A Block-Coordinate Approach of Multi-level Optimization with an Application to Physics-Informed Neural Networks

Serge Gratton*, Valentin Mercier†, Elisa Riccietti‡, Philippe L. Toint§

11 V 2023

Abstract

Multi-level methods are widely used for the solution of large-scale problems, because of their computational advantages and exploitation of the complementarity between the involved sub-problems. After a re-interpretation of multi-level methods from a block-coordinate point of view, we propose a multi-level algorithm for the solution of nonlinear optimization problems and analyze its evaluation complexity. We apply it to the solution of partial differential equations using physics-informed neural networks (PINNs) and consider two different types of neural architectures, a generic feedforward network and a frequency-aware network. We show that our approach is particularly effective if coupled with these specialized architectures and that this coupling results in better solutions and significant computational savings.

Keywords: nonlinear optimization, multi-level methods, partial differential equations (PDEs), physics-informed neural networks (PINNs), deep learning.

1 Introduction

Many numerical optimization problems of interest today are large dimensional and techniques to solve them efficiently are thus an active field of research. A very powerful class of algorithms for the solution of large problems is that of multi-level methods. Originally the concept of a method exploiting multiple levels (i.e. multiple resolutions of an underlying problem) was introduced for the solution of large scale systems arising from the discretization of partial differential equations (PDEs). In this context these methods are known as multigrid (MG) methods for the linear case or full approximation schemes (FAS) for the nonlinear one [54]. These schemes were later extended to nonlinear optimization problems, in which context they are known as multi-level optimization techniques [2, 13, 14, 15, 7]. The central idea of all these approaches is to use the structure of the problem in order to significantly reduce the computational cost compared to standard approaches applied to the full unstructured problem.

In this paper we introduce a new interpretation of multi-level methods as block coordinate descent (BCD) methods: iterations at coarse levels (i.e. low resolution) can be interpreted as the (possibly approximate) solution of a subproblem involving a set of variables smaller than that required to describe the fine level (high resolution). We propose a framework that allows us to encompass multi-level methods for several classes of problems as well as a unifying complexity analysis based on a generic block coordinate descent, which is simple yet comprehensive.

To illustrate the effectiveness of the proposed approach, we apply our framework in the context of deep learning. The idea of exploiting multiple scales in learning has been explored for different kind of networks. For instance, [23, 1, 46, 42] propose multilevel methods for the training of

*Université de Toulouse, INP-ENSEEIHT, IRIT, Toulouse, France.

†Université de Toulouse, ANITI, CERFACS, IRIT, Toulouse, and BRLi, France.

‡ENS Lyon: Ecole normale superieure de Lyon (Université de Lyon, INRIA, ENSL, UCBL, CNRS, LIP UMR 5668, Lyon, France). Corresponding author. Email: elisa.riccietti@ens-lyon.fr.

§Namur Center for Complex Systems (naXys), University of Namur, Namur, Belgium.

deep residual networks ResNets in which the multilevel hierarchy and the transfer operators are constructed by exploiting a dynamical system s interpretation. Multi-scale methods for convolutional neural networks and recurrent networks have also been proposed in 17 41 and respectively. ur focus in this paper is on physics-informed neural networks PINNs . These networks have been introduced in 27 and have exhibited good performance in practice soon supported by theoretical results 2 37 . ee 6 for a comprehensive review on the topic.

Despite their success the training of such networks may remain di cult in particular for highly nonlinear or multi-frequency problems 43 . In particular choosing an e cient detailed network architecture and an associated training procedure is far from obvious especially if the problem s solution involves high frequency components 47 has described under the name of *F-principle* why it might be so see also 33 36 44 11 . hile speci c "frequency aware" network architectures such as P 43 structures and Mscale networks 26 24 have been proposed to circumvent this latter di culty the mere size of the networks necessary to represent solutions of PD s with su cient accuracy still make their computationally e cient training very challenging. ur ob ective is to make this challenge more tractable.

Contributions. In this context the speci c contributions of this paper may be summarized as follows.

1. e present a unifying framework for a large class of multi-level problems ection 2 .
2. e then introduce a suitable block-coordinate descent algorithm and analyze its evaluation complexity bound under standard assumptions ection 3 .
3. e next investigate how this approach can be applied to PINNs for the solution of Laplace and heat propagation problems on complex geometries.
   a In a rst step we show that a multi-level technique based on alternate training of "coarse" and " ne" networks may bring substantial computational bene ts ection 4 .
   b e then exploit frequency aware network architectures in this multi-level context allowing the e cient solution of more complex problems ection 5 .

   Compared with standard single-level training both approaches are shown to yield better solutions at a much reduced computational cost.

2 A block-coordinate perspective on Galerkin multilevel optimization

ur purpose is to present a new at least as far as we know but yet simple perspective on multilevel optimization using Galerkin approximations. Given some space $\mathcal{F}$ of continuous functions from $\mathbb{R}^p$ to $\mathbb{R}^q$ and some ob ective function from $\mathcal{F}$ into $\mathbb{R}$ our global aim is to compute a function y which is a possibly approximate solution of the variational problem

$$\min_{y \in \mathcal{F}} f\ y\ . \qquad 1$$

The ob ective function f is often given by some norm of the residual of a problem of interest PD D boundary value problem linear system or other but other cases are possible such as minimum surface or contact problems for instance . e assume that $\mathcal{F}$ consists of functions constructed by linearly or nonlinearly combining elemental basis functions using a parametrization involving the parameters $x \in \mathbb{R}^n$ so that y is denoted by $y\ x$ whereas the value of $y\ x$ at z will be denoted by $y\ x\ z$. The problem then reduces to nding the value s of x such that $f\ y\ x$ is minimized. In this paper we focus on a class of "splitting" techniques whose ob ective is to reduce the computational cost associated with this minimization. More speci cally we consider

the case where y is viewed as the sum of two terms y_1 and y_2 themselves depending on their own sets of parameters $x_1 \in \mathbb{R}^{n_1}$ and $x_2 \in \mathbb{R}^{n_2}$ that is

$$y\ x \quad y_1\ x_1 \quad y_2\ x_2\ . \tag{2}$$

This yields the optimization problem

$$\min_{(x_1,x_2)\in\mathbb{R}^n} f\ y_1\ x_1 \quad y_2\ x_2\ ,$$

where $n \quad n_1 \quad n_2$. e also associate with the splitting 2 the following "approximation sets"

$$\mathcal{A}_{12} \quad \{y \in \mathcal{F} \mid y\ x \quad y_1\ x_1 \quad y_2\ x_2 \quad \text{for some} \quad x_1, x_2\ \in \mathbb{R}^n\} \tag{3}$$

as well as

$$\mathcal{A}_i \quad \{y \in \mathcal{F} \mid y\ x \quad y_i\ x_i \quad \text{for some}\ x_i \in \mathbb{R}^{n_i}\} \quad i \quad 1, 2\ . \tag{4}$$

hile our present development is based on an additive structure of the function y other formulations could obviously be of interest. e limited the number of terms to two in order to simplify exposition but this not restrictive as we discuss below .

e now discuss some examples of this approach in which we distinguish two main contexts.

The hierarchical context: The terms y_1 and y_2 are constructed such that

$$\mathcal{A}_2 \subset \mathcal{A}_1 \quad \mathcal{A}_{12}. \tag{5}$$

This may occur in variety of cases the simplest being the classical multigrid framework in which one assumes that f is a strictly convex quadratic and the y_i are linear. The quadratic s minimization is then equivalent to the solution of a positive de nite linear system. The y_i are constructed as linear combination of basis functions $\{b_j\}_{i=1}^m$ of $\mathcal{F}$ typically from a nite-di erences or nite-elements basis that is

$$y_1\ x_1 \quad \sum_{j=1}^{m} x_1\ _j\, b_j \quad \text{and} \quad y_2\ x_2 \quad \sum_{j=1}^{m} Px_2\ _j\, b_j, \tag{6}$$

where $m \quad n_1$ is the dimension of $\mathcal{F}$ and where P is a $n_1 \times n_2$ linear "prolongation" operator from a "coarse" space of dimension $n_2 \leq n_1$ to the " ne" space of dimension n_1. In the multigrid framework these coarse and ne spaces often correspond to coarse and ne discretizations of an underlying continuous problem but other interpretations such as domain decomposition are possible. The quadratic optimization problem then becomes

$$\min_{(x_1,x_2)\in\mathbb{R}^{n_1+n_2}} \tfrac{1}{2}\ x_1 \quad Px_2\ ^T A\ x_1 \quad Px_2 \quad b^T\ x_1 \quad Px_2 \tag{7}$$

for some positive de nite matrix A and right-hand side b depending of the basis $\mathcal{B} \quad \{b_j\}_{i=1}^m$. ne easily veri es that "restricting" the minimization to the x_2 variables amounts to solving the n_2-dimensional problem

$$\min_{x_2\in\mathbb{R}^{n_2}} \tfrac{1}{2} x_2^T PAP^T x_2 \quad P^T b \quad P^T Ax_1\ ^T x_2,$$

which is the usual Galerkin approximation of f at the coarse level. e clearly have that

$$\mathcal{A}_2 \quad \left\{\sum_{j=1}^{m} Px_2\ _j\, b_j\right\} \subset \left\{\sum_{j=1}^{m} x_{1,j} \quad Px_2\ _j\ b_j\right\} \quad \text{span}\ \mathcal{B} \quad \mathcal{A}_1 \quad \mathcal{A}_{12},$$

ensuring 5 . Classical multigrid methods then alternate approximate resolutions of the n_2-dimensional "coarse level" problem and the n-dimensional " ne level" one given by 7 .

A second more nonlinear case is when f may no longer be a convex quadratic but a smooth nonlinear function which we assume is bounded below for consistency while keeping 6 and its interpretation in terms of "coarse" and " ne" spaces. Reusing 6 and we may now consider

$$\widehat{f}\ x_1, x_2 \ \overset{\text{def}}{=}\ f\left(\sum_{j=1}^{m} x_{1,j} b_j \quad \sum_{j=1}^{m} Px_2\ _j\, b_j\right) \quad f\left(\sum_{j=1}^{m} x_{1,j} \quad Px_2\ _j\ b_j\right) \overset{\text{def}}{=} F\ x_1 \quad Px_2\ , \tag{1}$$

which is a reformulation of the original ob ective function incorporating the dependence of the basis $\mathcal{B}$ and as above alternatively perform iterations on the problems

$$\min_{(x_1,x_2)\in\mathbb{R}^{n_1+n_2}} \widehat{f}\ x_1, x_2 \quad \text{and} \quad \min_{\substack{x_2\ \in\ \mathbb{R}^{n_2}\\ x_1 \quad \text{xed}}} \widehat{f}\ x_1, x_2$$

and note that because of the rst of these problems is equivalent in the sense that they yield the same value for $y_1\ x_1 \quad y_2\ x_2$ to the lower-dimensional

$$\min_{\substack{x_1\ \in\ \mathbb{R}^{n_1}\\ x_2 \quad \text{xed}}} \widehat{f}\ x_1, x_2\ .$$

This second approach has been explored in the framework of nonlinear optimization by the MG- PT 2 and RMTR 15 methods. Both these algorithms consider a subproblem where a coarse-level model h of the ob ective function is approximately minimized to nd an increment δ to update the ne iterate. ur hierarchical approach corresponds to the choice of the Galerkin Taylor models de ned by

$$h_1\ \delta x_2 \quad \sigma\ P^T\nabla_x F\ x_1 \quad Px_2\ \ ^T\delta x_2$$

for rst order and

$$h_2\ \delta x_2 \quad \sigma\ P^T\nabla_x F\ x_1 \quad Px_2\ \ ^T\delta x_2 \quad \tfrac{1}{2}\sigma^2\delta x_2^T P\nabla_x^2 F\ x_1 \quad Px_2\ P^T\delta x_2,$$

for second order where σ is a xed positive constant and δx_2 is an increment in x_2 from the point x_1, x_2 . Note that the form of h_2 is identical to that of .

The framework ust described is also closely related to the FA approach 5 p. for solving a set of nonlinear equations. If we consider the equation $\nabla F\ x$ the standard FA approach would consist in solving at a given ne iterate x_1 the problem $P^T\nabla F\ P^T x_1 \quad x_2\ P$ rhs where the right-hand side rhs is such that if x_1 solves the problem i.e. annihilates the gradient of $F \quad x_2$ is zero. This approach is di erent from ours in that the correction Px_2 in the coarse problem is added to $PP^T x_1$ in the coarse problem de nition and not to x_1 itself. Taking this into account we may consider the coarse equation in $x_2\ P^T\nabla F\ x_1 \quad Px_2$ for which we obtain a formulation that is now in line with our hierarchical context since the derivative of this coarse equation are identical to those involved in the above de nition of h_2. Note that the right-hand side of this equation is now zero since x_2 solves the coarse problem when $\nabla F\ x_1$.

In what follows we will be especially interested in the case where

$$y_i\ x_i \quad \mathtt{NN}_i\ x_i \tag{11}$$

where $\mathtt{NN}_i\ x_i$ is a neural network of input z parameters weights and biases given by $x_i \in \mathbb{R}^{n_i}$ and output $\mathtt{NN}_i\ x_i \quad z$. Neural networks are clearly nonlinear and nonconvex functions of the parameters x_i and the hierarchical context occurs when y_2 is a subnetwork of y_1.

The distributed context: e may now abandon 5 and consider a situation where neither $\mathcal{A}_1$ or $\mathcal{A}_2$ is identical to $\mathcal{A}_{12}$. This is for instance the case when 11 holds but y_2 is not a subnetwork of y_1 and n_1 and n_2 the number of network parameters in y_1 and y_2 respectively are now independent of m. ur proposal is to use a similar methodology for this more complex case and alternate approximate minimizations on the "$\mathcal{A}_2$ subproblem" given by

$$\begin{array}{ll} \min\limits_{x_2 \in \mathbb{R}^{n_2}} & f\ y_1\ x_1 \quad y_2\ x_2 \\ x_1 \quad \text{xed} \end{array} \tag{12}$$

with that on the "$\mathcal{A}_1$ subproblem" given by

$$\begin{array}{ll} \min\limits_{x_1 \in \mathbb{R}^{n_1}} & f\ y_1\ x_1 \quad y_2\ x_2 \quad . \\ x_2 \quad \text{xed} \end{array} \tag{13}$$

In all cases described above the computational cost of the overall minimization is expected to decrease because we may choose n_2 and for the distrubuted case n_1 to be signi cantly smaller than n. Alternating standard minimization steps on the ne $\mathcal{A}_1$ level with cheap ones at the coarse $\mathcal{A}_2$ level is therefore computationally attractive *provided the coarse steps significantly contribute to the overall minimization.*

e conclude this section by noting that it is obviously possible to consider more general additive splittings of the form

$$y\ x, z \quad \sum_{i=1}^{s} y_i\ x_i, z$$

in our above developments. In the the hierarchical context that is in the classical multigrid approach and in the RMTR MG- PT algorithms this is achieved by considering a hierarchy of nested approximations sets

$$\mathcal{A}_{\{\ell,\ldots,s\}} \quad \left\{ y \mid y \quad \sum_{j=1}^{m} x_{\ell,j} \quad P_1\ x_{\ell+1,j} \quad P_2\ \ldots P_k\ x_{s-1,j} \quad P_s x_{s,j} \ \ldots \quad b_j \right\}$$

for $\ell \in \{1, \ldots, s\}$ so that $\mathcal{A}_{\{\ell+1,\ldots,s\}} \subset \mathcal{A}_{\{\ell,\ldots,s\}}$ for each $\ell \in \{1, \ldots, s-1\}$.

In the distributed context one could consider the sets

$$\mathcal{A}_{\mathcal{S}} \quad \left\{ y \mid y \quad \sum_{i \in \mathcal{S}} y_i\ x_i \right\} \tag{14}$$

for all nonempty subsets $\mathcal{S}$ of $\{1, \ldots, s\}$. This allows for a wide variety of set architectures such as the "recursive" one using $\mathcal{A}_{\{\ell,\ldots,s\}}$ for $\ell \in \{1, \ldots, s\}$ the " at" one using $\mathcal{A}_{\{\ell\}}$ for $\ell \in \{1, \ldots, s\}$ or any mixture of these. In full generality the approximation sets 14 need not be dis oint and it is then useful for a computationally e ective architecture to identify which subsets $\mathcal{S}$ generate identical $\mathcal{A}_{\mathcal{S}}$ and to ignore those for which $\sum_{i \in \mathcal{S}} n_i$ the dimension of the associated minimization problem is not minimal.

As a consequence of the above discussion we see that a wide variety of multilevel optimization methods may be viewed as block-coordinate minimization problems where the blocks of variables are given by the x_i.

3 A generic BCD algorithm and its convergence

e now examine why splitting minimization into alternate block minimization can be useful and consider the associated convergence guarantees. e rst note that such guarantees are already available for linear multigrid case 5 1 4 for FA -like methods 3 as well as for MG- PT

2 and RMTR 15 14 16 7 . The ob ective of this section is to motivate and state a simple yet comprehensive complexity analysis covering the two contexts described above.

The success of existing multilevel methods is based on exploiting a "complementarity" between the various minimization problems involved which we pursue as follows. Given the overall problem 1 one starts by considering a particular minimization method and isolate a class of problems for which this method is e cient. In the hierarchical context this is typically the Gauss- eidel or acobi method and the class of problems where such "smoothing methods" are e cient is that of problems involving high-frequency behaviour in the sought y function of the underlying variable z see 5 Chapter 2 . This suggests that one might wish to split the problem if at all possible depending of its frequency content. To achieve this one chooses at least implicitly the basis $\mathcal{B}$ which spans $\mathcal{A}_{12}$ and $\mathcal{A}_1$ to be a Fourier basis of $\mathcal{F}$ and split the problem into nding the coe - cients of the high-frequency basis functions using a smoothing method in $\mathcal{A}_{12}$ $\mathcal{A}_1$ and nding those of the low frequency ones $\mathcal{A}_2$. The key of the approach is to transform the low-frequency subproblem into a high frequency one by shifting frequencies making the smoothing algorithm e cient also for this subproblem. This shift is usually achieved by considering a coarser discretization of the underlying continuous problem the "coarse space" . Classical multigrid methods and also nonlinear methods in the hierarchical context then alternate minimizations steps for the high-frequency " ne" subproblem with minimizations in the low-frequency "coarse" one $\mathcal{A}_2$. Note that since the frequency shift may be obtained only by considering the underlying geometrical space an explicit expression in the Fourier basis in unnecessary. Access to the high frequency basis elements may be unavailable as such but is included in the contribution of the complete basis spanning $\mathcal{A}_{12}$ and $\mathcal{A}_1$.

e propose to follow the same approach for the distributed context. e then select a particular minimization method. In our focus example where $y\ x$ is a neural net rst-order training methods such as variants of gradient descent are a natural choice. Remarkably it has been shown in 47 that such methods are signi cantly more e cient for the solution of problems whose solution involves low frequencies. This observation called the "F-principle" is interesting on two accounts. The rst is that it stresses the fact that frequency content is also signi cant for neural net training and the second is that it acts "in the opposite direction" when compared to a multigrid approach low frequencies are favourable instead of being problematic. ne is then led to consider using a Fourier basis also in the new context split the problem into a part containing the high-frequency basis nding a suitable network $y_1\ x_1\ \in \mathcal{A}_1$ and its low frequency part nding a network $y_2\ x_1\ \in \mathcal{A}_2$. and then to shift the frequencies of the subproblems to make them more e ciently solvable. As we will discuss when presenting our numerical examples in ection 5 this can be achieved by using Mscale networks 26 and P 43 . As above this fortunately does not require the explicit problem formulation in the Fourier basis although one needs to be somewhat speci c regarding the subproblems frequency content.

e also note that if the ob ective function f were separable in x_1 and x_2 in the selected basis $\mathcal{B}$ the Fourier basis in our examples then only one of each subproblem minimization would be su cient for solving the overall problem. This is not the case in general but an argument based on quadratic approximation shows that a weak coupling within f between x_1 and x_2 improves the speed of convergence for the block-coordinate minimization.

At each stage of the minimization of f we may therefore compute one or more step s for a subproblem de ned by selecting a subset of variables or equivalently a set of y_i to approximately minimize in a typical block-coordinate descent BCD approach.

Pure cycling between the relevant subproblems is clearly an option and is the strategy most often used in the multigrid case where or cycles are de ned to organise the cycling. Alternatively we may opt for some sort of randomized cycling see 35 31 for the convex case or follow as we choose to do below the procedure used in the RMTR algorithm for the hierarchical case and select a subset of variables for which the expected rst-order decrease in the ob ective function as measured by the norm of the ob ective s gradient with respect to the variables in the subset is su ciently large.

e may therefore consider a simple block-coordinate descent algorithm for minimizing f where we use a second subscript for x to denote iterations numbers and where we have limited

the exposition to the bi-level blocks case.

Algorithm 3.1: Multilevel Optimization (ML-BCD) - Deterministic version

Step 0: Initialization: An initial point $x_{1,0}, x_{2,0} \in \mathbb{R}^n$ a threshold $\tau \in \left(, \frac{1}{\sqrt{2}} \right)$ and a gradient accuracy threshold $\epsilon \in \ , 1$ are given. et k .

Step 1: Termination test. valuate the gradients $\nabla^1_x f\ x_k$ $\nabla^1_{x_1} f\ x_k\ , \nabla^1_{x_2} f\ x_k\ ^T$. Terminate if $\|\nabla^1_x f\ x_k\ \| \leq \epsilon$.

Step 2: Select a subproblem and a subproblem termination rule.
For instance
select i such that $\|\nabla^1_{x_i} f\ x_k\ \| > \tau \|\nabla^1_x f\ x_k\ \|$ and choose to minimize $f\ x_1, x_2$ as a function of x_i.
Also select a termination rule for the chosen subproblem.

Step 3: Approximately solve the chosen subproblem. Apply a rst-order minimization method to the chosen subproblem starting from $x_{1,k}, x_{2,k}$ and iterate for ℓ_k iterations until the selected termination rule is activated. This yields a new iterate $x_{1,k+\ell_k}, x_{2,k+\ell_k}$.

Step 4: Loop. Increment k by ℓ_k and go to tep 1.

In the form stated above the ML-BCD algorithm requires computing the full gradient at every "ma or" iteration i.e. iterations where tep 2 is used at variance with a pure potentially randomized cycling where only the successive subproblem s gradients need to be computed. Thus the number of ma or iterations must remain small compared to the total number of iterations for this approach to be useful.

Also note that the subproblem termination criterion may take di erent forms the number of subproblem iterations may be limited a threshold may be imposed on the norm of the subproblem s gradient and or on the ob ective function decrease below which the subproblem minimization is terminated or any combination of these. In any case it does not make sense to continue the subproblem minimization if the subproblem s gradient becomes smaller that $\frac{\epsilon}{\sqrt{2}}$ $\frac{\epsilon}{\sqrt{\#\text{ of blocks}}}$ in the general case .

The convergence theory for block-coordinate optimization has a long history starting with a famous paper by Powell 32 showing that the method may fail on nonconvex continuously di erentiable functions. hile the theory was further developed for the convex case see the excellent survey 45 and the references therein it was only recently that further progress was made for nonconvex functions overcoming Powell s reservations and that a worst-case complexity analysis implying convergence was produced 1 . The idea is quite simple and rest on the notion of "su cient descent" which requires when using rst-order methods that

$$f\ x_k\ - f\ x_{k+1}\ \geq \kappa \|\nabla^1_x f\ x_k\ \|^2,$$

for all $k \geq$ where κ is a positive constant only depending on f. This su cient descent is in particular guaranteed by the Lipschitz continuity of $\nabla^1_x f$ along the path of iterates $\cup_{k\geq 0}\ x_k, x_{k+1}$ see Notes for ection 2.4 in which case κ is proportional to the inverse of the gradient s Lipschitz constant*. For the sake of completeness we give a simple proof in Appendix A.1 for a

*This assumption is standard in complexity analysis and prevents arbitrary change in the gradient over short steps. In particular, it prevents the occurence of infnite gradients at iterates, which would cause the algorithm to fail. It is more general than assuming bounded gradients, as can be seen by considering convex quadratics. See also the discussion at the end of the present Section.

version of the ML-BCD algorithm using xed-stepsize gradient descent in tep 3 i.e. when for all $k \geq$

$$x_{k+1} \quad x_k - \alpha \overline{\nabla^1_{x_i} f\ x_k} \tag{15}$$

where $\overline{\nabla^1_{x_i} f\ x_k}$ $\nabla^1_{x_1} f\ x_k\ ^T,\ \ ^T$ if i 1 and $,\nabla^1_{x_2} f\ x_k\ ^T\ ^T$ if i 2. This proof rephrases a standard "single block" result see for instance 3 xample 1.2.3 for the BCD case. The key complexity result can be stated as follows.

Theorem 3.1 uppose that f is continuously di erentiable with Lipschitz continuous gradient on the path of iterates $\cup_{k\geq 0}\ x_k, x_{k+1}$ generated according to 15 by the ML-BCD algorithm with xed stepsize and that it is bounded below by f_{low}. uppose also that the stepsize α is small enough to ensure $\alpha < 1/L$ where L is the gradient s Lipschitz constant and that the i-th subproblem is terminated at the latest as soon as $\|\nabla^1_{x_i} f\ x_k\ \| \leq \epsilon/\sqrt{2}$. Then the ML-BCD algorithm requires at most $\kappa_* \epsilon^{-2}$ iterations to produce an iterate x_k such that $\|\nabla^1_x f\ x_k\ \| \leq \epsilon$ where κ_* is a positive constant only depending of α the initial gap $f\ x_0\ - f_{\mathsf{low}}$ and the number of blocks.

ther versions of the algorithm including more elaborate globalization techniques such as linesearches trust-regions or adaptive regularization are also possible and yield the same $\kappa_* \epsilon^{-2}$ complexity bound for di erent values of κ_* we then say that their complexity is $\mathcal{O}\ \epsilon^{-2}$.

e now consider a complexity result for a stochastic variant of ML-BCD in which a xed-stepsize rst-order stochastic gradient G method is used in tep 3 i.e. when for all $k \geq$

$$x_{k+1} \quad x_k - \alpha \overline{g_i\ x_k, \xi_k} \tag{16}$$

where $g_i\ x_k, \xi_k$ is a stochastic estimate of $\nabla^1_{x_i} f\ x_k$ and $\overline{g_i\ x_k, \xi_k}$ $g_i\ x_k, \xi_k\ ^T,\ \ ^T$ if i 1 and $,g_i\ x_k, \xi_k\ ^T\ ^T$ if i 2. The proof is given in Appendix A.2. The result can easily be extended to the case of diminishing step-sizes.

Theorem 3.2 uppose that f is continuously di erentiable with L-Lipschitz continuous gradient on the path of iterates $\cup_{k\geq 0}\ x_k, x_{k+1}$ generated according to 16 by the ML-BCD algorithm and that it is bounded below by f_{low}. Assume that for some $\mu >$ $\sigma_1 \geq$ $\sigma_2 >$ and all $k \in \mathbb{N}$ the stochastic gradient estimate $g_i\ x_k, \xi_k$ satis es

$$\nabla^1_{x_i} f\ x_k\ ^T \mathbb{E}_{\xi_k}\ g_i\ x_k, \xi_k \quad \geq \mu \|\nabla^1_{x_i} f\ x_k\ \|^2 \tag{17a}$$

$$\mathbb{E}_{\xi_k}\ \|g_i\ x_k, \xi_k\ \|^2 \ \leq \sigma_1^2 \quad \sigma_2^2 \|\nabla^1_{x_i} f\ x_k\ \|^2, \tag{17b}$$

and that the xed stepsize is such that $\alpha \leq \mu/L\sigma_2^2$. uppose also that the i-th subproblem is chosen according to the rule $\|\nabla^1_{x_i} f\ x_k\ \| > \tau \|\nabla^1_x f\ x_k\ \|$ for some $\tau \in (\ , 1/\sqrt{2})$ and terminated as soon as $\|g_i\ x_k, \xi_k\ \| \leq c_\epsilon$ with $c_\epsilon > \sqrt{2}\sigma_1$ or after a maximum number of iterations ℓ. Then for all $K \in \mathbb{N}$

$$\mathbb{E}\left(\sum_{k=1, k\in\mathcal{M}}^{K} \frac{\|\nabla^1_x f\ x_k\ \|^2}{K} \right) \leq \frac{2\ f\ x_1\ - f_{\mathsf{min}}}{\alpha\mu K\tau} \quad \frac{\alpha L\sigma_1^2}{\mu\tau} \quad \frac{\ell\sigma_1^2}{\tau}\left(\frac{\alpha L}{\mu} - \frac{1}{\sigma_2^2}\right).$$

where $\mathcal{M}$ is the set of ma or iterations for which tep 2 of ML-BCD is performed.

The assumptions in Theorem 3.2 are standard see 4 for instance except for the condition $c_\epsilon > \sqrt{2}\sigma_1$. This condition is however reasonable because it does not make sense to optimize for

a block beyond the noise level. Note that the last term in the theorem s bound is negative because of the assumption on α. Large values of ℓ will thus improve the bound but but this improvement is limited because the subproblem may also be terminated because of the ϵ-dependent gradient test.

e note that BCD methods have been considered and analysed in a stochastic setting in 2 in which a convergence result is stated based on a bound on the average-squared gradients. This results assumes a distributed context with non overlapping blocks which is slightly less general than the corresponding result for G method proposed in 4 Th. 4. . Interestingly the condition we use to select the blocks allows us to provide a bound that also covers the hierarchical setting. The bound is not directly comparable to that for the standard G because it is established considering ust the subsequence of the ma or iterations.

The situation is more involved and the complexity bound worse when it exists as soon as the function is not Lipschitz continuous on the path of iterates and is for instance discussed for the single block case in 2 3 4 . The recent paper 22 analyzes the complexity of a " rst-order" monotonic descent method applied to a wide class of functions† . The method assumes that one can evaluate for any point x and any direction d the value of $f\ x$ its directional derivative along d $f'\ x,d$ and a "directional subgradient" $G\ x,d$ such that its inner product with d returns $f'\ x,d$. nder these conditions the method achieves Goldstein 12 ϵ-approximate optimality in at most $\mathcal{O}\ \epsilon^{-4}$ such evaluations. nsurprisingly the convergence proof once more relies on showing that it is possible to obtain "su cient descent" in this case given by a multiple of ϵ albeit at a possible cost of $\mathcal{O}\ \epsilon^{-3}$ evaluations. e refer the reader to 22 for details. The monotonic nature of the algorithm then implies as is the case for the simple proof in appendix that su cient descent obtained in the solution of the subproblem before termination translates into su cient descent on the complete problem so that the complexity result obtained by 22 for single block minimization extends to the case where there is a bounded number of blocks.

Interestingly Theorem 3.1 subsumes and simpli es the convergence theory for RMTR 5 15 by recasting this latter method when used with rst-order Galerkin low-level approximations as a trust-region BCD algorithm in the complete space. Also note that the use of Galerkin approximations in this case avoids the need of a "tau correction" or " rst-order coherency" condition 5 15 which is typically requested for less structured low-level approximations.

4 Application to Physics-Informed Neural Networks

Techniques using neural networks have recently emerged as an alternative to classical methods for solving partial di erential equations PD s . In particular the physics-informed neural networks PINNs have raised signi cant interest 27 . Their advantage is their ability to exploit physical knowledge to solve PD s without using simulation data which motivates our choice to illustrate our multilevel approach in this context. ur presentation proceeds in two stages after a brief introduction to PINNs and their multilevel version we rst focus on showing the advantage of alternate training of "coarse" and " ne" networks in the hierarchical context of ection 2 before showing how a re ned use of the frequency content in a distributed context may yield further bene ts.

4.1 Physics-informed neural networks

Given a domain $\subset \mathbb{R}^d$ we consider the following di erential system

$$\begin{cases} \mathcal{L}\ u\ z \quad\quad r\ z & \text{in} \\ \mathcal{B}\ u\ z \quad\quad g\ z & \text{on} \end{cases} \qquad 1$$

where is the boundary of $\mathcal{L}$ and $\mathcal{B}$ are two possibly nonlinear di erential operators and r and g are two given functions.

† Technically, the objective function must be bounded below, have a bounded directional subgradient map and a nite nonconvexity modulus.

PINNs approximate the solution of the problem by a su ciently smooth neural network $y\ x$ $\mathbb{R}^d \to \mathbb{R}$. The neural network is trained by minimization of a loss that takes into account the physical information contained in the PD . peci cally denoting $Z_\Omega \quad \{z_\Omega^j\}_{j=1}^{N_\Omega}$ $Z_\Gamma \quad \{z_\Gamma^j\}_{j=1}^{N_\Gamma}$ a set of training points sampled in and respectively the loss function is de ned as

$$f\ x \quad \frac{\lambda_\Omega}{N_\Omega}\sum_{j=1}^{N_\Omega}\mathcal{L}\ y\ x\ \ z_\Omega^j \quad -r\ z_\Omega^j\ ^2 \quad \frac{\lambda_\Gamma}{N_\Gamma}\sum_{j=1}^{N_\Gamma}\mathcal{B}\ y\ x\ \ z_\Gamma^j \quad -g\ z_\Gamma^j\ ^2, \qquad 1$$

where λ_Ω λ_Γ are positive weights which balance the contribution of the residual of the PD and the residual of the boundary conditions. The di erentiability properties of the neural networks are exploited to explicitly compute the di erential operators $\mathcal{L}\ y\ x$ and $\mathcal{B}\ y\ x$ which are then evaluated on the set of training points.

ur ob ective in what follows is to use several PINNs in con unction with the ML-BCD algorithm broadly mimicking multigrid methods in the hope of obtaining similar computational advantages. e de ne multi-level PINN MPINNs as follows.

Consider a feedforward neural networks with $N-1$ hidden layers. Let $d_j \in \mathbb{N}$ be the number of hidden neurons in the i-th hidden layer for $j \quad 1, ..., N-1$ and let d_0 and d_N be the number of neurons of the input and the output layers respectively. Let $W^j \in \mathbb{R}^{d_j \times d_{j-1}}$ be the matrix of weights between the $j-1$ -th and the j-th layers for $j \quad 1, \ldots, N$. e denote the set of all such networks by $H^{N,\{d_j\}}$. Assuming a method is selected to minimize the loss functions 1 we may now use the ML-BCD algorithm by selecting our neural network as

$$y\ x\ \ z \quad \sum_{i=1}^{s} y_i\ x_i\ \ z \qquad 2$$

with $y_i \in H^{N_i,\{d_{j,i}\}}$.

4.2 Alternating training in a hierarchical context

e start by considering the question of whether imitating the multigrid approach of alternating between a coarse and a ne grid can be computationally advantageous.

4.2.1 Test problems

Inspired by 34 we perform our experiments on several instances of the Poisson problem with source term r in the domain and Dirichlet Neumann conditions on its boundary . Moreover we assume that contains a closed embedded boundary $_i$ dividing in $_e$ the exterior of $_i$ and $_i$ the interior of $_i$ so that the complete boundary is given by $_e \bigcup\ _i$. The problem is thus stated as

$$\begin{cases} u\ z \quad r\ z \ \text{in} \\ au\ z \quad b\dfrac{\partial u}{\partial \zeta}\ z \quad g\ z \ \text{on}\ _e \\ cu\ z \quad d\dfrac{\partial u}{\partial \zeta}\ z \quad h\ z \ \text{on}\ _i \end{cases} \qquad 21$$

where $\frac{\partial u}{\partial \zeta} \quad \nabla u^T \zeta$ with ζ being the boundary normal vector pointing into . The domain for the test problem is the square $-1, 1\ ^2$ with an embedded circle of radius $R \quad .5$ centered at the origin de ning $_i$. e consider Dirichlet boundary condition on $_e$ and $_i$. e choose r to ensure that exact solution is

$$u\ z \quad cos\ \alpha\pi z_1 \quad \pi z_2 \quad cos\ \pi z_1 \quad \beta\pi z_2$$

where α and β are integers de ning the frequency content of the solution.

4.2.2 Networks architectures

In this rst set of experiments we simplify 2 to

$$y\ x\ \ z \qquad y_1\ x\ \ z \qquad y_2\ x\ \ z$$

where $y_1 \in H^{3,\{d_{j,1}\}}$ is the " ne" network and $y_2 \in H^{3,\{d_{j,2}\}}$ is the "coarse" one. In accordance with our earlier discussion we choose y_2 smaller than y_1 in the sense that y_2 is an independent copy of a subnetwork of y_1. Thus our setting is hierarchical in the sense of ection 2 and $\mathcal{A}_2 \subset \mathcal{A}_1 \quad \mathcal{A}_{12}$.

e consider three di erent MPINNs corresponding to di erent sizes of the coarse network while keeping the size of the ne network xed. e compare them to a standard PINN network of size approximately equal to the total number of parameters in the ne and coarse networks as well as a network of the same size as the ne network.

xperiment s name	Network s size d_1, d_2, d_3 Fine	Coarse	Number of parameters
ML1	14 14 14	14 14 14	4 4 4 4
ML2	14 14 14	1 1 1	4 4 2 6
ML3	14 14 14	7 7 7	4 4 1 22
L1	14 14 14		4 4
L2	2 2 2		1 2

Table 1 Network architecture for the experiments with alternating training in the hierarchical context

Table 1 details the ve architectures tested in our experiments and the size of the involved networks ML1 ML2 and ML3 are the multilevel ones training two networks using the ML-BCD algorithm and L1 and L2 are single level ones standard training of a single network provided for comparison. All hidden layers use the tanh activation function thereby ensuring the necessary di erentiability properties.

4.2.3 Training setup

Training points are uniformly sampled using the Latin hypercube sampling in and ∂ . ere we have chosen to use the same grid to train the coarse and the ne networks with $N_\Omega \quad 5$ points sampled in the domain and $N_\Gamma \quad 4$ points sampled on the boundary. These values have been chosen by performing a grid search on the optimization of the network with classical training.

At each epoch we use a random subset of these points composed of 2 inner points and 5 boundary points. e have chosen the coe cients $\lambda_\Omega \quad \lambda_\Gamma \quad 1$ to weight internal and boundary losses.

To evaluate the accuracy of the di erent models we consider a set of testing points $\{z^t\}_{t=1}^T$ with $T \quad 3$ randomly chosen using the Latin hypercube sampling in $_e$ and consider the mean squared error given by

$$MSE \quad \frac{1}{T}\sum_{t=1}^{T} y\ x\ \ z^t\ - u\ z^t\ \ ^2$$

All networks are trained with Adam optimizer 21 . The initial learning rate is set at $2 \times 1\ ^{-4}$ with an exponential decay of . at each epoch for all networks. All our codes are implemented in TensorFlow2 version 2.2. and run with a single N IDIA GeForce GT 1 Ti.

It is important to notice that using an alternate training strategy requires special care when setting the hyperparameters of the optimizer. Indeed each time we select a sub-network to train the optimization problem also changes. To ensure a fair and coherent comparison with standard network training procedures we decided to maintain the current learning rate when restarting the

solver. This allowed us to maintain the property that the learning rate gradually decreases over time which is a common technique used to prevent the model from overshooting the optimal values and enhances convergence. For other parameters we decide to use Adam as a black box and to restart it each time we change subproblem that is we do not transfer the speci c hyperparameters that are speci c to this optimization method such as momentum. This means that we have to initialize the optimizer from scratch with default hyperparameters except for the learning rate for each subproblem. The consequence of such a restart is a peak in the loss at the rst iteration of the optimization process which is a common behavior when initializing an optimizer with random or default values. owever it remains an open question to nd the best strategy to tune these hyperparameters is it better to restart the optimizer or to nd a good way to transfer the parameters from one subproblem to the other e also estimate the number of oating-point operations performed as the number of FL Ps required for the matrix-vector product operations during the forward pass. For MPINNs it takes into account both the operations performed at ne and at coarse levels. For instance for a network with two layers d_h neurons in the layers an output size of 1 and N training points of size d the number of FL Ps for the forward pass would be computed as

$$\text{FL Ps} \quad N \times \ 2d \times d_h \quad d_h^2 \quad d_h \times 1$$

e select the subnetwork to train at each cycle according to tep 2 of the ML-BCD algorithm. Moreover we chose to terminate the subproblem training after a xed number of epochs on each problem see numerical results . e also chose α 2 and β 4. For each experiment we alternately performed 2 epochs on y_1 and 2 epochs on y_2.

4.2.4 Results

The results of the test are reported in Figures 1 and 2. e see that at equivalent computational cost MPINNs MLs converge signi cantly faster than conventional PINNs Ls . It is worth noting that the PINN with fewer parameters L1 converges faster than the larger one L2 . The choice of the coarse network s dimension also seems to a ect the speed of convergence smaller coarse networks yielding better results in these tests.

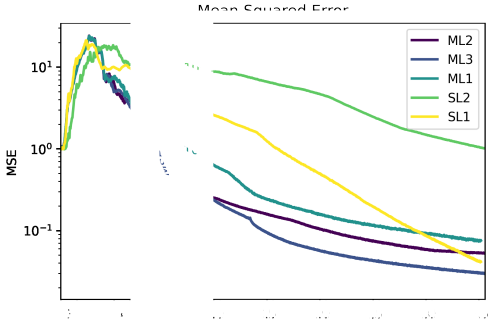

Figure 1 volution of the M as a function of the computational cost for our di erent models

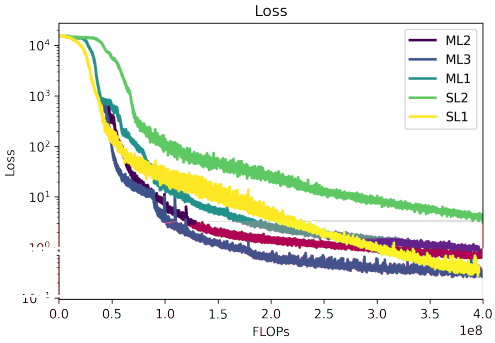

Figure 2 volution of the loss as a function of the computational cost for our di erent models

These result may seem encouraging but this approach remains limited in that it does not overcome a limitation inherent to classical neural network training high-frequency tting. This is highlighted in Figure 3 where we also test our method on problem 21 with parameters α 2 and β 2 .

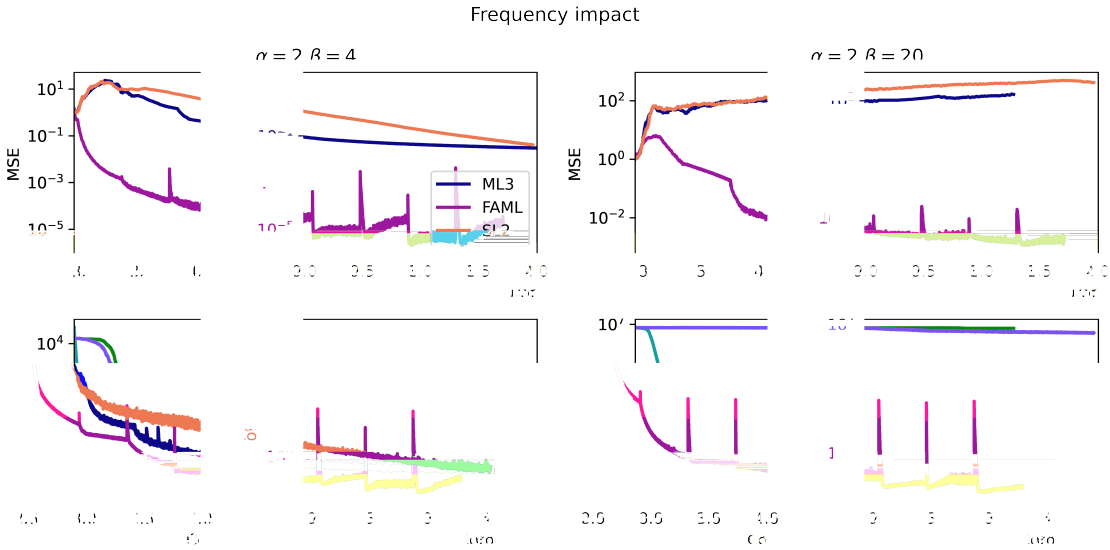

Figure 3 Results of our method on the Poisson problem 21 for di erent frequencies. The method indicated as FAML Frequency-Aware-Multilevel is the method proposed in the forthcoming ection 5.

An alternative approach is thus warranted which we develop in the next section.

5 A "frequency-aware" distributed approach

In the hierarchical cases detailed in ection 2 the reduction in computational cost is typically directly proportional to the separation of frequencies. For instance in multigrid methods switching to a coarse grid helps to target low frequencies that converge slowly in the ne problem. This does not seem to be the case for neural networks. The F-Principle 47 states that "deep neural networks often t target functions from low to high frequencies during the training process" which is problematic in some cases and seems to a ect also MPINNs leading to a slow convergence as illustrated at the end of the previous section. The simple alternation of coarse and ne problems doesn t seem to be su cient in the context of neural networks.

everal papers have addressed the issue related to the F-Principle by proposing architectures that transform high frequencies of the problem into low frequencies thereby allowing a more e cient use of the neural networks. Inspired by these papers we propose a frequency-aware architecture that retains the computational cost savings of multilevel training while incorporating the frequency separation aspect of classical methods.

5.1 A frequency-aware network architecture

ur architecture proposal is mainly inspired by Multi-scale deep neural networks MscaleDNNs 26 and P 43 which were designed to mitigate the e ect of the F-principle in the standard single level context. The basic idea is to transform high frequency learning in low frequency learning and to facilitate the separation of the frequency contents of the target function. As a result these networks show uniform convergence over multiple scales.

The MscaleDNNs architecture achieves this ob ective thanks to two main ingredients

- radial scaling in the frequency domain the rst layer is separated into N parts each receiving a di erently scaled input. everal variants of MscaleDNNs have been proposed we choose to focus here on the most e cient one which uses parallel sub-networks each dedicated to a speci c input scaling.

- the use of wavelet-inspired and frequency-located activation functions. These functions with compact support have good scale separation properties.

P networks were proposed by ang ang and Perdikaris. They use the same principle of input scaling combined with soft Fourier mapping FM de ned as

$$\gamma\ z \quad s \begin{bmatrix} \cos\ z \\ \sin\ z \end{bmatrix}$$

with $s \in \ ,1$ a relaxation parameter.

A "Fourier features network" was rst proposed in 3 which uses a random Fourier features mapping γ as a coordinate embedding of the inputs followed by a conventional fully-connected neural network. This method did mitigate the pathology of spectral bias and helped networks learn high frequencies more e ectively. Recent developments have made it possible to model fully connected neural networks and thus PINNs as kernel regression. The authors of 3 suggested that using Fourier mapping a ects the width of the kernels and thus the network s capacity to capture high frequencies. This idea was extended to PINNs in 43 with convincing results provided that the xed scalings are too far from the frequencies contained in the solution.

Inspired by their success we now propose an architecture that combines the positive features of both methods. ur architecture s output is the sum of several networks that use di erent scaling vectors specializing each network for di erent frequency scales matching the structure presented in our theory in ection 2. To mitigate possible convergence problems arising from a bad choice of the xed scalings we have chosen to add learnable weights to the input scaling integers using FM as a classic activation function.

For low frequency resolution we choose a classical network using hyperbolic tangent activation functions. For the other networks we use FM activation functions for the rst layer associated with a scaling from a centred normal distribution whose variance grows with the targeted frequencies. The other layers of the networks use hyperbolic tangent activation functions thus ensuring our di erentiability requirements. e refer to this architecture as Parallel- P P- P . nfortunately the wavelet-based and frequency-based activation functions utilized in the M CAL are not continuously di erentiable and thus fail to satisfy our theoretical assumptions. e have however included some successful experiments conducted using these functions in Appendix B.

A similar architecture based on a lower number of subnetworks has been proposed in 25 for the deep Ritz method 1 which produces a variational solution to problem 1 . To the best of our knowledge this is the rst time this architecture has been proposed for PINNs.

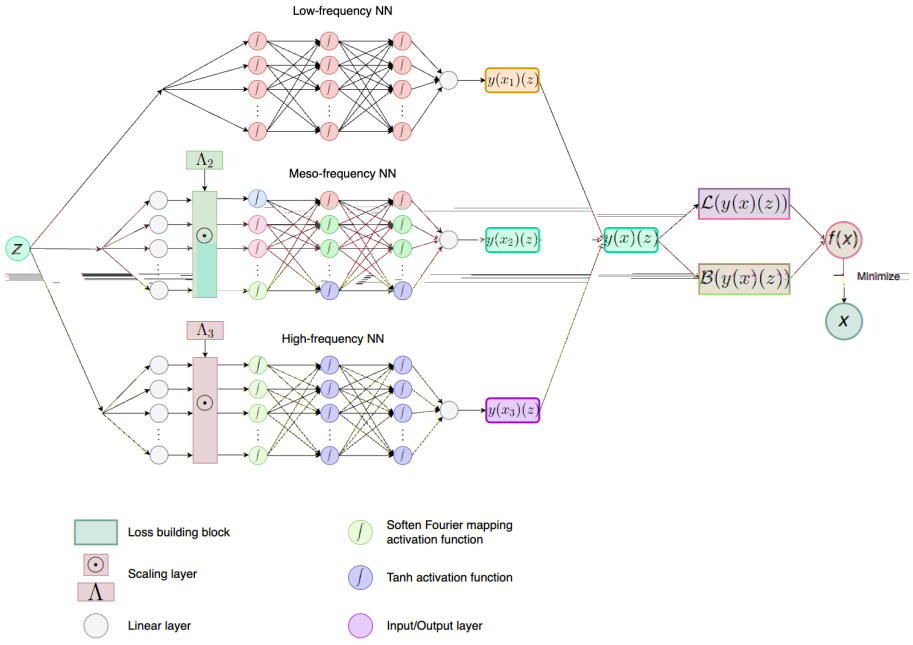

Figure 4 An example of P- P architecture

An example of our architecture is illustrated in Figure 4 which consists of three sub-networks targeting three di erent frequency ranges de ned by their input scaling $_i$. The lowest frequency network employs tanh activation functions while the others use FM and tanh.

To be e ective this architectures must cover most of the frequencies contained in the a priori unknown PD s solution. Therefore a broad range of frequencies must be covered employing a large number of neurons as the target frequencies are de ned by the input scalings. As a result the considerable improvement in accuracy provided by parallel frequency-aware architectures is obtained at the price of a signi cant increase in the training s computational cost making it an ideal candidate for our multilevel approach.

These architectures can be easily incorporated into our framework where the global network is de ned as a sum of y_i sub-networks each responsible for a frequency range de ned by its input scaling. Because of this latter characteristics the resulting approach now belongs to the distributed context discussed in ection 2. In particular we refer to the combination of the P- P network with the ML-BCD algorithm as the FAML Frequency Aware Multi-Level method. e use the same selection and termination criteria as in ection 4.2. hen several sub-networks are available the network with the highest ratio $\|\nabla^1_{x_i} f\ x_k\ \|/\|\nabla^1_x f\ x_k\ \|$ is chosen.

6 Numerical results

e illustrate the performance of the proposed method on some Poisson problems and on the heat equation. Further tests performed with a slightly di erent frequency-aware architecture are presented in Appendix B.

6.1 Poisson problems

e now illustrate the e ciency of the FAML method on some examples of the form 21 de ned in 34 . e rst describe the test problems themselves then specify the considered network architectures and the training setup before describing and commenting the obtained results.

6.1.1 Test problems

Test problem 1: Circle embedded in a square domain The domain for the rst problem is the square $-1,1^2$ with an embedded circle of radius R .5 centered at the origin de ning $_i$. e consider Neumann boundary conditions on $_i$ and Dirichlet boundary condition on $_e$ thus a c 1 and b d .

The source term and boundary term are given by

$$\begin{cases} r\ \rho,\omega & D\ k^2-n^2\ \rho^{k-2}\sin n\omega \ \text{ in} \\ g\ \rho,\omega & -\dfrac{Dk}{n}\left(\dfrac{\rho}{R}\right)^{(n-k)}\rho^k\sin n\omega \quad Rq\log\rho \ \text{ on } \ _e \\ h\ \rho,\omega & 1 \ \text{ on } \ _i \end{cases} \tag{22}$$

where ρ,ω are the polar coordinates in the plane $n\in\mathbb{Z}$ $z\in\mathbb{N}$ and D $\sqrt{2}^{-\max(k,n)}$. e choose k 1 and n -5. The solution is given by $u\ \rho,\omega$ $-\frac{Dk}{n}\left(\frac{\rho}{R}\right)^{(n-k)}\rho^k\sin n\omega$ $R\log\rho$. The solution and the source terms are depicted in Figure 5.

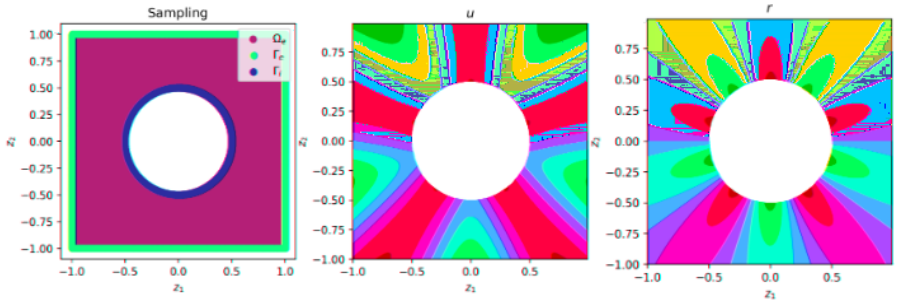

Figure 5 Test problem 1 cf. 21 22 . Left the domain and its subdomains. Center the solution u. Right the source term r.

Notice the variation in angular frequency as a function of the radius.

Test problem 2: Four-lobe structure The domain for the second problem is the unit square $,1^2$ with an embedded surface de ned by $\rho\ \omega$ R_m $R_d\cos 4\omega$ with R_m . 3 5 R_d .117. e consider Dirichlet boundary conditions for both $_e$ and $_i$. The source term and boundary term are given by

$$\begin{cases} r\ \rho,\omega & 12\ 1\ \rho^2-1\ e^{-10^2}\quad \displaystyle\sum_{k=1}^{4}4\quad 1\ r_k^2-1\ \text{ in} \\ g\ \rho,\omega & .3e^{-10\rho^2}\quad \displaystyle\sum_{k=1}^{4}e^{-10r_k^2}\ \text{ on } \ _e\cup\ _i \\ r_k & \sqrt{x\pm\ .45^{\ 2}\quad y\pm\ .45^{\ 2}},\ k\quad 1,\ldots,4 \end{cases} \tag{23}$$

where ρ and ω are the polar coordinates and x and y the corresponding Cartesian coordinates. The solution is given by $u\ \rho,\omega$ $.3e^{-10\rho^2}$ $\sum_{k=1}^{4}e^{-10r_k^2}$. The solution and the source terms are depicted in Figure 6.

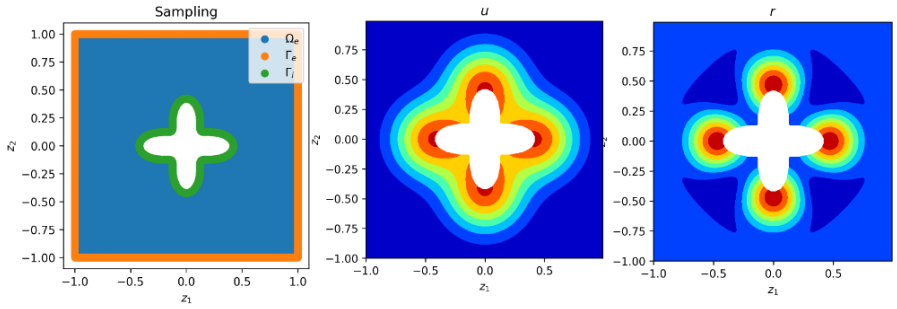

Figure 6 Test problem 2 cf. 21 23 . Left the domain and its subdomains. Center the solution u. Right the source term r.

Test problem 3: Annulus domain with homogeneous source e now consider a domain de ned by an annulus with inner radius R_i .25 and outer radius R_o .75 and consider a Neumann boundary condition on $_i$ and a Dirichlet boundary condition on $_e$. The source term and boundary term are given by

$$\begin{cases} r\ \rho,\omega & \text{in} \\ g\ \rho,\omega & \text{on}\ _e \\ h\ \rho,\omega & 1\ \text{on}\ _i \end{cases} \tag{24}$$

where ρ and ω are the polar coordinates. The solution is given by $u\ \rho,\omega$ $R_i \log \frac{\rho}{R_0}$ which is depicted in Figure 7.

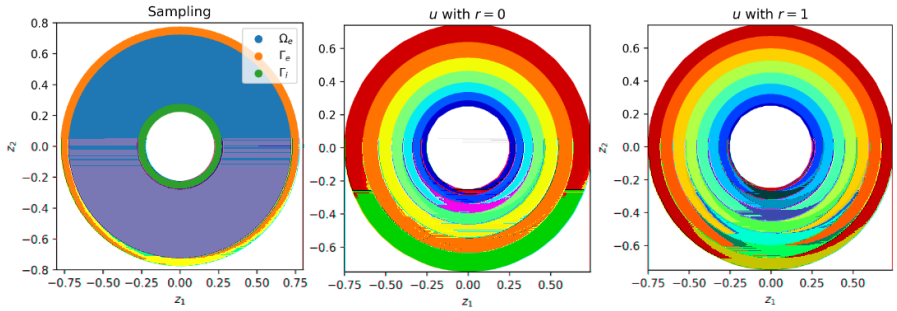

Figure 7 Test problems 3 cf. 21 24 and 4 cf. 21 25 . Left the domain and its subdomains. Center the solution u for Test Problem 3. Right the solution u for Test Problem 4.

Test problem 4: Annulus domain with inhomogeneous source e nally consider again the domain de ned by an annulus with inner radius R_i .25 and the outer radius R_o .75 with Neumann boundary condition on $_i$ and Dirichlet boundary condition on $_e$. The source and boundary terms now given by

$$\begin{cases} r\ \rho,\omega & 1\ \text{in} \\ g\ \rho,\omega & \text{on}\ _e \\ h\ \rho,\omega & 1\ \text{on}\ _i. \end{cases} \tag{25}$$

The solution is then given by $u\ \rho,\omega$ $\frac{\rho^2-R_0^2}{4}$ $R_i\ 1-\frac{R_i}{2}\ \log \frac{\rho}{R_o}$.

6.1.2 Network architecture

e now give the details of the network architectures used in our second set of experiments. It is important to notice that our method is not ust applicable to P- P networks but can be used

to train any architecture composed of sum of subnetworks.

	input scaling	First Activation	ther Activations
subnetwork 1	None	tanh	tanh
subnetwork 2	$\mathcal{N}$, 2	FM .5	tanh
subnetwork 3	$\mathcal{N}$, 4	FM .5	tanh
subnetwork 4	$\mathcal{N}$, 6	FM .5	tanh

ach of the subnetworks is composed of 3 hidden layers of 1 neurons each. In order to assess computational performance we also consider the standard single level training applied on the complete network.

6.1.3 Training setup

e will use the same setup as in ection 4 except for the coe cients λ_{Ω_e} 1, λ_{Γ_e} 1 and λ_{Γ_i} 1 that weight internal and boundary losses. These values where chosen by a grid search on the optimization of the network with classical training.

e consider as a unit of cost the cost of optimising our complete neural network over an epoch. hen a subnetwork is selected in the ML-BCD algorithm the unselected part of the complete network is cached in memory and does not contribute to the subnetwork optimization cost. ince the cost per iteration is linear in the number of parameters when using rst-order methods the cost of optimizing for an epoch one of four subnetworks of identical sizes is $\frac{1}{4}$ .25 cost units. For these experiments we chose to do 1 epochs in full network cycle and 4 epochs in subnetwork cycle. Thus the cost of the full and partial training is similar. e do a total of cycles with 1 epochs on the full network to start the training for a total computational cost of 1 $\times$ 1 $\frac{4000}{4}$ 1 units.

For each problem we rst compute the curve of the median values of the loss over 1 runs and of the M as a function of epochs and for two di erent computational budgets 1 k and 1 k units . e then select the lowest median loss obtained for a given budget and record the associated median M .

6.1.4 Numerical results

Table 2 reports the results ust described obtained with the Frequency-Aware-Multilevel FAML cf. also Figure 1 .

Problem	Budget	M	M FAML	Loss	Loss FAML
Test pb 1	1	1.4 - 5	2.54 - 7	1.64 - 1	1.1 - 2
	1	7.7 - 6	2.4 - 7	6.33 - 2	4. - 3
Test pb 2	1	1.66 - 5	3. - 7	1.45 - 1	1.17 - 2
	1	7.77 - 6	2.57 - 7	5.3 - 2	4.42 - 3
Test pb 3	1	1. 5 - 5	1.36 - 7	1. 4 - 1	6.76 - 3
	1	3. 1 - 6	1.11 - 7	3.71 - 2	2.55 - 3
Test pb 4	1	.53 - 6	1.62 - 7	1. 2 - 1	7.4 - 3
	1	3.5 - 6	1.32 - 7	3.75 - 2	2.63 - 3

Table 2 Best value of the median loss for the given budget and associated M for standard single-level and FAML training 1 independent runs

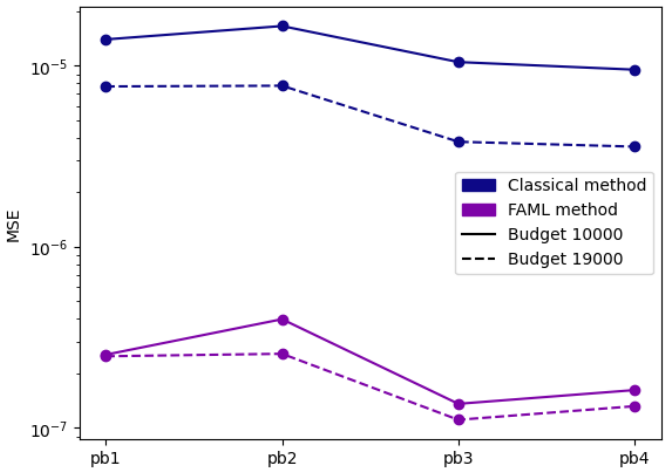

Figure Comparison of average M values from Table 2

In each case the proposed multi-level training yields lower losses than the standard one. The di erences are particularly signi cant for a small computational budget where the improvement is of several orders of magnitude. The multi-level training also always results in a lower associated M .

To provide further insight we nally provide a typical example allowing the comparison of standard and multi-level training for Test problem 3 we illustrate in Figure the decrease of the total loss and of its boundary and interior components as well as that of the M as a function of computational cost. The multi-level approach clearly outperforms the standard one. e remind the reader that peaks correspond to the restarts of the optimizer.

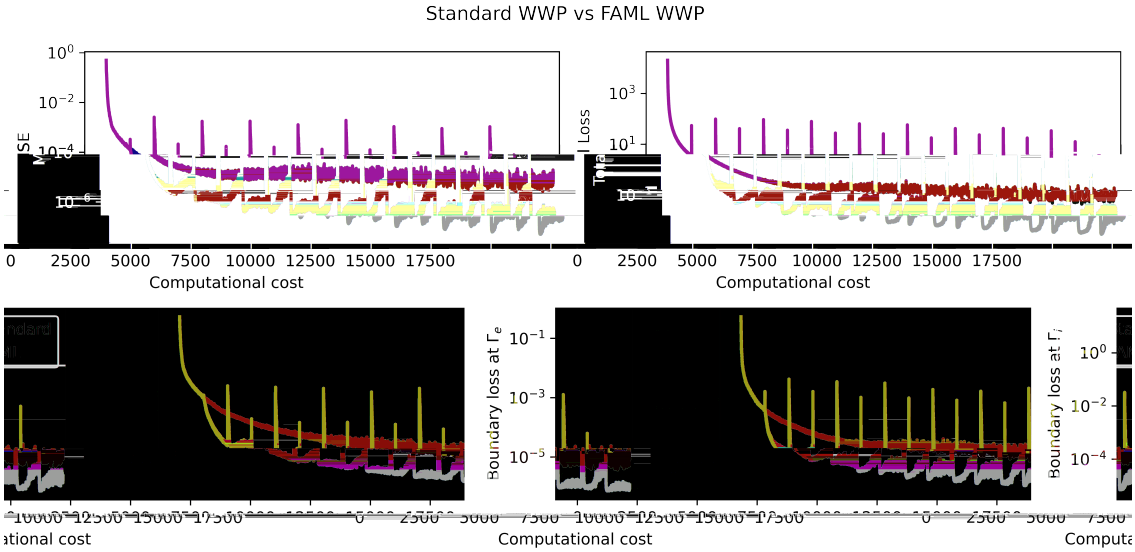

Figure volution of M and loss values for a typical run on Test problem 3

6.2 The heat equation

e next consider a non-elliptic heat equation. Given a positive parameter ω the problem is stated as

$$\begin{cases} \dfrac{\partial u\ z,t}{\partial t} \quad \dfrac{1}{\omega\pi\ ^2}\dfrac{\partial^2 u\ z,t}{\partial z^2} \text{ with } z,t \in \ ,1 \times \ ,1 , \\ u\ z, \quad \sin \omega\pi z\ ,\ z \in \ ,1 , \\ u \ ,t \quad u\ 1,t \quad ,\ t \in \ ,1 , \end{cases} \tag{26}$$

and its exact solution is given by

$$u\ z,t \quad e^{-t}\sin \omega\pi z\ .$$

e consider di erent test problems for values of ω $1,2\ ,5$. As for the Poisson problems we use P- P networks with the following architecture.

	input scaling	First Activation	ther Activations
subnetwork 1	None	tanh	tanh
subnetwork 2	$\mathcal{N}\ ,5$	FM .5	tanh
subnetwork 3	$\mathcal{N}\ ,1$	FM .5	tanh

ach of the subnetworks is composed of 3 hidden layers of 5 neurons each. The training setup remains the same. In this case as we have three subnetworks the cost of one epoch in subnetworks training is .33 cost units. e report the result of our method in Table 3 cf. also Figure 1 .

Problem	Budget	M	M FAML	Loss	Loss FAML
ω 1	1	5.64 - 6	1.4 - 6	1.1 - 1	1.65 - 2
	1	.1 - 7	2. - 7	5.23 - 3	4.5 - 3
ω 2	1	5.5 - 4	5.1 - 5	1. 7 - 2	.6 - 3
	1	1.77 - 5	7. 7 - 6	2. - 3	2.45 - 3
ω 5	1	1.13 - 1	1.13 - 2	1. 1	1.1 - 1
	1	2.5 - 2	2.55 - 4	2. - 1	1.35 - 2

Table 3 Best median of the loss for the given budget and associated M for standard single-level and FAML training 1 independent runs

As in the case for the elliptic equation our method yields a faster decrease of the M in the rst part of the training as compared to the classical training. This is true for all values of ω. Note that much lower errors are obtained at the end of the training and for high values of ω.

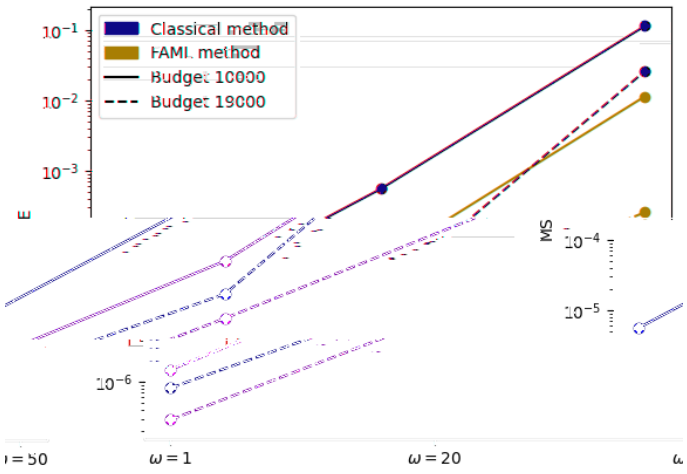

Figure 1 Comparison of average M values from Table 3

7 Conclusion and perspectives

e rst developed a new point of view on multilevel optimization methods arguing they can be seen as block-coordinate minimization methods in a higher dimensional space. Distinguishing two contexts hierarchical and distributed we reformulated a class of multilevel algorithms in block form and showed how convergence results for block-coordinate methods can be applied to the multilevel case.

e next illustrated this approach for the approximate solution of partial di erential equations using physics-informed neural networks as a block solver and presented numerical experiments with a method using pure alternating training in the hierarchical context and a more elaborate frequency-aware technique in the distributed one on complex Poisson and heat propagation problems. n these problems both proposed multilevel PINN methods consistently produced lower losses than those obtained using conventionally trained networks. Convergence was also shown to be much faster resulting in a considerable reduction of the computational cost to obtain good solutions often better than those obtained by a classical training .

hile these initial results are encouraging more research remains desirable for a better understanding of the algorithms. Their dependence on techniques to transfer algorithmic hyperparemeters between levels is of particular interest. Applying our approach to more general operator learning is also worth further investigation. Finally we plan to develop a more e cient implementation of the method in terms of GP memory management.

Declarations

7.0.0.1 Funding ork partially supported by 3IA Arti cial and Natural Intelligence Toulouse Institute ANITI French "Investing for the Future - PIA3" program under the Grant agreement ANR-1 -PI3A- 4 and by the GDR I I pro ect M MIG .

7.0.0.2 Data availability statement The tests presented in this manuscript do not rely on any particular dataset and they can be reproduced by the information provided in the numerical results sections.

References

1 . . Amaral R. Andreani . G. Birgin D. . Marcondes and . M. Mart nez. n complexity and convergence of high-order coordinate descent algorithms for smooth nonconvex box-constrained minimization. ournal of Global ptimization 4 3 527 561 2 22.

2 tefania Bellavia Francesco Della anta and Alessandra Papini. Alternate training of shared and task-speci c parameters for multi-task neural networks. ar iv preprint ar iv 2312.1634 2 23.

3 A. Borzi and . unisch. A globalisation strategy for the multigrid solution of elliptic optimal control problems. ptimization Methods and oftware 21 3 445 45 2 6.

4 Leon Bottou Frank Curtis and orge Nocedal. ptimization methods for large-scale machine learning. IAM review 6 2 223 311 2 1 .

5 . L. Briggs . . enson and . F. McCormick. A Multigrid Tutorial. IAM Philadelphia A 2nd edition 2 .

6 . Cai . Mao . ang M. in and G. . arniadakis. Physics-informed neural networks PINNs for uid mechanics A review. Acta Mechanica inica 37 12 1727 173 2 21.

7 . Calandra . Gratton . Riccietti and . asseur. n high-order multilevel optimization strategies. IAM ournal on ptimization 31 1 3 7 33 2 21.

C. Cartis N. I. M. Gould and Ph. L. Toint. valuation complexity of algorithms for nonconvex optimization. Number 3 in M - IAM eries on ptimization. IAM Philadelphia A une 2 22.

. Chung . Ahn and . Bengio. ierarchical multiscale recurrent neural networks. ar iv 16 . 17 4 2 16.

1 B. u et al. The deep Ritz method a deep learning-based numerical algorithm for solving variational problems. Communications in Mathematics and tatistics 6 1 12 2 1 .

11 G. Farhani A. azachek and B. ang. Momentum diminishes the e ect of spectral bias in physics-informed neural networks. ar iv 22 6.14 62 2 22.

12 A. A. Goldstein. ptimization of Lipschitz continuous functions. Mathematical Programming eries A 13 1 14 22 1 77.

13 . Gratton A. artenaer and Ph. L. Toint. n recursive multiscale trust-region algorithms for unconstrained minimization. In F. arre C. Lemarechal and . owe editors berwolfach Reports ptimization and Applications 2 5.

14 . Gratton A. artenaer and Ph. L. Toint. econd-order convergence properties of trust-region methods using incomplete curvature information with an application to multigrid optimization. ournal of Computational and Applied Mathematics 24 6 676 6 2 2 6.

15 . Gratton A. artenaer and Ph. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. IAM ournal on ptimization 1 1 414 444 2 .

16 Ch. Gross and R. rause. n the globalization of A PIN employing trust-region control strategies - convergence analysis and numerical examples. Technical Report 2 11- 3 niversita della vizzera italiana Lugano C 2 11.

17 . aber and L. Ruthotto. table architectures for deep neural networks. Inverse Problems 34 1 14 4 2 17.

1 . aber L. Ruthotto . oltham and .- . un. Learning across scales multiscale methods for convolution neural networks. In Thirty- econd AAAI Conference on Arti cial Intelligence pages 3142 314 2 1 .

1 . ackbusch. Multi-grid Methods and Applications. Number 4 in pringer eries in Computational Mathematics. pringer erlag eidelberg Berlin New ork 1 5.

2 M. I. ordan T. Lin and M. ampetakis. n the complexity of deterministic nonsmooth and nonconvex optimization. ar iv 22 .124 3 2 22.

21 D. ingma and . Ba. Adam A method for stochastic optimization. In Proceedings in the International Conference on Learning Representations ICLR 2 15.

22 . ong and A. Lewis. The cost of nonconvexity in nonconvex nonsmooth optimization. ar iv 221 . 652 2 22.

23 A. opanicakova and R. rause. Globally convergent multilevel training of deep residual networks. IAM ournal on cienti c Computing 254 2 2 22.

24 .-A. Li. A multi-scale DNN algorithm for nonlinear elliptic equations with multiple scales. Communications in Computational Physics 2 5 1 6 1 6 2 2 .

25 .-A. Li .- . . u and L. hang. ubspace decomposition based DNN algorithm for elliptic type multi-scale PD s. ar iv 2112. 666 2 21.

26 . Liu . Cai and .- . . u. Multi-scale deep neural network MscaleDNN for solving Poisson-Boltzmann equation in complex domains. Communications in Computational Physics 2 5 1 7 2 1 2 2 .

27 P. Perdikaris M. Raissi and G. . arniadakis. Physics-informed neural networks A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. ournal of Computational Physics 37 6 6 7 7 2 1 .

2 . Mishra and R. Molinaro. stimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PD s. IMA ournal of Numerical Analysis 42 1 1 22 2 22.

2 . G. Nash. A multigrid approach to discretized optimization problems. ptimization Methods and oftware 14 116 2 .

3 . Nesterov. Introductory Lectures on Convex ptimization. Applied ptimization. luwer Academic Publishers Dordrecht The Netherlands 2 4.

31 . Nesterov. ciency of coordinate descent methods for huge-scale optimization problems. IAM ournal on ptimization 22 341 362 2 12.

32 M. . D. Powell. n search directions for minimization algorithms. Mathematical Programming 4 1 3 2 1 1 73.

33 N. Rahaman A. Baratin D. Arpit F. Draxler M. Lin F. amprecht . Bengio and A. Courville. n the spectral bias of neural networks. In International Conference n Machine Learning pages 53 1 531 2 1 .

34 N. R. Rapaka and R. amtaney. An e cient Poisson solver for complex embedded boundary domains using the multi-grid and fast multipole methods. ournal of Computational Physics 41 1 3 7 2 2 .

35 Peter Richtarik and Martin Takac. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Mathematical Programming 144 1-2 1 3 2 14.

36 B. Ronen D. acobs . asten and . ritchman. The convergence rate of neural networks for learned functions of di erent frequencies. Proceedings of the 33rd International Conference on Neural Information Processing ystems page 4761 4771 2 1 .

37 . hin. n the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PD s. Communications in Computational Physics 2 5 2 42 2 74 2 2 .

3 Matthew Tancik Pratul rinivasan Ben Mildenhall ara Fridovich- eil Nithin Raghavan tkarsh inghal Ravi Ramamoorthi onathan Barron and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing ystems 33 7537 7547 2 2 .

3 L. Tian and A. Man-Cho o. Computing Goldstein ϵ, δ -stationary points of Lipschitz functions in $\mathcal{O}\left(\epsilon^{-3}\delta^{-1}\right)$ iterations via random conic perturbation. ar iv 2112. 2 2 21.

4 . Trottenberg C. . osterlee and A. chuller. Multigrid. lsevier Amsterdam The Netherlands 2 1.

41 . Tsung- ei M. Maire and . . u. Multigrid neural architectures. In Proceedings of the I Conference on Computer ision and Pattern Recognition pages 6665 6673 2 17.

42 C. von Planta A. opanicakova and R. rause. Training of deep residual networks with stochastic MG PT. ar iv 21 . 4 52 2 21.

43 . ang . ang and P. Perdikaris. n the eigenvector bias of Fourier feature networks From regression to solving multi-scale PD s with physics-informed neural networks. Computer Methods in Applied Mechanics and ngineering 3 4 113 3 2 21.

44 ifan ang inling u and Paris Perdikaris. hen and why pinns fail to train A neural tangent kernel perspective. ournal of Computational Physics 44 11 76 2 22.

45 . . right. Coordinate descent algorithms. Mathematical Programming eries A 151 1 3 34 2 15.

46 C.- . u R. Girshick . e Ch. Feichtenhofer and Ph. rahenbuhl. A multigrid method for e ciently training video models. In Proceedings of the I C F Conference on Computer ision and Pattern Recognition pages 153 162 2 2 .

47 .- . . u. Frequency principle Fourier analysis sheds light on deep neural networks. Communications in Computational Physics 2 5 1746 1767 2 2 .

4 . hang . Lin . egelka . ra and A. adbabaie. Complexity of nding stationary points of nonconvex nonsmooth functions. In Proceedings of the 37th International Conference on Machine Learning PMLR volume 11 pages 11173 111 2 2 2 .

Appendix

A Proofs of convergence theorems

A.1 Proof of Theorem 3.1

Consider iteration k of ML-BCD and suppose that this iteration occurs in the minimization of subproblem $\mathcal{A}_i$ on the variables given by x_i. From the Lipschitz continuity of the gradient and

15 we obtain that

$$f\ x_{k+1}\ \leq f\ x_k\ -\alpha\nabla_x f\ x_k\ ^T\overline{\nabla^1_{x_i} f\ x_k}\quad \frac{\alpha^2 L}{2}\|\overline{\nabla^1_{x_i} f\ x_k}\|^2$$
$$f\ x_k\ -\alpha\|\nabla^1_{x_i} f\ x_k\ \|^2\quad \frac{\alpha^2 L}{2}\|\nabla^1_{x_i} f\ x_k\ \|^2 < f\ x_k\ -\frac{\alpha}{2}\|\nabla^1_{x_i} f\ x_k\ \|^2.$$

ince the minimization of subproblem $\mathcal{A}_i$ has not yet terminated we must have that $\|\nabla^1_{x_i} f\ x_k\ \| \geq \frac{\epsilon}{\sqrt{2}}$ and thus

$$f\ x_{k+1}\ < f\ x_k\ -\frac{\alpha}{2}\epsilon^2.$$

umming this inequality on all iterations we deduce that for all k before termination of the ML-BCD algorithm

$$f\ x_0\ -f_{\mathsf{low}} \geq f\ x_0\ -f\ x_{k+1}\quad \sum_{j=0}^{k}\Big(f\ x_j\ -f\ x_{j+1}\ \Big) \geq \frac{k\quad \mathbf{1}\ \alpha}{2}\epsilon^2.$$

This implies that the algorithm cannot take more than

$$\frac{\mathbf{2}\ f\ x_0\ -f_{\mathsf{low}}}{\alpha\epsilon^2}-\mathbf{1}$$

iterations before it terminates proving the theorem with

$$\kappa_*\quad \frac{\mathbf{2}\ f\ x_0\ -f_{\mathsf{low}}}{\alpha} > \mathbf{2}L\ f\ x_0\ -f_{\mathsf{low}}\ .$$

□

A.2 Proof of Theorem 3.2

Consider iteration k of ML-BCD and suppose that this iteration occurs in the minimization of subproblem $\mathcal{A}_i$ on the variables given by x_i. From the Lipschitz continuity using the G update of the selected block 16 we obtain

$$\mathbb{E}_{\xi_k}\ f\ x_{k+1}\ \leq f\ x_k\ -\alpha\nabla^1_x f\ x_k\ ^T\mathbb{E}_{\xi_k}\ \overline{g_i\ x_k,\xi_k}\quad \frac{L\alpha^2}{2}\mathbb{E}_{\xi_k}\ \|\overline{g_i\ x_k,\xi_k}\|^2\ . \qquad \mathbf{27}$$

sing assumption 17 and the bound on α we deduce that

$$\mathbb{E}_{\xi_k}\ f\ x_{k+1}\ -f\ x_k\ \leq -\alpha\mu\|\nabla^1_{x_i} f\ x_k\ \|^2\quad \frac{\alpha^2 L\sigma_1^2}{\mathbf{2}}\quad \frac{\alpha^2 L\sigma_2^2}{\mathbf{2}}\|\nabla^1_{x_i} f\ x_k\ \|^2$$
$$\alpha\left(\frac{\alpha L\sigma_2^2}{\mathbf{2}}-\mu\right)\|\nabla^1_{x_i} f\ x_k\ \|^2\quad \frac{\alpha^2 L\sigma_1^2}{\mathbf{2}}$$
$$\leq -\frac{\alpha\mu}{\mathbf{2}}\|\nabla^1_{x_i} f\ x_k\ \|^2\quad \frac{\alpha^2 L\sigma_1^2}{\mathbf{2}},$$

and if k is a ma or iteration we have that

$$\mathbb{E}_{\xi_k}\ f\ x_{k+1}\ -f\ x_k\ \leq -\frac{\alpha\mu\tau}{\mathbf{2}}\|\nabla^1_x f\ x_k\ \|^2\quad \frac{\alpha^2 L\sigma_1^2}{\mathbf{2}}. \qquad \mathbf{2}$$

Taking the total expectation gives that

$$\mathbb{E}\ f\ x_{k+1}\ -f\ x_k\ \leq -\frac{\alpha\mu}{\mathbf{2}}\mathbb{E}\ \|\nabla^1_{x_i} f\ x_k\ \|^2\quad \frac{\alpha^2 L\sigma_1^2}{\mathbf{2}}. \qquad \mathbf{2}$$

Consider now a ma or iteration k denoting by $x_{k+\ell}$ for $\ell \in \{1, \ldots, \ell_k\}$ and $\ell_k \leq \ell$ the iterations performed on the selected subproblem. e deduce from 2 that

$$\mathbb{E}\, f\ x_{k+\ell_k}\ - f\ x_k \quad \sum_{\ell=1}^{\ell_k-1} \mathbb{E}\, f\ x_{k+\ell+1}\ - f\ x_{k+\ell} \quad \mathbb{E}\, f\ x_{k+1}\ - f\ x_k$$
$$\leq \sum_{\ell=1}^{\ell_k-1} \left(-\frac{\alpha\mu}{2} \mathbb{E}\, \|\nabla^1_{x_i} f\ x_{k+\ell}\ \|^2 \quad \frac{\alpha^2 L \sigma_1^2}{2} \right) - \frac{\alpha\mu}{2} \mathbb{E}\, \|\nabla^1_{x_i} f\ x_k\ \|^2 \quad \frac{\alpha^2 L \sigma_1^2}{2}.$$

If the minimization of the subproblem is not stopped this means that $\|g_i\ x_{k+\ell}, \xi_{k+\ell}\ \| \geq c_\epsilon$ so that $\mathbb{E}_{\xi_{k+\ell}}\ \|g_i\ x_{k+\ell}, \xi_{k+\ell}\ \|^2\ \geq c_\epsilon^2$. From 17 we have that

$$\|\nabla^1_{x_i} f\ x_{k+\ell}\ \|^2 \geq \frac{\mathbb{E}_{\xi_{k+\ell}}\ \|g_i\ x_{k+\ell}, \xi_{k+\ell}\ \|^2\ - \sigma_1^2}{\sigma_2^2} \geq \frac{c_\epsilon^2 - \sigma_1^2}{\sigma_2^2}$$

and since k is a ma or iteration from the condition for the choice of the blocks we have $\|\nabla^1_{x_i} f\ x_k\ \| > \tau \|\nabla^1_x f\ x_k\ \|$. Thus

$$\mathbb{E}\, f\ x_{k+\ell_k}\ - f\ x_k\ \leq -\frac{\alpha\mu\ell}{2} \frac{c_\epsilon^2 - \sigma_1^2}{\sigma_2^2} - \frac{\alpha\mu\tau}{2} \mathbb{E}\, \|\nabla^1_x f\ x_k\ \|^2 \quad \frac{\alpha^2 L \sigma_1^2\ \ell \quad 1}{2}.$$

Taking into account the fact that that f is bounded from below summing up K ma or iterations and using the assumption $c_\epsilon > 2\sigma_1^2$ we obtain

$$f_{\min} - f\ x_1\ \leq \sum_{k=1}^{K} \mathbb{E}\, f\ x_{k+\ell_k}\ - f\ x_k$$
$$\leq -\frac{\alpha\mu\tau}{2} \sum_{k=1}^{K} \mathbb{E}\, \|\nabla^1_x f\ x_k\ \|^2 \quad \frac{K\ \ell \quad 1\ \alpha^2 L \sigma_1^2}{2} - \frac{\alpha\mu\ell K}{2} \frac{\sigma_1^2}{\sigma_2^2}.$$

Thus

$$\mathbb{E}\left(\sum_{k=1}^{K} \frac{\|\nabla^1_x f\ x_k\ \|^2}{K} \right) \leq \frac{2\ f\ x_1\ - f_{\min}}{\alpha\mu K\tau} \quad \frac{\alpha L \sigma_1^2}{\mu\tau} \quad \frac{\ell\sigma_1^2}{\tau} \left(\frac{\alpha L}{\mu} - \frac{1}{\sigma_2^2} \right) \qquad 3$$
$$\leq \frac{2\ f\ x_1\ - f_{\min}}{\alpha\mu K\tau} \quad \frac{\alpha L \sigma_1^2}{\mu\tau} \qquad 31$$

where the last inequality follows from the fact that $\ell\sigma_1^2 \left(\frac{\alpha L}{\mu} - \frac{1}{\sigma_2^2} \right) <$ by the assumption on α. Taking ℓ large we can thus improve the bound as long as ϵ remains larger than the variance. □

B A Mscale FAML variant and its performance

B.1 Another frequency-aware architecture

This appendix reports on tests conducted using a frequency-aware network architecture based on MscaleDNN de ned in 26 instead of the P networks used in ection 5. A large part of the success of Mscale networks is due to their wavelet-inspired activation functions. These compactly-supported have good scale separation properties and are constructed so that the bandwidth of their Fourier transform increases with their input scaling. everal activation functions have been proposed in 26 but the most e cient one from a practical point of view has been introduced in 24 and is given by

$$\text{s2ReL}\quad z \quad \sin 2\pi z\ \text{ReL}\quad - z - 1\ \text{ReL}\quad z\ , \qquad 32$$

where z is the input scaling. This is a continuous function which decays faster and has better localization property in the frequency domain than the previously proposed sRelu. The amplitude peaks of the di erently scaled s2ReL in the frequency domain are well separated in the Fourier domain. They are indicated by black stars in Figure 11 and we observe their expected monotonic growth with scaling.

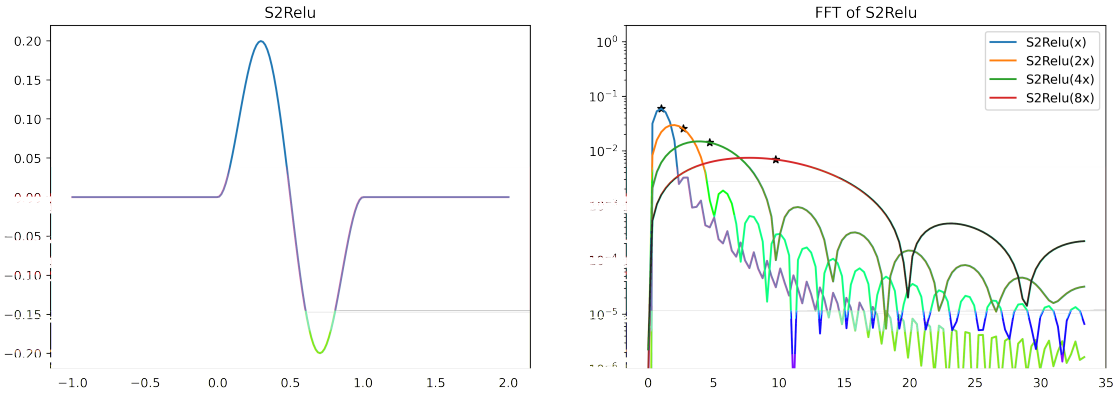

Figure 11 The s2Relu activation function left and its Fourier transform for several scalings right . The black stars highlight the amplitude peaks.

As in ection 5 we use these functions to create a neural network composed of a sum of subnetworks each targeting a di erent frequency range analogously to Figure 4 . In this modi ed architecture the lower frequency networks still use tanh activation functions while the higher frequency networks now use 32 . For each subnetwork the rst activation function is a SFM function associated with a xed input scaling as in standard Mscales networks. The details of this architecture are given in Table 4. ach of the subnetworks is composed of 3 hidden layers of 1 neurons each.

	input scaling	First Activation	ther Activations
subnetwork 1	.5 1 1.5 ... 1	FM 1.	tanh
subnetwork 2	11 12 ... 2 3	FM .5	s2ReL
subnetwork 3	31 32 ... 5 51	FM .5	s2ReL
subnetwork 4	51 52 ... 6 7	FM .5	s2ReL

Table 4 Details of the parallel Mscale architecture

Associated with the ML-BCD algorithm exactly as in ection 5 this modi ed architecture de nes an Mscale variant of the FAML approach.

B.2 Numerical results

e tested this approach using the same experimental setup and methodology as that of ection 5 and again compared its performance to that the standard single level training applied on the complete network. The results are reported in Table 5.

Problem	Budget	M	M FAML	Loss	Loss FAML
Test Pb 1	1	1.33 - 5	2.76 - 6	2.4 - 1	7. - 3
	1	1. 5 - 6	2.51 - 6	4.53 - 3	2.5 - 3
Test Pb 2	1	1.22 - 5	3. 7 - 7	1.73 - 1	.14 - 4
	1	6.11 - 7	2.74 -	1.22 - 5	3. 7 - 7
Test Pb 3	1	1. 6 - 5	1. 5 - 7	2. - 1	1. 6 - 3
	1	4. - 7	1.23 -	6.17 - 4	2. - 4
Test Pb 4	1	1.56 - 5	1. - 7	2. 4 - 1	1. 4 - 3
	1	5.24 - 7	1.21 -	6.26 - 4	1. 5 - 4

Table 5 Best value of the median loss and associated M for standard single-level and Mscale-FAML training 1 independent runs

As was the case when using P- P networks the Mscale FAML variant produced lower losses than the standard training in each case. The di erences are particularly signi cant for a small computational budget where the improvement is of several orders of magnitude. This is also almost always the case for the associated M . These results thus remain excellent despite the fact that our complexity theory does not formally cover the Mscale FAML variant because of the non-smoothness of 32 .