L(\Psi^1) = 1$,
(ii) $M(\Psi^1) \le 2^{3d-1} m_\ell (\varepsilon/h)^d \lesssim ((\ell+1)H/h)^d \lesssim (\ell H/h)^d$.

Step 2. To approximate the inversion $\mathrm{vec}(\mathbf{S}) \mapsto \mathrm{vec}(\mathbf{S}^{-1})$, we utilize Theorem 3.10 to construct a suitable network Ψ^2. In order to do so, however, we first have to rescale the input $\mathrm{vec}(\mathbf{S})$ with a value $\mathfrak{v}_{\mathrm{resc}}$ in such a way that $\|\mathbf{Id}_{n_\ell^2} - \mathfrak{v}_{\mathrm{resc}}\mathbf{S}\|_2 \le 1 - \delta$ for some $\delta \in (0,1)$. Since all coefficients under consideration are bounded from below and above by α and β, respectively, there exist optimal values $\mathfrak{V}_{\mathrm{inf}}, \mathfrak{V}_{\mathrm{sup}}$ (dependent on h, α, and β) such that

$$0 < \mathfrak{V}_{\mathrm{inf}} \le \inf_{\mathbf{v} \in \mathbb{R}^{n_\ell} \setminus \{0\}} \frac{\mathbf{v}^T \mathbf{S} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \le \sup_{\mathbf{v} \in \mathbb{R}^{n_\ell} \setminus \{0\}} \frac{\mathbf{v}^T \mathbf{S} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \le \mathfrak{V}_{\mathrm{sup}} < \infty$$

for any $A \in \mathfrak{A}$. That is, the spectrum of $\mathbf{S}$ is always contained in $[\mathfrak{V}_{\mathrm{inf}}, \mathfrak{V}_{\mathrm{sup}}]$. Choosing

$$\mathfrak{v}_{\mathrm{resc}} \in (0, \mathfrak{V}_{\mathrm{sup}}^{-1}) \tag{A.2}$$

and setting $\delta := \mathfrak{v}_{\mathrm{resc}} \mathfrak{V}_{\mathrm{inf}}$, the symmetry of $\mathbf{S}$ then implies

$$\big\|\mathbf{Id}_{n_\ell^2} - \mathfrak{v}_{\mathrm{resc}}\mathbf{S}\big\|_2 \le |1 - \mathfrak{v}_{\mathrm{resc}} \mathfrak{V}_{\mathrm{inf}}| = 1 - \delta.$$

The step of rescaling the input corresponds to feeding it through the one-layer network $\big((\mathfrak{v}_{\mathrm{resc}} \mathbf{Id}_{n_\ell^2}, \mathbf{0}_{n_\ell^2})\big)$. Moreover, it can be easily seen that if $((\mathbf{W}, \mathbf{b}))$ is the neural network that implements $\mathfrak{v}_{\mathrm{resc}}\mathrm{vec}(\mathbf{S})$, then $((-\mathbf{W}, -\mathbf{b} + \mathrm{vec}(\mathbf{Id}_{n_\ell^2})))$ implements $\mathrm{vec}(\mathbf{Id}_{n_\ell^2} - \mathfrak{v}_{\mathrm{resc}}\mathbf{S})$. For any $\theta \in (0, \eta)$, Theorem 3.10 then guarantees the existence of a neural network $\Psi_{\mathrm{inv},\theta}^{1-\delta, n_\ell^2}$ such that

$$\Big\| (\mathfrak{v}_{\mathrm{resc}}\mathbf{S})^{-1} - \mathrm{mat}\Big(R_\rho^{K_{n_\ell}^{1-\delta}} \big(\Psi_{\mathrm{inv},\theta}^{1-\delta, n_\ell^2}\big) \Big(\mathrm{vec}(\mathbf{Id}_{n_\ell^2} - \mathfrak{v}_{\mathrm{resc}}\mathbf{S}) \Big) \Big) \Big\|_2 \le \theta. \tag{A.3}$$

Lastly, after the approximate inversion is performed, one has to scale the output back to the original scaling to obtain an approximation to $\mathrm{vec}(\mathbf{S}^{-1})$ rather than $(1/\mathfrak{v}_{\mathrm{resc}})\mathrm{vec}(\mathbf{S}^{-1})$. This is again done with the one-layer network $\big((\mathfrak{v}_{\mathrm{resc}} \mathbf{Id}_{n_\ell^2}, \mathbf{0}_{n_\ell^2})\big)$. Taking those operations together, we define

$$\Psi^2 := \big((\mathfrak{v}_{\mathrm{resc}} \mathbf{Id}_{n_\ell^2}, \mathbf{0}_{n_\ell^2})\big) \bullet \Psi_{\mathrm{inv},\eta}^{1-\delta, n_\ell^2} \odot \big((-\mathfrak{v}_{\mathrm{resc}} \mathbf{Id}_{n_\ell^2}, \mathrm{vec}(\mathbf{Id}_{n_\ell^2}))\big).$$

Combining Theorem 3.10 with Lemmas 3.4 and 3.7, we obtain the following bounds for the inversion network Ψ^2:

(i) $L(\Psi^2) \lesssim \log(m(\theta,\delta))\big(\log(1/\theta) + \log(m(\theta,\delta)) + \log(n_\ell)\big) + 1$,

(ii) $M(\Psi^2) \lesssim m(\theta,\delta)\log^2(m(\theta,\delta))n_\ell^3\big(\log(1/\theta) + \log(m(\theta,\delta)) + \log(n_\ell)\big) + n_\ell^2$.

Step 3*:* Similarly to step 1, multiplying the neural network approximation of $\mathbf{S}^{-1}$ with $\mathbf{I}_\omega^T$ from the right can be implemented by a single-layer neural network. Since both $\mathbf{S}^{-1}$ and its approximation are symmetric matrices (see Remark 3.11), it holds for both matrices that $\mathbf{S}^{-1}\mathbf{I}_\omega^T = (\mathbf{I}_\omega\mathbf{S}^{-1})^T$, and therefore, $\operatorname{vec}(\mathbf{S}^{-1}\mathbf{I}_\omega^T) = \mathbf{Q}\operatorname{vec}(\mathbf{I}_\omega\mathbf{S}^{-1})$ for a suitable permutation matrix $\mathbf{Q}$ that maps $\operatorname{vec}(\mathbf{M})$ to $\operatorname{vec}(\mathbf{M}^T)$ for an arbitrary matrix $\mathbf{M} \in \mathbb{R}^{N_\ell \times n_\ell}$. To implement the map $\operatorname{vec}(\mathbf{S}^{-1}) \mapsto \operatorname{vec}(\mathbf{I}_\omega\mathbf{S}^{-1})$, we consider the parallelization of n_ℓ identical copies $\Psi_1^3, \dots, \Psi_{n_\ell}^3$ of the single-layer network

$$\Psi_{\mathbf{I}_\omega} := \big((\mathbf{I}_\omega, \mathbf{0}_{N_\ell})\big).$$

Concatenating the output of this parallelization with the one-layer network $((\mathbf{Q}, \mathbf{0}_{n_\ell \cdot N_\ell}))$, i.e., setting

$$\Psi^3 := ((\mathbf{Q}, \mathbf{0}_{n_\ell \cdot N_\ell})) \bullet P(\Psi_1^3, \dots, \Psi_{n_\ell}^3),$$

then exactly implements the desired transformation. If the interpolation operator $\mathcal{I}_H$ given in (2.4) is used, the corresponding matrix $\mathbf{I}_\omega \in \mathbb{R}^{N_\ell \times n_\ell}$ is a sparse matrix that satisfies $\|\mathbf{I}_\omega\|_0 \le N_\ell(2H/h+1)^d$. Moreover, since $\mathbf{Q}$ is a permutation matrix, it does not change the number of nonzero parameters when concatenated with the parallelization network due to Lemma 3.4. With Lemma 3.9, we finally obtain the following complexity estimates:

(i) $L(\Psi^3) = 1$,

(ii) $M(\Psi^3) = n_\ell N_\ell(2H/h+1)^d = (((2\ell+1)H/h-1)(2\ell+2)(2H/h+1))^d$
$\lesssim (\ell^2 H/h(H/h+1))^d \lesssim (\ell H/h)^{2d}$.

Step 4. Analogous to the previous step, we consider N_ℓ identical copies $\Psi_1^4, \dots, \Psi_{N_\ell}^4$ of the linear transformation network $\Psi_{\mathbf{I}_\omega} = \big((\mathbf{I}_\omega, \mathbf{0}_{N_\ell})\big)$ that implements multiplication of an input vector with $\mathbf{I}_\omega$. Then define the parallelization

$$\Psi^4 := P(\Psi_1^4, \dots, \Psi_{N_\ell}^4),$$

which implements the desired transformation exactly. Note that in this step, no permutation of the output of the parallelization is necessary. We obtain the same bounds as in the previous step, i.e.,

(i) $L(\Psi^4) = 1$,

(ii) $M(\Psi^4) \lesssim (((2\ell+1)H/h-1)(2\ell+2)(2H/h+1))^d \lesssim (\ell H/h)^{2d}$.

Step 5. This step is similar to step 2 and utilizes again Theorem 3.10 to approximate the inversion of $\mathbf{Y} := \mathbf{I}_\omega\mathbf{S}^{-1}\mathbf{I}_\omega^T \in \mathbb{R}^{N_\ell \times N_\ell}$. However, we have to consider that at this stage the approximation

$$\widehat{\mathbf{Y}} := \mathbf{I}_\omega \mathfrak{v}_{\mathrm{resc}} \operatorname{mat}\Big(R_\rho^{K_{n_\ell}^{1-\delta}}\big(\Psi_{\mathrm{inv},\theta}^{1-\delta,n_\ell^2}\big)(\operatorname{vec}(\mathbf{Id}_{n_\ell^2} - \mathfrak{v}_{\mathrm{resc}}\mathbf{S}))\Big)\mathbf{I}_\omega^T$$

instead of the true matrix $\mathbf{Y}$ will be given as the input to the network to be constructed due to the inexact inversion of step 2 above. Observe that under the additional condition

$$\theta < \min\left\{\lambda_{\min}(\mathbf{Y}), \frac{\lambda_{\min}(\mathbf{Y})}{2\,\mathfrak{v}_{\mathrm{resc}}\|\mathbf{I}_\omega\|_2^2}\right\},$$

it holds true that

$$|\lambda_{\min}(\mathbf{Y}) - \lambda_{\min}(\widehat{\mathbf{Y}})| \leq \|\mathbf{Y} - \widehat{\mathbf{Y}}\|_2 \leq \mathfrak{v}_{\mathrm{resc}} \|\mathbf{I}_\omega\|_2^2 \, \theta < 0.5 \, \lambda_{\min}(\mathbf{Y}),$$

and therefore,

$$\lambda_{\min}(\widehat{\mathbf{Y}}) = \lambda_{\min}(\mathbf{Y}) - \big(\lambda_{\min}(\mathbf{Y}) - \lambda_{\min}(\widehat{\mathbf{Y}})\big) > 0.5 \, \lambda_{\min}(\mathbf{Y}), \tag{A.4}$$

i.e., the minimal eigenvalue of $\widehat{\mathbf{Y}}$ can be bounded in terms of the minimal eigenvalue of $\mathbf{Y}$. This and the fact that $\mathbf{Y}$ is a symmetric and positive definite matrix (cf. also the proof of Corollary 3.13) implies that $\widehat{\mathbf{Y}}$ is symmetric and positive definite as well, since the neural network approximating matrix inversion can always be constructed such that symmetry of the input matrix is preserved (cf. Remark 3.11). Further, we have

$$\lambda_{\max}(\widehat{\mathbf{Y}}) = \|\widehat{\mathbf{Y}}\|_2 \leq \|\mathbf{Y}\|_2 + \|\mathbf{Y} - \widehat{\mathbf{Y}}\|_2 \leq \lambda_{\max}(\mathbf{Y}) + 0.5 \, \lambda_{\min}(\mathbf{Y}) < 2 \, \lambda_{\max}(\mathbf{Y}). \tag{A.5}$$

This implies the existence of optimal values $\widehat{\mathfrak{V}}_{\mathrm{inf}}$, $\widehat{\mathfrak{V}}_{\mathrm{sup}}$ (dependent on H, h, α, and β) such that

$$0 < \widehat{\mathfrak{V}}_{\mathrm{inf}} \leq \inf_{\mathbf{v} \in \mathbb{R}^{N_\ell} \setminus \{0\}} \frac{\mathbf{v}^T \widehat{\mathbf{Y}} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \leq \sup_{\mathbf{v} \in \mathbb{R}^{N_\ell} \setminus \{0\}} \frac{\mathbf{v}^T \widehat{\mathbf{Y}} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \leq \widehat{\mathfrak{V}}_{\mathrm{sup}} < \infty \tag{A.6}$$

and

$$0 < \widehat{\mathfrak{V}}_{\mathrm{inf}} \leq \inf_{\mathbf{v} \in \mathbb{R}^{N_\ell} \setminus \{0\}} \frac{\mathbf{v}^T \mathbf{Y} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \leq \sup_{\mathbf{v} \in \mathbb{R}^{N_\ell} \setminus \{0\}} \frac{\mathbf{v}^T \mathbf{Y} \mathbf{v}}{\mathbf{v}^T \mathbf{v}} \leq \widehat{\mathfrak{V}}_{\mathrm{sup}} < \infty. \tag{A.7}$$

For a discussion on the scaling of the values $\widehat{\mathfrak{V}}_{\mathrm{inf}}$, $\widehat{\mathfrak{V}}_{\mathrm{sup}}$ and the assumptions on θ in terms of the mesh sizes H and h, we refer to the proof of Corollary 3.13 below. When the matrices $\mathbf{Y}$ and $\widehat{\mathbf{Y}}$ are rescaled with $\widehat{\mathfrak{v}}_{\mathrm{resc}} \in (0, \widehat{\mathfrak{V}}_{\mathrm{sup}}^{-1})$, it holds that

$$\|\mathbf{Id}_{N_\ell^2} - \widehat{\mathfrak{v}}_{\mathrm{resc}} \mathbf{Y}\|_2 \leq 1 - \widehat{\delta}, \quad \text{as well as} \quad \|\mathbf{Id}_{N_\ell^2} - \widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathbf{Y}}\|_2 \leq 1 - \widehat{\delta},$$

with $\widehat{\delta} := \widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathfrak{V}}_{\mathrm{inf}}$. The rescaling is implemented by the network $\big((\widehat{\mathfrak{v}}_{\mathrm{resc}} \mathbf{Id}_{N_\ell^2}, \mathbf{0}_{N_\ell^2})\big)$. Moreover, changing this network to $\big((-\widehat{\mathfrak{v}}_{\mathrm{resc}} \mathbf{Id}_{N_\ell^2}, \mathrm{vec}(\mathbf{Id}_{N_\ell^2}))\big)$ by switching signs of the weight matrix and adding a vectorized identity matrix as a bias term leads to the implementation of $(\mathbf{Id}_{N_\ell^2} - \widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathbf{Y}})$ if $\mathrm{vec}(\widehat{\mathbf{Y}})$ is fed through this network. Theorem 3.10 yields the existence of a neural network $\Psi_{\mathrm{inv},\gamma}^{1-\widehat{\delta}, N_\ell^2}$ such that

$$\Big\| \big(\widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathbf{Y}}\big)^{-1} - \mathrm{mat}\Big(R_\rho^{K_{N_\ell}^{1-\widehat{\delta}}} \big(\Psi_{\mathrm{inv},\gamma}^{1-\widehat{\delta}, N_\ell^2}\big)\big(\mathrm{vec}(\mathbf{Id}_{N_\ell^2} - \widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathbf{Y}})\big)\Big) \Big\|_2 \leq \gamma. \tag{A.8}$$

Note that this, in turn, implies that

$$\Big\| \big(\widehat{\mathfrak{v}}_{\mathrm{resc}} \mathbf{Y}\big)^{-1} - \mathrm{mat}\Big(R_\rho^{K_{N_\ell}^{1-\widehat{\delta}}} \big(\Psi_{\mathrm{inv},\eta}^{1-\widehat{\delta}, N_\ell^2}\big)\big(\mathrm{vec}(\mathbf{Id}_{N_\ell^2} - \widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathbf{Y}})\big)\Big) \Big\|_2 \leq \frac{\mathfrak{v}_{\mathrm{resc}} \|\mathbf{I}_\omega\|_2^2}{\widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathfrak{V}}_{\mathrm{inf}}^2} \, \theta + \gamma. \tag{A.9}$$

Indeed, it holds by the triangle inequality and (A.8) that

$$\begin{aligned}
&\Big\| \big(\widehat{\mathfrak{v}}_{\mathrm{resc}} \mathbf{Y}\big)^{-1} - \mathrm{mat}\Big(R_\rho^{K_{N_\ell}^{1-\widehat{\delta}}} \big(\Psi_{\mathrm{inv},\eta}^{1-\widehat{\delta}, N_\ell^2}\big)\big(\mathrm{vec}(\mathbf{Id}_{N_\ell^2} - \widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathbf{Y}})\big)\Big) \Big\|_2 \\
&\quad \leq \Big\| \big(\widehat{\mathfrak{v}}_{\mathrm{resc}} \mathbf{Y}\big)^{-1} - \big(\widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathbf{Y}}\big)^{-1} \Big\|_2 \\
&\qquad + \Big\| \big(\widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathbf{Y}}\big)^{-1} - \mathrm{mat}\Big(R_\rho^{K_{N_\ell}^{1-\widehat{\delta}}} \big(\Psi_{\mathrm{inv},\eta}^{1-\widehat{\delta}, N_\ell^2}\big)\big(\mathrm{vec}(\mathbf{Id}_{N_\ell^2} - \widehat{\mathfrak{v}}_{\mathrm{resc}} \widehat{\mathbf{Y}})\big)\Big) \Big\|_2 =: (\star) + \gamma.
\end{aligned}$$

Using $\|\mathbf{A}^{-1}-\mathbf{B}^{-1}\|_2 \le \|\mathbf{A}-\mathbf{B}\|_2\|\mathbf{A}^{-1}\|_2\|\mathbf{B}^{-1}\|_2$, the term $(\star)$ can be estimated by

$$(\star) \le \left\| \widehat{\mathfrak{v}}_{\mathrm{resc}}\mathbf{Y} - \widehat{\mathfrak{v}}_{\mathrm{resc}}\widehat{\mathbf{Y}} \right\|_2 \left\| \left(\widehat{\mathfrak{v}}_{\mathrm{resc}}\mathbf{Y}\right)^{-1} \right\|_2 \left\| \left(\widehat{\mathfrak{v}}_{\mathrm{resc}}\widehat{\mathbf{Y}}\right)^{-1} \right\|_2 \le \frac{1}{\widehat{\mathfrak{v}}_{\mathrm{resc}}} \|\mathbf{I}_\omega\|_2^2\, \mathfrak{v}_{\mathrm{resc}}\, \theta\, \frac{1}{\widehat{\mathfrak{V}}_{\mathrm{inf}}}\, \frac{1}{\widehat{\mathfrak{V}}_{\mathrm{inf}}},$$

where we have used (A.3) and the definitions as well as the spectral bounds given on $\mathbf{Y}, \widehat{\mathbf{Y}}$ in (A.6), (A.7). This yields the inequality (A.9).

Rescaling the output of $\Psi_{\mathrm{inv},\gamma}^{1-\widehat{\delta},N_\ell^2}$ again, we get a network with the desired approximation properties defined by

$$\Psi^5 := \left((\widehat{\mathfrak{v}}_{\mathrm{resc}}\,\mathbf{Id}_{N_\ell^2}, \mathbf{0}_{N_\ell^2})\right) \bullet \Psi_{\mathrm{inv},\gamma}^{1-\widehat{\delta},N_\ell^2} \odot \left((-\widehat{\mathfrak{v}}_{\mathrm{resc}}\,\mathbf{Id}_{N_\ell^2}, \mathrm{vec}(\mathbf{Id}_{N_\ell^2}))\right).$$

Further, we have the following bounds on depth and number of parameters:

(i) $L(\Psi^5) \lesssim \log(m(\gamma,\widehat{\delta}))\,(\log(1/\gamma)+\log(m(\gamma,\widehat{\delta}))+\log(N_\ell))+1$,

(ii) $M(\Psi^5) \lesssim m(\gamma,\widehat{\delta})\log^2(m(\gamma,\widehat{\delta}))N_\ell^3\,(\log(1/\gamma)+\log(m(\gamma,\widehat{\delta}))+\log(N_\ell))+N_\ell^2$.

Step 6. The multiplication of $\mathbf{Y}^{-1}$, respectively $\widehat{\mathbf{Y}}^{-1}$, with $\mathbf{I}_\omega\mathbf{P}_{\omega,K}$ from the right is analogous to step 3. Again, we have that both $\mathbf{Y}^{-1}$ and $\widehat{\mathbf{Y}}^{-1}$ are symmetric and thus the expression $\mathbf{Y}^{-1}\mathbf{I}_\omega\mathbf{P}_{\omega,K}$ can also be written as $((\mathbf{I}_\omega\mathbf{P}_{\omega,K})^T\mathbf{Y}^{-1})^T$. With the same argument as in step 3, the map $\mathrm{vec}(\mathbf{Y}^{-1}) \mapsto \mathrm{vec}(\mathbf{Y}^{-1}\mathbf{I}_\omega\mathbf{P}_{\omega,K})$ is thus exactly implemented by

$$\Psi^6 := ((\widetilde{\mathbf{Q}}, \mathbf{0}_{2^dN_\ell})) \bullet P(\Psi_1^6, \dots, \Psi_{N_\ell}^6),$$

where $\widetilde{\mathbf{Q}}$ is a suitable permutation matrix of dimension $N_\ell 2^d \times N_\ell 2^d$ and $\Psi_1^6, \dots, \Psi_{N_\ell}^6$ are identical copies of

$$\Psi_{(\mathbf{I}_\omega\mathbf{P}_{\omega,K})^T} := \left((\mathbf{I}_\omega\mathbf{P}_{\omega,K})^T, \mathbf{0}_{2^d}\right).$$

Since $\mathbf{I}_\omega$ is a quasi-interpolation operator on ω and $\mathbf{P}_{\omega,K}$ is the prolongation operator from the element K to the whole patch, we roughly estimate $\|\mathbf{I}_\omega\mathbf{P}_{\omega,K}\|_0 \le 6^d$. Combining this with Lemma 3.9 yields

(i) $L(\Psi^6) = 1$,

(ii) $M(\Psi^6) = 6^dN_\ell = 6^d(2\ell+2)^d \lesssim \ell^d$.

Step 7. Analogous to step 4, we consider the parallelization

$$\Psi^7 := P(\Psi_1^7, \dots, \Psi_{N_\ell}^7),$$

where $\Psi_1^7, \dots, \Psi_{N_\ell}^7$ are identical copies of

$$\Psi_{\mathbf{P}_\omega^T\mathbf{I}_\omega^T} := \left((\mathbf{P}_\omega^T\mathbf{I}_\omega^T, \mathbf{0}_{2^d})\right).$$

Again, no permutation is necessary in this step. With our choice of the quasi-interpolation operator, the rough estimate $\|\mathbf{P}_\omega^T\mathbf{I}_\omega^T\|_0 \le 3^dN_\ell$ holds, which leads to

(i) $L(\Psi^7) = 1$,

(ii) $M(\Psi^7) \le 3^dN_\ell^2 = 3^d(2\ell+2)^{2d} \lesssim \ell^{2d}$.

By connecting the individual networks $\Psi^1, \dots, \Psi^7$ in series by sparsely concatenating them, i.e., defining

$$\Psi_\eta^{\mathrm{pg}} := \Psi^7 \odot \cdots \odot \Psi^1,$$

we finally obtain a network with the desired properties. In particular, we have

$$\left\|\mathbf{S}_\omega^{\mathrm{pg}} - \mathrm{mat}(\mathcal{R}_\rho^{\mathfrak{A}_{\varepsilon,\omega}}(\Psi_\eta^{\mathrm{pg}}(\mathbf{A}_{\varepsilon,\omega})))\right\|_2 \le \frac{\mathfrak{v}_{\mathrm{resc}}\,\|\mathbf{P}_{\omega,K}\|_2\,\|\mathbf{P}_\omega\|_2\,\|\mathbf{I}_\omega\|_2^4}{\widehat{\mathfrak{V}}_{\mathrm{inf}}^2}\,\theta + \|\mathbf{P}_{\omega,K}\|_2\,\|\mathbf{P}_\omega\|_2\,\|\mathbf{I}_\omega\|_2^2\,\gamma,$$

since with the abbreviation

$$\widehat{\mathrm{inv}}(\widehat{\mathfrak{v}}_{\mathrm{resc}}\widehat{\mathbf{Y}}) := \mathrm{mat}\Big(R_\rho^{K_{N_\ell}^{1-\widehat{\delta}}}\big(\Psi_{\mathrm{inv},\eta}^{1-\widehat{\delta},N_\ell^2}\big)(\mathrm{vec}(\mathbf{Id}_{N_\ell^2} - \widehat{\mathfrak{v}}_{\mathrm{resc}}\widehat{\mathbf{Y}}))\Big),$$

it holds that

$$\begin{aligned}
&\Big\|\mathbf{S}_\omega^{\mathrm{pg}} - \mathrm{mat}(\mathcal{R}_\rho^{\mathfrak{A}_{\varepsilon,\omega}}(\Psi_\eta^{\mathrm{pg}}(\mathbf{A}_{\varepsilon,\omega})))\Big\|_2 \\
&\quad = \Big\| \mathbf{I}_\omega\mathbf{P}_\omega\widehat{\mathfrak{v}}_{\mathrm{resc}}\left(\widehat{\mathfrak{v}}_{\mathrm{resc}}\mathbf{Y}\right)^{-1}\mathbf{I}_\omega^T\mathbf{P}_{K,\omega} - \mathbf{I}_\omega\mathbf{P}_\omega\widehat{\mathfrak{v}}_{\mathrm{resc}}\widehat{\mathrm{inv}}(\widehat{\mathfrak{v}}_{\mathrm{resc}}\widehat{\mathbf{Y}})\mathbf{I}_\omega^T\mathbf{P}_{K,\omega}\Big\|_2 \\
&\quad \le \|\mathbf{P}_\omega\|_2\,\|\mathbf{P}_{K,\omega}\|_2\,\|\mathbf{I}_\omega\|_2^2\,\widehat{\mathfrak{v}}_{\mathrm{resc}}\left(\frac{\mathfrak{v}_{\mathrm{resc}}\,\|\mathbf{I}_\omega\|_2^2}{\widehat{\mathfrak{v}}_{\mathrm{resc}}\widehat{\mathfrak{V}}_{\mathrm{inf}}^2}\,\theta + \gamma\right)
\end{aligned}$$

due to (A.9). Choosing θ, γ such that

$$\frac{\mathfrak{v}_{\mathrm{resc}}\,\|\mathbf{P}_{\omega,K}\|_2\,\|\mathbf{P}_\omega\|_2\,\|\mathbf{I}_\omega\|_2^4}{\widehat{\mathfrak{V}}_{\mathrm{inf}}^2}\,\theta + \|\mathbf{P}_{\omega,K}\|_2\,\|\mathbf{P}_\omega\|_2\,\|\mathbf{I}_\omega\|_2^2\,\gamma \le \eta$$

then yields the estimate. As above, we refer to the proof of Corollary 3.13 below for a discussion of the scaling of the relevant spectral norms in terms of the mesh sizes H, h. Using Lemma 3.7, we obtain the following bounds for the final network:

(i) $L(\Psi^{\mathrm{pg}}) \le \sum_{i=1}^7 L(\Psi^i) \lesssim \log(m(\theta,\delta))\big(\log(1/\theta) + \log(m(\theta,\delta)) + \log(n_\ell)\big)$
$+ \log(m(\gamma,\widehat{\delta}))\big(\log(1/\gamma) + \log(m(\gamma,\widehat{\delta})) + \log(N_\ell)\big) + 1,$

(ii) $M(\Psi^{\mathrm{pg}}) \lesssim \sum_{i=1}^7 M(\Psi^i)$
$\lesssim m(\theta,\delta)\log^2(m(\theta,\delta))n_\ell^3\big(\log(1/\theta) + \log(m(\theta,\delta)) + \log(n_\ell)\big)$
$+ m(\gamma,\widehat{\delta})\log^2(m(\gamma,\widehat{\delta}))N_\ell^3\big(\log(1/\gamma) + \log(m(\gamma,\widehat{\delta})) + \log(N_\ell)\big) + (\ell H/h)^{2d}.$

This is the assertion.

A.2. Proof of Corollary 3.13. The claim follows directly from Theorem 3.12 with $\eta \approx H^k$ and the estimation of the quantities $\theta, \gamma, \delta, \widehat{\delta}$ in terms of the mesh sizes H, h. Starting with θ and γ, we have the condition that

$$\theta < \min\left\{H^k, \lambda_{\min}(\mathbf{I}_\omega\mathbf{S}^{-1}\mathbf{I}_\omega^T), \frac{\lambda_{\min}(\mathbf{I}_\omega\mathbf{S}^{-1}\mathbf{I}_\omega^T)}{2\,\mathfrak{v}_{\mathrm{resc}}\|\mathbf{I}_\omega\|_2^2}\right\}, \tag{A.10}$$

which requires an estimation of the smallest eigenvalue of $\mathbf{I}_\omega\mathbf{S}^{-1}\mathbf{I}_\omega^T$. Observe that

$$\lambda_{\min}(\mathbf{I}_\omega\mathbf{S}^{-1}\mathbf{I}_\omega^T) \ge \lambda_{\min}(\mathbf{S}^{-1})\lambda_{\min}(\mathbf{I}_\omega\mathbf{I}_\omega^T) = \frac{\lambda_{\min}(\mathbf{I}_\omega\mathbf{I}_\omega^T)}{\lambda_{\max}(\mathbf{S})} \gtrsim \frac{\lambda_{\min}(\mathbf{I}_\omega\mathbf{I}_\omega^T)}{h^{d-2}},$$

where we have used the well-known fact that $\lambda_{\max}(\mathbf{S}) \approx h^{d-2}$ as the maximal eigenvalue of a finite element stiffness matrix. In order to estimate the eigenvalues of $\mathbf{I}_\omega\mathbf{I}_\omega^T$, we have to look into the practical computation of the matrix $\mathbf{I}_\omega$. It holds that

$$\mathbf{I}_\omega = \mathbf{R}_{\mathrm{c},\omega}\mathbf{E}_{\mathrm{c},\omega}(\mathbf{M}_{\mathrm{c},\omega}^{\mathrm{dg}})^{-1}(\mathbf{P}_\omega^{\mathrm{dg}})^T\mathbf{M}_\omega^{\mathrm{dg}}\mathbf{C}_\omega\mathbf{R}_\omega^T,$$

where $\mathbf{E}_{\mathrm{c},\omega}$ is the algebraic realization of the averaging operator E_H introduced in (2.4), $\mathbf{M}_{\mathrm{c},\omega}^{\mathrm{dg}}$ and $\mathbf{M}_\omega^{\mathrm{dg}}$ are the mass matrices corresponding to $Q1$ discontinuous finite element functions on $\mathcal{T}_H(\omega)$ and $\mathcal{T}_h(\omega)$, respectively, $\mathbf{P}_\omega^{\mathrm{dg}}$ is the prolongation map from $\mathcal{T}_H(\omega)$ to $\mathcal{T}_h(\omega)$ for $Q1$ discontinuous finite element functions, $\mathbf{C}_\omega$ maps the vector representation of a continuous finite element function to the vector of its discontinuous representation, and $\mathbf{R}_\omega$ is a restriction operator that removes all entries corresponding

to boundary nodes of the patch ω on the fine mesh. The restriction operator $\mathbf{R}_{\mathrm{c},\omega}$ removes coarse boundary nodes on ∂D only. Since $\mathbf{I}_\omega$ has full rank, it holds that

$$\lambda_{\min}(\mathbf{I}_\omega\mathbf{I}_\omega^T) \geq \lambda_{\min}(\mathbf{R}_{\mathrm{c},\omega}\mathbf{R}_{\mathrm{c},\omega}^T)\,\lambda_{\min}(\mathbf{E}_{\mathrm{c},\omega}\mathbf{E}_{\mathrm{c},\omega}^T)\,\lambda_{\min}((\mathbf{M}_{\mathrm{c},\omega}^{\mathrm{dg}})^{-2})\,\lambda_{\min}((\mathbf{P}_\omega^{\mathrm{dg}})^T\mathbf{P}_\omega^{\mathrm{dg}}) \cdot \lambda_{\min}((\mathbf{M}_\omega^{\mathrm{dg}})^2)\,\lambda_{\min}(\mathbf{C}_\omega^T\mathbf{C}_\omega)\,\lambda_{\min}(\mathbf{R}_\omega\mathbf{R}_\omega^T).$$

Using that $\lambda_{\min}(\mathbf{R}_{\mathrm{c},\omega}\mathbf{R}_{\mathrm{c},\omega}^T) = \lambda_{\min}(\mathbf{R}_\omega\mathbf{R}_\omega^T) = 1$, $\lambda_{\min}(\mathbf{E}_{\mathrm{c},\omega}\mathbf{E}_{\mathrm{c},\omega}^T) \gtrsim 2^{-d}$, $\lambda_{\min}((\mathbf{M}_{\mathrm{c},\omega}^{\mathrm{dg}})^{-2}) \gtrsim H^{-2d}$, $\lambda_{\min}((\mathbf{M}_\omega^{\mathrm{dg}})^2) \gtrsim h^{2d}$, $\lambda_{\min}((\mathbf{P}_\omega^{\mathrm{dg}})^T\mathbf{P}_\omega^{\mathrm{dg}}) \gtrsim (H/h)^d$ and $\lambda_{\min}(\mathbf{C}_\omega^T\mathbf{C}_\omega) = 1$, we obtain

$$\lambda_{\min}(\mathbf{I}_\omega\mathbf{I}_\omega^T) \gtrsim 2^{-d}H^{-2d}h^{2d}(H/h)^d \gtrsim (h/H)^d.$$

This, in turn, yields

$$\lambda_{\min}(\mathbf{I}_\omega\mathbf{S}^{-1}\mathbf{I}_\omega^T) \gtrsim \frac{(h/H)^d}{h^{d-2}} \approx \frac{h^2}{H^d}.$$

With a similar argument for the maximal eigenvalue, we also get $\|\mathbf{I}_\omega\|_2^2 \lesssim (h/H)^d$. Further, we have

$$\mathfrak{v}_{\mathrm{resc}} < \frac{1}{\mathfrak{V}_{\mathrm{sup}}} \leq \frac{1}{\lambda_{\max}(\mathbf{S})} \approx h^{2-d}.$$

Combining the above estimates, we arrive at

$$\frac{\lambda_{\min}(\mathbf{I}_\omega\mathbf{S}^{-1}\mathbf{I}_\omega^T)}{\mathfrak{v}_{\mathrm{resc}}\|\mathbf{I}_\omega\|_2^2} \gtrsim \frac{h^2/H^d}{h^{2-d}(h/H)^d} \approx 1.$$

Recall that according to Theorem 3.12, the parameter γ has to satisfy the condition

$$\frac{\mathfrak{v}_{\mathrm{resc}}\,\|\mathbf{P}_{\omega,K}\|_2\,\|\mathbf{P}_\omega\|_2\,\|\mathbf{I}_\omega\|_2^4}{\widehat{\mathfrak{V}}_{\mathrm{inf}}}\,\theta + \|\mathbf{P}_{\omega,K}\|_2\,\|\mathbf{P}_\omega\|_2\,\|\mathbf{I}_\omega\|_2^2\,\gamma \leq H^k.$$

Since $\mathbf{P}_\omega$ and $\mathbf{P}_{\omega,K}$ realize prolongations, we have $\|\mathbf{P}_\omega\|_2 \lesssim (H/h)^{d/2}$ as well as $\|\mathbf{P}_{\omega,K}\|_2 \lesssim (H/h)^{d/2}$. With a suitable choice of θ, i.e.,

$$\theta \approx \frac{h^2}{H}H^k\left(1+\frac{H^d}{h^2}\right)^{-1},$$

the estimate (A.10) is fulfilled. Further, we have with (A.4) that

$$\widehat{\mathfrak{V}}_{\mathrm{inf}} \approx \lambda_{\min}(\mathbf{I}_\omega\mathbf{S}^{-1}\mathbf{I}_\omega^T) \gtrsim \frac{h^2}{H^d}.$$

Choosing $\gamma \approx \theta$ thus leads to the rough estimate

$$\begin{aligned}\frac{\mathfrak{v}_{\mathrm{resc}}\,\|\mathbf{P}_{\omega,K}\|_2\,\|\mathbf{P}_\omega\|_2\,\|\mathbf{I}_\omega\|_2^4}{\widehat{\mathfrak{V}}_{\mathrm{inf}}^2}\,\theta + \|\mathbf{P}_{\omega,K}\|_2\,\|\mathbf{P}_\omega\|_2\,\|\mathbf{I}_\omega\|_2^2\,\gamma \\ \lesssim \frac{h^{2-d}(H/h)^d(h/H)^{2d}}{(h^4/H^{2d})}\theta + (H/h)^d(h/H)^d\gamma \\ \approx \left(1+\frac{H^d}{h^2}\right)\theta \lesssim \left(1+\frac{H^d}{h^2}\right)\left(1+\frac{H^d}{h^2}\right)^{-1}H^k = H^k.\end{aligned}$$

In order to estimate the quantities $m(\theta,\delta)$ and $m(\gamma,\widehat{\delta})$, we derive lower bounds on δ, $\widehat{\delta}$, and θ. If $\mathfrak{v}_{\mathrm{resc}}$ is chosen close to $\mathfrak{V}_{\mathrm{sup}}^{-1}$, we have that

$$\delta = \mathfrak{v}_{\mathrm{resc}}\mathfrak{V}_{\mathrm{inf}} \approx \frac{\mathfrak{V}_{\mathrm{inf}}}{\mathfrak{V}_{\mathrm{sup}}} \approx \frac{\lambda_{\min}(\mathbf{S})}{\lambda_{\max}(\mathbf{S})} \approx \frac{h^d}{h^{d-2}} = h^2,$$

using the fact that $\lambda_{\min}(\mathbf{S}) \approx h^d$. Analogously, for a choice of $\widehat{\mathfrak{v}}_{\mathrm{resc}}$ close to $\widehat{\mathfrak{V}}_{\mathrm{sup}}^{-1}$ we obtain

$$\widehat{\delta} \approx \frac{\widehat{\mathfrak{V}}_{\mathrm{inf}}}{\widehat{\mathfrak{V}}_{\mathrm{sup}}} \approx \frac{\lambda_{\min}(\mathbf{I}_\omega \mathbf{S}^{-1}\mathbf{I}_\omega^T)}{\lambda_{\max}(\mathbf{I}_\omega \mathbf{S}^{-1}\mathbf{I}_\omega^T)} \gtrsim \frac{h^2 H^{-d}}{H^{-d}} \approx h^2,$$

where we have used that $\lambda_{\max}(\mathbf{I}_\omega \mathbf{S}^{-1}\mathbf{I}_\omega^T) \lesssim H^{-d}$, which can be derived analogously to the scaling of the smallest eigenvalue. For θ (and thus γ), a very rough estimation yields

$$\theta \approx \frac{h^2}{H} H^k \left(1 + \frac{H^d}{h^2}\right)^{-1} \approx H^{k-1}h^2 \frac{h^2}{h^2 + H^d} \gtrsim H^{k-1}h^4.$$

Therefore, we obtain

$$m(\theta,\delta) = \left\lceil \frac{\log(0.5\theta\delta)}{\log(1-\delta)} \right\rceil \lesssim \frac{\log(0.5H^{k-1}h^6)}{\log(1-h^2)} \lesssim \frac{(k-1)|\log(H)| + 6|\log(h)|}{h^2},$$

and similarly,

$$m(\gamma,\widehat{\delta}) \lesssim \frac{(k-1)|\log(H)| + 6|\log(h)|}{h^2}.$$

Moreover, it holds that $N_\ell < n_\ell \lesssim (|\log(H)|H/h)^d$ and thus

$$\log(n_\ell) \lesssim d\log(H|\log(H)|/h) \lesssim |\log(h)|.$$

Inserting all these results into the bounds derived in Theorem 3.10 and using without loss of generality that $\theta \le \gamma$ and $\delta \le \widehat{\delta}$, we obtain

$$\begin{aligned} L(\Psi_{H^k}^{\mathrm{pg}}) &\lesssim \log(m(\theta,\delta))\Big(\log(1/\theta) + \log(m(\theta,\delta)) + \log(n_\ell)\Big) \\ &\lesssim \Big(|\log(h)| + |\log(k)|\Big)\Big(|\log(h)| + |\log(k)|\Big) \\ &\lesssim |\log(h)|^2 + |\log(k)|^2 \end{aligned}$$

as well as

$$\begin{aligned} M(\Psi_{H^k}^{\mathrm{pg}}) &\lesssim m(\theta,\delta)\log^2(m(\theta,\delta))\, n_\ell^3 \Big(\log(1/\theta) + \log(m(\theta,\delta)) + \log(n_\ell)\Big) \\ &\quad + (|\log(H)|H/h)^{2d} \\ &\lesssim h^{-2}\Big(k|\log(H)| + |\log(h)|\Big)\Big(|\log(h)|^3 + |\log(k)|^3\Big)\big(|\log(H)|H/h\big)^{3d}. \end{aligned}$$

This is the assertion.

Appendix B. Proofs in section 4.

B.1. Proof of Lemma 4.1. Taking the difference of (2.7) and (2.8) leads to

$$a((\mathsf{id}-\mathcal{Q}^\ell)(u_H^{\mathrm{c}}-u_H^{\mathrm{pg}}),(\mathsf{id}-\mathcal{Q}^\ell)v_H)=a((\mathsf{id}-\mathcal{Q}^\ell)u_H^{\mathrm{pg}},\mathcal{Q}^\ell v_H)$$

for all $v_H\in V_H$. Therefore, with $v_H=(u_H^{\mathrm{c}}-u_H^{\mathrm{pg}})$, we have

$$\begin{aligned}
&a((\mathsf{id}-\mathcal{Q}^\ell)(u_H^{\mathrm{c}}-u_H^{\mathrm{pg}}),(\mathsf{id}-\mathcal{Q}^\ell)(u_H^{\mathrm{c}}-u_H^{\mathrm{pg}}))\\
&\quad=a((\mathsf{id}-\mathcal{Q}^\ell)u_H^{\mathrm{pg}},\mathcal{Q}^\ell(u_H^{\mathrm{c}}-u_H^{\mathrm{pg}}))\\
&\quad=\underbrace{a((\mathsf{id}-\mathcal{Q})u_H^{\mathrm{pg}},\mathcal{Q}^\ell(u_H^{\mathrm{c}}-u_H^{\mathrm{pg}}))}_{=0}+a((\mathcal{Q}-\mathcal{Q}^\ell)u_H^{\mathrm{pg}},\mathcal{Q}^\ell(u_H^{\mathrm{c}}-u_H^{\mathrm{pg}}))\\
&\quad\lesssim\exp(-c\ell)\|\nabla u_H^{\mathrm{pg}}\|_{L^2(D)}\,\|(\mathsf{id}-\mathcal{Q}^\ell)(u_H^{\mathrm{c}}-u_H^{\mathrm{pg}})\|_{L^2(D)}
\end{aligned}$$

using (2.11) in the last step. This leads to

$$\|\nabla(\mathsf{id}-\mathcal{Q}^\ell)(u_H^{\mathrm{c}}-u_H^{\mathrm{pg}})\|_{L^2(D)}\lesssim\exp(-c\ell)\|\nabla u_H^{\mathrm{pg}}\|_{L^2(D)}$$

and, with the Friedrichs inequality, interpolation estimates, and the stability of the solution u_H^{pg}, we get

$$\|u_H^{\mathrm{c}}-u_H^{\mathrm{pg}}\|_{L^2(D)}\lesssim\exp(-c\ell).$$

Further, we have

$$\|\mathbf{u}^{\mathrm{c}}-\mathbf{u}^{\mathrm{pg}}\|_2\le CH^{-d/2}\|u_H^{\mathrm{c}}-u_H^{\mathrm{pg}}\|_{L^2(D)}\lesssim H^{-d/2}\exp(-c\ell),$$

which is the assertion.

B.2. Proof of Lemma 4.2. Due to the definition of (2.7) (which defines $\mathbf{S}^{\mathrm{c}}$), we have for any vector $\mathbf{v}$ with corresponding function $v\in V_H$ that

$$\begin{aligned}
\mathbf{v}^T\mathbf{S}^{\mathrm{c}}\mathbf{v}&=a((\mathsf{id}-\mathcal{Q}^\ell)v,(\mathsf{id}-\mathcal{Q}^\ell)v)\ge\alpha\|\nabla(\mathsf{id}-\mathcal{Q}^\ell)v\|_{L^2(D)}^2\\
&\ge\alpha C^{-2}\|\nabla v\|_{L^2(D)}^2
\end{aligned}\tag{B.1}$$

using the interpolation estimate (2.3) and $\mathcal{I}_H(\mathsf{id}-\mathcal{Q}^\ell)v=v$. With the Poincaré–Friedrichs inequality with constant C_{P} and the estimate (4.5), we further get

$$\mathbf{v}^T\mathbf{S}^{\mathrm{c}}\mathbf{v}\ge\alpha C^{-2}C_{\mathrm{P}}^{-2}\|v\|_{L^2(D)}^2=\alpha C^{-2}C_{\mathrm{P}}^{-2}\mathbf{v}^T\mathbf{M}\mathbf{v}\ge cH^d\|\mathbf{v}\|_2^2.\tag{B.2}$$

The claim follows since the minimal eigenvalue is bounded from above by the Rayleigh quotient.

B.3. Proof of Theorem 4.3. We only show the first estimate. The second one follows directly from the equivalence

$$H^{d/2}\|\mathbf{u}^{\mathrm{pg}}-\mathbf{u}^{\mathrm{nn}}\|_2\lesssim\|u_H^{\mathrm{pg}}-u_H^{\mathrm{nn}}\|_{L^2(D)}\lesssim H^{d/2}\|\mathbf{u}^{\mathrm{pg}}-\mathbf{u}^{\mathrm{nn}}\|_2,\tag{B.3}$$

which follows from (4.5). With Lemma 4.1, we have

$$\|\mathbf{u}^{\mathrm{pg}}-\mathbf{u}^{\mathrm{nn}}\|_2\le\|\mathbf{u}^{\mathrm{pg}}-\mathbf{u}^{\mathrm{c}}\|_2+\|\mathbf{u}^{\mathrm{c}}-\mathbf{u}^{\mathrm{nn}}\|_2\lesssim H^{-d/2}e^{-c_{\mathrm{loc}}\ell}+\|\mathbf{u}^{\mathrm{c}}-\mathbf{u}^{\mathrm{nn}}\|_2.\tag{B.4}$$

We now bound the second term. Note that the right-hand side involving f is deterministic and thus equal for the discrete problems (2.8), (2.7), and the system based

on the neural network. Denoting the vector-version of the right-hand side with $\mathbf{f}$, we particularly have the linear systems

$$\mathbf{S}^{\mathrm{pg}}\mathbf{u}^{\mathrm{pg}} = \mathbf{f}, \qquad \mathbf{S}^{\mathrm{c}}\mathbf{u}^{\mathrm{c}} = \mathbf{f}, \qquad \mathbf{S}^{\mathrm{nn}}\mathbf{u}^{\mathrm{nn}} = \mathbf{f}, \tag{B.5}$$

where $\mathbf{S}^{\mathrm{pg}}$ is the PG-LOD stiffness matrix, $\mathbf{S}^{\mathrm{c}}$ the (symmetric) C-LOD stiffness matrix, and $\mathbf{S}^{\mathrm{nn}}$ the one assembled from the neural network. Using (B.5), we estimate

$$\begin{aligned}\|\mathbf{u}^{\mathrm{c}} - \mathbf{u}^{\mathrm{nn}}\|_2 &= \big\|\big(\mathbf{S}^{\mathrm{c}}\big)^{-1}\mathbf{f} - \mathbf{u}^{\mathrm{nn}}\big\|_2 = \big\|\big(\mathbf{S}^{\mathrm{c}}\big)^{-1}(\mathbf{S}^{\mathrm{nn}}\mathbf{u}^{\mathrm{nn}} - \mathbf{S}^{\mathrm{c}}\mathbf{u}^{\mathrm{nn}})\big\|_2 \\ &\le \big\|\big(\mathbf{S}^{\mathrm{c}}\big)^{-1}\big\|_2 \big\|\mathbf{S}^{\mathrm{nn}} - \mathbf{S}^{\mathrm{c}}\big\|_2 \big\|\mathbf{u}^{\mathrm{nn}}\big\|_2.\end{aligned} \tag{B.6}$$

With the eigenvalue bound on $\mathbf{S}^{\mathrm{c}}$ that is proven in Lemma 4.2, we obtain

$$\big\|\big(\mathbf{S}^{\mathrm{c}}\big)^{-1}\big\|_2 = \big(\lambda_{\min}(\mathbf{S}^{\mathrm{c}})\big)^{-1} \lesssim H^{-d}. \tag{B.7}$$

For the second factor on the right-hand side of (B.6), we estimate

$$\|\mathbf{S}^{\mathrm{nn}} - \mathbf{S}^{\mathrm{c}}\|_2 \le \|\mathbf{S}^{\mathrm{nn}} - \mathbf{S}^{\mathrm{pg}}\|_2 + \|\mathbf{S}^{\mathrm{pg}} - \mathbf{S}^{\mathrm{c}}\|_2.$$

With the choice $k = 2d+1$ in Corollary 3.13 and (4.2)–(4.4), we obtain the bounds in (4.6) as well as

$$\begin{aligned}\|\mathbf{S}^{\mathrm{nn}} - \mathbf{S}^{\mathrm{pg}}\|_2 &\le \sum_{K\in\mathcal{T}_H} \big\|\Phi_K\big(\boldsymbol{\Theta}_K - \mathbf{S}^{\mathrm{pg}}_{\mathtt{N}^\ell(K)}\big)\big\|_2 \\ &\le \sum_{K\in\mathcal{T}_H} \big\|\boldsymbol{\Theta}_K - \mathbf{S}^{\mathrm{pg}}_{\mathtt{N}^\ell(K)}\big\|_2 \lesssim H^{-d}H^k \lesssim H^{d+1}.\end{aligned} \tag{B.8}$$

The difference of the C-LOD and the PG-LOD stiffness matrices can be bounded as follows. Let $\mathbf{v}$ be a vector (with corresponding function $v \in V_H$) such that

$$\|\mathbf{S}^{\mathrm{pg}} - \mathbf{S}^{\mathrm{c}}\|_2 = |\lambda_{\max}(\mathbf{S}^{\mathrm{pg}} - \mathbf{S}^{\mathrm{c}})| = \frac{\big|\mathbf{v}^T(\mathbf{S}^{\mathrm{pg}} - \mathbf{S}^{\mathrm{c}})\mathbf{v}\big|}{\mathbf{v}^T\mathbf{v}}.$$

With the definition of $\mathbf{S}^{\mathrm{pg}}$ and $\mathbf{S}^{\mathrm{c}}$ corresponding to the discrete problems (2.8) and (2.7), respectively, we thus have

$$\begin{aligned}\|\mathbf{S}^{\mathrm{pg}} - \mathbf{S}^{\mathrm{c}}\|_2 &= \frac{|a((\mathsf{id} - \mathcal{Q}^\ell)v, v) - a((\mathsf{id} - \mathcal{Q}^\ell)v, (\mathsf{id} - \mathcal{Q}^\ell)v)|}{\mathbf{v}^T\mathbf{v}} \\ &= \frac{|a((\mathsf{id} - \mathcal{Q}^\ell)v, \mathcal{Q}^\ell v)|}{\mathbf{v}^T\mathbf{v}} \lesssim \frac{|a((\mathcal{Q} - \mathcal{Q}^\ell)v, \mathcal{Q}^\ell v)|}{H^{-d}\|v\|^2_{L^2(D)}}\end{aligned}$$

using the definition of the globally defined correction $\mathcal{Q} := \mathcal{Q}^\infty$ and (4.5) in the last step. Employing the estimate (2.11), the stability of the correction operator $\mathcal{Q}^\ell$, and a classical inverse inequality, we arrive at

$$\begin{aligned}\|\mathbf{S}^{\mathrm{pg}} - \mathbf{S}^{\mathrm{c}}\|_2 &\lesssim \frac{|a((\mathcal{Q} - \mathcal{Q}^\ell)v, \mathcal{Q}^\ell v)|}{H^{-d}\|v\|^2_{L^2(D)}} \lesssim \frac{\beta\,\|\nabla(\mathcal{Q} - \mathcal{Q}^\ell)v\|_{L^2(D)}\,\|\nabla\mathcal{Q}^\ell v\|_{L^2(D)}}{H^{-d}\|v\|^2_{L^2(D)}} \\ &\lesssim \frac{e^{-c_{\mathrm{loc}}\ell}\,\|\nabla v\|^2_{L^2(D)}}{H^{-d+2}\|\nabla v\|^2_{L^2(D)}} \lesssim H^{d-2}e^{-c_{\mathrm{loc}}\ell}.\end{aligned} \tag{B.9}$$

For the last term on the right-hand side of (B.6), we estimate

$$\begin{aligned}\|\mathbf{u}^{\mathrm{nn}}\|_2 &\le \|\mathbf{u}^{\mathrm{pg}} - \mathbf{u}^{\mathrm{nn}}\|_2 + \|\mathbf{u}^{\mathrm{pg}}\|_2 \le \|\mathbf{u}^{\mathrm{pg}} - \mathbf{u}^{\mathrm{nn}}\|_2 + CH^{-d/2}\|u_H^{\mathrm{pg}}\|_{L^2(D)} \\ &\le \|\mathbf{u}^{\mathrm{pg}} - \mathbf{u}^{\mathrm{nn}}\|_2 + CH^{-d/2}\|f\|_{L^2(D)},\end{aligned}$$

where we employ the stability of the solution u_H^{pg} and the Friedrichs inequality. We now go back to (B.4) and (B.6) and obtain

$$\begin{aligned}\|\mathbf{u}^{\mathrm{pg}}-\mathbf{u}^{\mathrm{nn}}\|_2 &\le CH^{-d/2}e^{-c_{\mathrm{loc}}\ell}+\left\|\left(\mathbf{S}^{\mathrm{c}}\right)^{-1}\right\|_2\left\|\mathbf{S}^{\mathrm{nn}}-\mathbf{S}^{\mathrm{c}}\right\|_2\left\|\mathbf{u}^{\mathrm{nn}}\right\|_2\\ &\le CH^{-d/2}e^{-c_{\mathrm{loc}}\ell}+\left\|\left(\mathbf{S}^{\mathrm{c}}\right)^{-1}\right\|_2\left\|\mathbf{S}^{\mathrm{nn}}-\mathbf{S}^{\mathrm{c}}\right\|_2 CH^{-d/2}\|f\|_{L^2(D)}\\ &\qquad+\left\|\left(\mathbf{S}^{\mathrm{c}}\right)^{-1}\right\|_2\left\|\mathbf{S}^{\mathrm{nn}}-\mathbf{S}^{\mathrm{c}}\right\|_2\left\|\mathbf{u}^{\mathrm{pg}}-\mathbf{u}^{\mathrm{nn}}\right\|_2\\ &\le CH^{-d/2}e^{-c_{\mathrm{loc}}\ell}+CH^{1-d/2}\|f\|_{L^2(D)}+\frac{1}{2}\|\mathbf{u}^{\mathrm{pg}}-\mathbf{u}^{\mathrm{nn}}\|_2.\end{aligned}\tag{B.10}$$

In the last step, we have used that

$$\left\|\left(\mathbf{S}^{\mathrm{c}}\right)^{-1}\right\|_2\left\|\mathbf{S}^{\mathrm{nn}}-\mathbf{S}^{\mathrm{c}}\right\|_2\lesssim H^{-d}\left(H^{d+1}+H^{d-2}e^{-c_{\mathrm{loc}}\ell}\right)\lesssim H\le\frac{1}{2}$$

according to (B.7), (B.8), and (B.9) if H is chosen small enough and $\ell\gtrsim|\log(H)|$ large enough with respect to the respective hidden constant. Absorbing the last term on the right-hand side of (B.10) leads to

$$\|\mathbf{u}^{\mathrm{pg}}-\mathbf{u}^{\mathrm{nn}}\|_2\lesssim H^{-d/2}e^{-c_{\mathrm{loc}}\ell}+H^{1-d/2}\,\|f\|_{L^2(D)}.$$

Omitting the dependence on f and using again that $\ell\gtrsim|\log(H)|$, we finally get

$$\|\mathbf{u}^{\mathrm{pg}}-\mathbf{u}^{\mathrm{nn}}\|_2\lesssim H^{1-d/2}.$$

Employing (B.3), this directly leads to

$$\|u_H^{\mathrm{pg}}-u_H^{\mathrm{nn}}\|_{L^2(D)}\lesssim H.$$

∎

REFERENCES

[ABS+20] H. Arbabi, J. E. Bunder, G. Samaey, A. J. Roberts, and I. G. Kevrekidis, *Linking machine learning with multiscale numerics: Data-driven discovery of homogenized equations*, JOM, 72 (2020), pp. 4444–4457.

[AH17] A. Abdulle and P. Henning, *Localized orthogonal decomposition method for the wave equation with a continuum of scales*, Math. Comp., 86 (2017), pp. 549–587.

[AHP21] R. Altmann, P. Henning, and D. Peterseim, *Numerical homogenization beyond scale separation*, Acta Numer., 30 (2021), pp. 1–86.

[BL11] I. Babuška and R. Lipton, *Optimal local approximation spaces for generalized finite element methods with application to multiscale problems*, Multiscale Model. Simul., 9 (2011), pp. 373–406, https://doi.org/10.1137/100791051.

[CAH21] M. J. Colbrook, V. Antun, and A. C. Hansen, *Can Stable and Accurate Neural Networks be Computed?–On the Barriers of Deep Learning and Smale's 18th Problem*, preprint, https://arxiv.org/abs/2101.08286, 2021.

[CE18] S. Chan and A. H. Elsheikh, *A machine learning approach for efficient uncertainty quantification using multiscale methods*, J. Comput. Phys., 354 (2018), pp. 493–511.

[CLPZ21] E. Chung, W. T. Leung, S.-M. Pun, and Z. Zhang, *A multi-stage deep learning based algorithm for multiscale model reduction*, J. Comput. Appl. Math., 394 (2021), 113506.

[CMP20] A. Caiazzo, R. Maier, and D. Peterseim, *Reconstruction of quasi-local numerical effective models from low-resolution measurements*, J. Sci. Comput., 85 (2020), 10.

[EG17] A. Ern and J.-L. Guermond, *Finite element quasi-interpolation and best approximation*, ESAIM Math. Model. Numer. Anal., 51 (2017), pp. 1367–1385.

[EGJS21] D. Elbrächter, P. Grohs, A. Jentzen, and C. Schwab, *DNN expression rate analysis of high-dimensional PDEs: Application to option pricing*, Constr. Approx., 55 (2022), pp. 3–71.

[EGW11] Y. R. Efendiev, J. Galvis, and X.-H. Wu, *Multiscale finite element methods for high-contrast problems using local spectral basis functions*, J. Comput. Phys., 230 (2011), pp. 937–955.

[Fri73] I. Fried, *Bounds on the spectral and maximum norms of the finite element stiffness, flexibility and mass matrices*, Internat. J. Solids Structures, 9 (1973), pp. 1013–1034.

[GHV18] D. Gallistl, P. Henning, and B. Verfürth, *Numerical homogenization of H(curl)-problems*, SIAM J. Numer. Anal., 56 (2018), pp. 1570–1596, https://doi.org/10.1137/17M1133932.

[GM21] S. Geevers and R. Maier, *Fast mass lumped multiscale wave propagation modelling*, IMA J. Numer. Anal., 43 (2023), pp. 44–72.

[GP15] D. Gallistl and D. Peterseim, *Stable multiscale Petrov–Galerkin finite element method for high frequency acoustic scattering*, Comput. Methods Appl. Mech. Engrg., 295 (2015), pp. 1–17.

[GP17] D. Gallistl and D. Peterseim, *Computation of quasi-local effective diffusion tensors and connections to the mathematical theory of homogenization*, Multiscale Model. Simul., 15 (2017), pp. 1530–1552, https://doi.org/10.1137/16M1088533.

[GRK20] I. Gühring, M. Raslan, and G. Kutyniok, *Expressivity of Deep Neural Networks*, preprint, https://arxiv.org/abs/2007.04759, 2020.

[HL22] J. Han and Y. Lee, *A Neural Network Approach for Homogenization of Multiscale Problems*, preprint, https://arxiv.org/abs/2206.02032, 2022.

[HP13] P. Henning and D. Peterseim, *Oversampling for the multiscale finite element method*, Multiscale Model. Simul., 11 (2013), pp. 1149–1175, https://doi.org/10.1137/120900332.

[HP21] M. Hauck and D. Peterseim, *Super-localization of Elliptic Multiscale Problems*, preprint, arXiv:2107.13211, 2021.

[HP22] M. Hauck and D. Peterseim, *Multi-resolution localized orthogonal decomposition for Helmholtz problems*, Multiscale Model. Simul., 20 (2022), pp. 657–684, https://doi.org/10.1137/21M1414607.

[HW22] P. Henning and J. Wärnegård, *Superconvergence of time invariants for the Gross-Pitaevskii equation*, Math. Comp., 91 (2022), pp. 509–555.

[KMP22] F. Kröpfl, R. Maier, and D. Peterseim, *Operator compression with deep neural networks*, Adv. Contin. Discrete Models, 2022 (2022), 29.

[KPRS21] G. Kutyniok, P. Petersen, M. Raslan, and R. Schneider, *A theoretical analysis of deep neural networks and parametric PDEs*, Constr. Approx., 55 (2021), pp. 73–125.

[LLZ22] W. T. Leung, G. Lin, and Z. Zhang, *NH-PINN: Neural homogenization-based physics-informed neural network for multiscale problems*, J. Comput. Phys., 470 (2022), 111539.

[LMM22] P. Ljung, R. Maier, and A. Målqvist, *A space-time multiscale method for parabolic problems*, Multiscale Model. Simul., 20 (2022), pp. 714–740, https://doi.org/10.1137/21M1446605.

[LMP21] P. Ljung, A. Målqvist, and A. Persson, *A generalized finite element method for the strongly damped wave equation with rapidly varying data*, ESAIM Math. Model. Numer. Anal., 55 (2021), pp. 1375–1403.

[Mai20] R. Maier, *Computational Multiscale Methods in Unstructured Heterogeneous Media*, Ph.D. thesis, University of Augsburg, Augsburg, Germany, 2020.

[Mai21] R. Maier, *A high-order approach to elliptic multiscale problems with general unstructured coefficients*, SIAM J. Numer. Anal., 59 (2021), pp. 1067–1089, https://doi.org/10.1137/20M1364321.

[MP14] A. Målqvist and D. Peterseim, *Localization of elliptic multiscale problems*, Math. Comp., 83 (2014), pp. 2583–2603.

[MP18] A. Målqvist and A. Persson, *Multiscale techniques for parabolic equations*, Numer. Math., 138 (2018), pp. 191–217.

[MP19] R. Maier and D. Peterseim, *Explicit computational wave propagation in micro-heterogeneous media*, BIT Numer. Math., 59 (2019), pp. 443–462.

[MP20] A. Målqvist and D. Peterseim, *Numerical Homogenization by Localized Orthogonal Decomposition*, SIAM Spotlights 5, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2020, https://doi.org/10.1137/1.9781611976458.

[MV22] R. Maier and B. Verfürth, *Multiscale scattering in nonlinear Kerr-type media*, Math. Comp., 91 (2022), pp. 1655–1685.

[OPS20] J. A. A. Opschoor, P. C. Petersen, and C. Schwab, *Deep ReLU networks and high-order finite element methods*, Anal. Appl., 18 (2020), pp. 715–770.

[Owh17] H. Owhadi, *Multigrid with rough coefficients and multiresolution operator decomposition from hierarchical information games*, SIAM Rev., 59 (2017), pp. 99–149, https://doi.org/10.1137/15M1013894.

[OZB14] H. Owhadi, L. Zhang, and L. Berlyand, *Polyharmonic homogenization, rough polyharmonic splines and sparse super-localization*, ESAIM Math. Model. Numer. Anal., 48 (2014), pp. 517–552.

[Pet01] M. Petzoldt, *Regularity and Error Estimators for Elliptic Problems with Discontinuous Coefficients*, Ph.D. thesis, Freie Universität Berlin, Berlin, Germany, 2001.

[Pet16] D. Peterseim, *Variational multiscale stabilization and the exponential decay of fine-scale correctors*, in Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations, Lect. Notes Comput. Sci. Eng. 114, Springer, Cham, 2016, pp. 341–367.

[Pet17] D. Peterseim, *Eliminating the pollution effect in Helmholtz problems by local subscale correction*, Math. Comp., 86 (2017), pp. 1005–1036.

[PS12] D. Peterseim and S. A. Sauter, *Finite elements for elliptic problems with highly varying, nonperiodic diffusion matrix*, Multiscale Model. Simul., 10 (2012), pp. 665–695, https://doi.org/10.1137/10081839X.

[PS17] D. Peterseim and M. Schedensack, *Relaxing the CFL condition for the wave equation on adaptive meshes*, J. Sci. Comput., 72 (2017), pp. 1196–1213.

[PV18] P. Petersen and F. Voigtlaender, *Optimal approximation of piecewise smooth functions using deep ReLU neural networks*, Neural Netw., 108 (2018), pp. 296–330.

[PZ21] G. A. Padmanabha and N. Zabaras, *A Bayesian Multiscale Deep Learning Framework for Flows in Random Media*, preprint, https://arxiv.org/abs/2103.09056, 2021.

[RHB19] X. Ren, A. Hannukainen, and A. Belahcen, *Homogenization of multiscale eddy current problem by localized orthogonal decomposition method*, IEEE Trans. Magn., 55 (2019), pp. 1–4.

[SSM23] S. Stepanov, D. Spiridonov, and T. Mai, *Prediction of numerical homogenization using deep learning for the Richards equation*, J. Comput. Appl. Math., 424 (2023), 114980.

[WHG+21] A. Waseem, T. Heuzé, M. G. D. Geers, V. G. Kouznetsova, and L. Stainier, *Data-driven reduced homogenization for transient diffusion problems with emergent history effects*, Comput. Methods Appl. Mech. Engrg., 380 (2021), 113773.