

Linear Attention with Global Context: A Multipole Attention Mechanism for Vision and Physics

Alex Colagrande¹, Paul Caillon¹, Eva Feillet¹, Alexandre Allauzen^{1,2}

¹ Miles Team, LMSADE, Université Paris Dauphine-PSL, Paris, France

² ESPCI PSL, Paris, France

{name}. {surname}@dauphine.psl.eu

Abstract

*Transformers have become the de facto standard for a wide range of tasks, from image classification to physics simulations. Despite their impressive performance, the quadratic complexity of standard Transformers in both memory and time with respect to the input length makes them impractical for processing high-resolution inputs. Therefore, several variants have been proposed, the most successful relying on patchification, downsampling, or coarsening techniques — often at the cost of losing the finest-scale details. In this work, we take a different approach. Inspired by state-of-the-art techniques in n-body numerical simulations, we cast attention as an interaction problem between grid points. We introduce the **Multipole Attention Neural Operator (MANO)** that computes attention in a distance-based multiscale fashion. MANO maintains, in each attention head, a **global receptive field** and has a **linear time and memory complexity** with respect to the number of grid points. Empirical results on image classification and Darcy flows demonstrate that MANO rivals state-of-the-art models, such as ViT and Swin transformer, while reducing runtime and peak memory usage by orders of magnitude. We open-source our code for reproducibility at: <https://github.com/AlexColagrande/MANO>.*

1. Introduction

Convolutional Neural Networks (CNNs) have formed the cornerstone of modern computer vision [22, 29, 31]. Their architectural design leverages the spatial locality and translational invariance properties of images by applying shared convolutional filters over local receptive fields, enabling an efficient parameter usage and a strong inductive bias for grid-structured data.

In recent years, Vision Transformers (ViTs) [16] have emerged as an alternative to CNNs. They are based on the Transformer architecture [54] introduced in the field of Nat-

ural Language Processing (NLP) for sequence-to-sequence learning. This neural architecture is characterized by the use of the self-attention mechanism [2] that allows modeling global contextual information across the *tokens* of a text or the *patches* of an image. Despite lacking the strong locality priors of CNNs, attention-based architectures have demonstrated competitive performance in image classification, particularly when trained on large-scale datasets [44].

Beyond computer vision and NLP, Transformer-based models have found application in scientific machine learning, particularly in the resolution of Partial Differential Equations (PDEs). PDEs constitute the fundamental mathematical framework for modeling a vast array of phenomena across the physical and life sciences - from molecular dynamics to fluid flows and climate evolution. Substantial efforts have been devoted to approximating the solution operators of such equations at scale. Classical numerical solvers — including finite difference [14], finite element [13], and spectral methods [6]— discretize the underlying continuous operators, thereby recasting the problem as a finite-dimensional approximation. More recently, the increasing availability of observational data on structured grids has fostered a paradigm shift towards data-driven approaches such as Physics-Informed Neural Networks (PINNs) [24, 39, 46]. PINNs harness these observations to learn PDE solutions directly, enforcing physical consistency through soft constraints without relying on explicit mesh-based formulations. However, like classical numerical solvers, PINNs are typically designed to approximate the solution of a specific PDE instance — for example, computing the solution corresponding to a fixed coefficient, boundary or initial condition. This means even minor variations in input parameters require re-solving the system or, in the case of neural models, costly re-training. In contrast, operator learning [25] targets a fundamentally more ambitious goal: to approximate a mapping between infinite-dimensional function spaces.

Although considerably more challenging, operator learning offers the advantage to generalize across input condi-

tions without further optimization, offering a scalable and computationally efficient alternative to traditional point-wise solvers. *Note that operator learning is not restricted to PDEs, as images can naturally be viewed as real-valued functions on 2-dimensional domains.*

As in computer vision, recent neural operator models benefit from the development of attention-based architectures [1, 3]. However, attention suffers from quadratic time and memory complexity with respect to the input size, making it impractical for high-resolution data. To tackle this, some variants of the attention mechanism process data in local patches or down-sample the input, drastically cutting computational cost but often sacrificing the fine-grained details crucial for dense-prediction tasks. Other methods replace full attention with low-rank approximations [55], sparsity-inducing schemes [59], or kernel-inspired formulations [12]. Alternatively, Synthesizer proposes to learn attention weights without relying on explicit query–key products [49]. However, these approaches often trade off run-time for expressiveness or ease of implementation.

In this work, we propose an efficient variant of the attention mechanism specifically suited for image classification as well as dense-prediction tasks such as physical simulations. Our method achieves computational gains by relaxing the classical attention formulation while preserving performance by preserving global context.

We propose the **Multipole Attention Neural Operator** (MANO), a novel transformer neural operator in which each head computes attention between a point and a multiscale expansion of the input centered at that point. The attention is performed against a hierarchical decomposition of the input, dynamically downsampled based on the query location. Importantly, we compute the query, key and value matrices Q , K and V at every scale using the same point-wise operator to allow the model to accept inputs at any resolution.

Our contributions are as follows:

- We propose the Multipole Attention Neural Operator (MANO) that formulates attention as an interaction problem and solves it using the Fast Multipole Method.
- By combining MANO with the Swinv2 architecture, we improve transfer learning results on several image classification tasks.
- MANO achieves state-of-the art results on Darcy flow simulation benchmarks matching, and sometimes surpassing, state of the art baselines.

2. Related work

The Vision Transformer (ViT) [16] was the first to successfully adapt the Transformer architecture to image classification, achieving remarkable performance. It divides the input image into fixed-size patches, flattens them into token embeddings, adds positional encodings, and processes the resulting sequence with a Transformer encoder. When

pretrained on large datasets such as ImageNet-21k [15] or LVD-142M[44], ViTs rival or exceed CNNs on image classification tasks. However, despite their efficiency, they suffer from limited local information interaction, single-feature representation and therefore low-resolution outputs making it sub-optimal for dense prediction tasks.

These limitations have motivated a number of efficient vision transformer variants.

2.1. Efficient Vision Transformer Variants

Swin Transformers The Swin Transformer [35] restricts self-attention to non-overlapping windows that are shifted between layers, yielding hierarchical, multi-scale representations without global attention. Swin Transformer V2 [36] augments this design with learnable-temperature scaled cosine attention, log-spaced relative position bias, and continuous pre-norm, improving high-resolution stability and enabling deeper networks—all while preserving the original’s efficient window-based computation.

Distilled and Compact ViTs : TinyViT [58] uses *pretraining-stage distillation* from a large teacher (e.g., Swin-B/L trained on ImageNet-21k). By caching teacher logits and applying neural architecture search under FLOPs/parameter constraints, TinyViT produces smaller models at only a small performance loss.

Data-Efficient Image Transformers (DeiT) [51] add a learnable *distillation token* that learns from a CNN teacher’s soft logits. Later work [52] adds self-supervised distillation and token pruning for further efficiency.

Collectively, these efforts have greatly extended ViT applicability across resource-constrained tasks. However, the inherent multi-scale structure of images remains only partially integrated into existing alternatives to the attention mechanisms, potentially hindering the overall performance.

2.2. Operator Learning via Multipole Attention

In this work, we illustrate the interest of our proposed multipole attention mechanism for learning solution operators of PDEs directly from input–output pairs, as encountered in tasks like fluid flow estimation and other dense prediction problems [25]. Operator learning was first explored by Lu et al. [37], who established a universal approximation theorem for nonlinear operators using DeepONets, laying theoretical foundations for neural operator approximation. Building on this foundation, the Fourier Neural Operator (FNO) [33] parameterizes an integral kernel in the Fourier domain—using efficient FFT-based convolutions to capture global interactions across the entire domain. These pioneering methods have since inspired a wealth of extensions—but their reliance on global or Fourier-based interactions limits their scalability to very high-resolution grids. These works paved the way for numerous extensions.

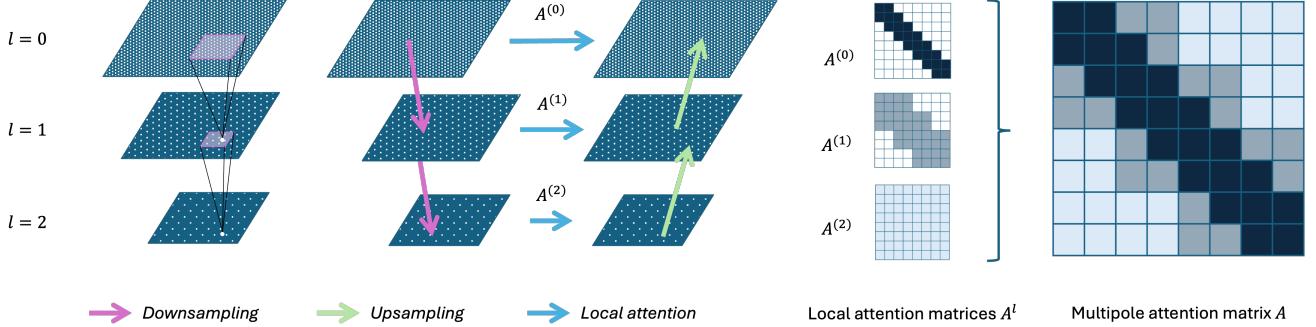


Figure 1. (Left) The multi-scale grid structure. (Center) The V-cycle structure for computing multipole attention with the fast multipole method. (Right) Attention matrices. Illustration with three levels. The attention matrix A is computed in a multiscale manner with respect to each level. The higher the level, the shorter the range of the interaction. At a given layer, down-sampling (resp. up-sampling) is performed using a convolution kernel (resp. deconvolution) shared across all different levels.

Transformer neural operators In [10] the classical transformer was adapted for the first time to operator learning problems related to PDEs. The paper explores two variants, respectively based on the Fourier transform and on the Galerkin method. The latter one uses a simplified attention based operator, without softmax normalization. This solution shares the linear complexity with our work but not the same expressivity. In this line of work, LOCA [34] uses kernel theory to map the input functions to a finite set of features and attends to them by output query location. Recently, [9] proposed to handle attention in a continuous setting and, as well as [56], proposed an operator-learning version of the more classical ViT. Notably, the Universal Physics Transformer (UPT) [1] scales efficiently based on a coarsening of the input mesh.

Multiscale numerical solvers. Our method is inspired by multi-scale numerical solvers [7, 8, 21], in particular the Fast Multipole Method (FMM). A new version of the Fast Multipole Method is introduced by [19] for the evaluation of potential fields in three dimensions and its specialization with the V-cycle algorithm, introduced by [43].

Theoretical studies on transformers. The particle-interaction interpretations of attention was first introduced in Sinkformer [48]. [17] views Transformers as interacting particle systems, they described the geometry of learned representations when the weights are not time dependent, and [18] developed a mathematical framework for analyzing Transformers based on their interpretation as interacting particle systems inspiring us to compute the attention using the most efficient techniques available for solving particle-interaction problems.

Multiscale neural architectures. Several transformer architectures related to the multiscale principle used in our

method were proposed in the one-dimensional setting of Natural Language Processing (NLP) [23, 41, 60, 62], and in graph learning methods [32, 40, 61]

Relation to Fast Multipole Attention Among existing approaches, the closest to ours is Fast Multipole Attention (FMA) [23], which reduces the $O(N^2)$ cost of 1D self-attention via hierarchical grouping: nearby queries attend at full resolution, while distant keys are merged into low-rank summaries, achieving $O(N \log N)$ or $O(N)$.

Our method differs in two key aspects:

- **Input domain.** FMA targets one-dimensional token sequences; we operate on two-dimensional image grids with multiscale spatial windows.
- **Downsampling.** FMA hierarchically downsamples queries, keys, and values. In contrast, we downsample the input feature map *prior* to attention, yielding a self-contained block that integrates seamlessly with standard transformer backbones (e.g., SwinV2) and preserves pre-trained attention weights.

3. Introducing MANO

3.1. Attention as an interaction problem

In this section, we cast the computation of self-attention as a dense n -body interaction problem. An n -body system consists of n entities (often referred to as bodies) whose state is described by a configuration (x_1, \dots, x_n) . The evolution of such a system is governed by a set of interaction laws, which in our setting are determined by pairwise interactions specified through a kernel function:

$$\begin{aligned} \kappa : \mathbb{R}^d \times \mathbb{R}^d &\rightarrow \mathbb{R} \\ (x_i, x_j) &\mapsto \kappa(x_i, x_j) \end{aligned}$$

An n -body simulation refers to a numerical method for computing these interactions, typically requiring $O(n^2)$ op-

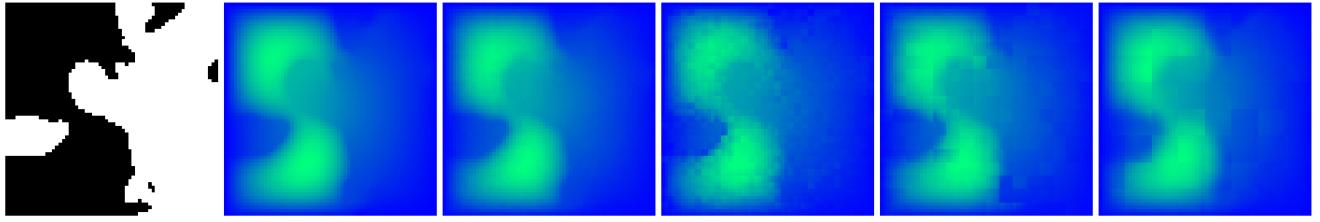


Figure 2. Darcy flow reconstruction: from left to right—input coefficient field, ground truth solution, MANO prediction, and ViT predictions using patch sizes 2, 4, and 8. MANO applies multipole attention using overlapping windows of size 2, and performs downsampling and upsampling across 5 levels using convolutions with kernel size 2×2 , stride 2, and zero padding.

erations due to the dense pairwise structure. This computational cost motivates the development of faster approximations, such as the Fast Multipole Method, which reduces the complexity to $O(n)$.

In the following, we consider a regular grid, corresponding either to the pixels of an image in the classification setting or to the discretized samples of an input function in the operator learning framework. We denote by X a sequence of N observations $(x_1, \dots, x_N)^\top \in \mathbb{R}^{N \times d}$ with elements embedded in dimension d . The self-attention mechanism firstly applies three learnable linear projections to obtain queries, keys and values [54]:

$$Q = XW_q, \quad K = XW_k, \quad V = XW_v \quad (1)$$

with $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$ and $b_q, b_k, b_v \in \mathbb{R}^d$. Next, it computes an $N \times N$ attention matrix A whose i -th row forms a probability distribution over all keys:

$$A_{ij} = \frac{\exp(Q_i^\top K_j / \sqrt{d})}{\sum_{l=1}^N \exp(Q_i^\top K_l / \sqrt{d})}. \quad (2)$$

Finally, each token is updated as a convex combination of the value vectors: $x_i \leftarrow \sum_{j=1}^N A_{i,j} V_j$. In this form, one can view the set $\{x_i\}_{i=1}^N$ as a cloud of N particles in \mathbb{R}^d interacting in a pairwise manner via a kernel κ defined as:

$$\kappa(Q_i, K_j) = \exp(Q_i^\top K_j / \sqrt{d}). \quad (3)$$

Therefore, we interpret a self-attention layer as a single time step of a discretized N -body dynamical system. Under this analogy, computing attention is equivalent to predicting the next state of an interacting particle system — and it becomes natural to accelerate this computation using the FMM [19], reducing the usual $O(N^2)$ cost of the pairwise sums to $O(N)$.

3.2. MANO

In this section we detail the Multipole Attention layer as well as the complexity of the model.

Method Overview : Let $X_0 = X \in \mathbb{R}^{H \times W \times d}$ be the original high-resolution image (height H , width W , embedding dimension d). We define L levels of downsampling by a convolutional kernel D with weights shared across levels, producing $X_\ell = D(X_{\ell-1})$ ($\ell = 1, \dots, L$) where $X_\ell \in \mathbb{R}^{H/2^\ell \times W/2^\ell \times d}$. At each level ℓ , we partition the feature map into potentially overlapping sliding windows and, within each window, compute the attention map $A_\ell = \text{Softmax}(Q_\ell K_\ell^\top / \sqrt{d})$ where $Q_\ell, K_\ell, V_\ell \in \mathbb{R}^{(H/2^\ell \times W/2^\ell) \times d}$ are the query, key, and value embeddings extracted from X_ℓ . This restricts self-attention to localized neighborhoods while still enabling cross-window interactions via the sliding overlap and the hierarchical mixing.

We then produce the attended features $\tilde{X}_\ell = A_\ell V_\ell$, and upsample back to the next-finer resolution via a transposed convolution U :

$$\hat{X}_\ell = U(\tilde{X}_\ell) \in \mathbb{R}^{H/2^{\ell-1} \times W/2^{\ell-1} \times d}. \quad (4)$$

Finally, we combine all levels by summation at the original resolution: $X_{\text{out}} = \sum_{\ell=0}^L U^\ell(\text{Attn}(X_\ell))$, where $\text{Attn}(X_\ell)$ denotes \tilde{X}_ℓ at level ℓ , and U^0 is the identity.

Sharing the same convolutional kernel for both down-sampling and up-sampling—and reusing the same attention weights—keeps the total parameter count constant, regardless of the number of layers L . The convolutions have the role to provide a representation of the input at the next scale, independently of the scale. This ensures that an attention map learned at the finest scale produce effective representations at different scale, even in the case of a pretrained attention from a windowed-attention based model, such as the SwinV2, with finetuning on the convolutional parameters but not on the attention weights. The shared convolutions act as scale-agnostic projectors, producing the next-scale feature representation in the same way at every level. As a result, an attention map learned at the finest resolution remains effective across all scales. In practice, this means one can take a windowed-attention backbone, such as SwinV2, freeze its attention weights, and fine-tune only the convolutional parameters to adapt it to new resolutions without increasing model size.

Computational Complexity. We analyze the cost under non-overlapping windows on a square image of side H (so $N = H^2$ tokens), embedding dimension d , window size w , and down-sampling factor k . The maximum number of levels is $L = \log_k(H)$. We denote by $N_\ell = \frac{N}{k^{2\ell}}$ the number of grid points at level ℓ .

Each windowed self-attention on $M = w^2$ tokens costs $O(M^2d)$ and is applied across $O(N_\ell/M)$ windows, for a total complexity of

$$O(N_\ell Md) \quad (5)$$

Windowed attention computes a standard self-attention for a complexity of $O(M^2d)$ within each of the N/M windows. The total complexity is thus $O(NMd)$. Our Multipole attention iterates the same windowed attention and aggregates the contribution at each level ℓ therefore the total complexity reads

$$\sum_{\ell=0}^L N_\ell Md = \sum_{\ell=0}^L O\left(\frac{N}{k^{2\ell}} Md\right) = O(NMd) \quad (6)$$

So interestingly, even though we apply windowed attention across multiple scales, the total cost remains dominated by the finest-scale pass, with coarser levels adding a negligible additional overhead. As a result, our approach preserves the linear complexity of single-scale windowed attention while delivering significantly greater expressive power.

4. Experimental settings

This section outlines the experimental setup for both image classification and physics simulations.

For image classification, we evaluate on several fine-grained datasets a Swin Transformer V2 modified with our proposed attention mechanism. Models are initialized with weights pretrained on ImageNet-1k. The encoder is frozen (except for the additional convolutions of MANO) and a linear classifier is learnt on the target classification tasks.

For physics simulations, we train all models on instances of the Darcy flow problem, from scratch and across different resolutions.

4.1. Image classification

Datasets. The ImageNet-1k [15] dataset is used to pre-train all models and we perform *linear probing* on several downstream classification benchmarks, namely CIFAR-100[28], Oxford Flowers-102 [42], Stanford Cars [26], Food101 [4], Tiny-ImageNet-202 [30] and Oxford-IIIT Pet Dataset [45].

Architecture. As the backbone for our Multipole Attention model, we adopt the “Tiny” version of the Swin Transformer V2 [36]. Since the attention block is shared across all levels of the multipole hierarchy, MANO can inherit the

pretrained weights from the original Swin Transformer, requiring only a small number of additional trainable parameters: one convolution and one transposed convolution per attention head, along with the classification head. Convolutions have a kernel size of 2 and a stride of 2. This design ensures that our variant introduces a minimal increase in parameter count relative to the base model. Specifically, for the Tiny version, the total number of parameters increases from 27,73M to 28,47M, corresponding to an additional 740,356 parameters, or 2.67% more than the original model.

Training. We freeze the pretrained encoder weights and train a single fully connected layer for 50 epochs on top of the frozen encoder using AdamW as optimizer with a cosine annealing learning rate schedule. Training the models with the original shifted attention of Swinv2 and the models with our proposed multipole attention only differs by a warm-up phase to learn the upsampling and downsampling convolutional filters introduced by MANO. We report the resulting top-1 accuracy of these experiments in Table 1.

4.2. Darcy flow simulation

Task We evaluate our method on the task of steady-state 2D Darcy Flow simulation, a widely used task in the neural operator literature [25]. The problem is based on the following second-order, linear elliptic PDE:

$$-\nabla \cdot (a(x)\nabla u(x)) = f(x), \quad x \in (0, 1)^2, \quad (7)$$

to solve with homogeneous Dirichlet boundary conditions: $u(x) = 0$, $x \in \partial(0, 1)^2$. In this PDE, the function $a(x)$ represents the spatially varying permeability of a porous medium. The forcing term $f(x)$ is fixed to $f(x) \equiv 1$ across all inputs. The output $u(x)$ is the scalar field representing the pressure within the domain. Although the PDE is linear in u , the map from the input $a(x)$ to the solution $u(x)$ is nonlinear due to the interaction of $a(x)$ with the gradient operator inside the divergence.

The task is to learn this solution operator: given a new input field $a(x)$, the model must predict the corresponding output $u(x)$. In our experiments, $a(x)$ is sampled as a binary field (i.e., values are either 0 or 1), representing a medium composed of two different materials.

Architecture. We use a classical transformer architecture of depth 8 with 4 attention heads per layer. In place of the conventional self-attention, we employ our multipole attention module. Additionally, we apply Layer Normalization at every level to improve training stability and mitigate issues of vanishing or exploding gradients, which can arise due to shared attention across hierarchical levels.

Model	Params.	Complexity	Tiny-IN-202	Cifar-100	Flowers-102	Food-101	StanfordCars-196	OxfordIIITPet
TinyViT [58]	21M	$O(N^2)$	-	75.2%	82.4%	-	<u>61.7%</u>	86.5%
ViT-base [16]	86M	$O(N^2)$	73.07%	<u>80.63%</u>	92.75%	<u>80.31%</u>	41.95%	<u>87.68%</u>
DeiT-small [52]	22M	$O(N^2)$	<u>81.34%</u>	75.15%	66.60%	71.39%	36.38%	87.56%
SwinV2-T [36]	28M	$O(N)$	80.53%	75.47%	56.46%	76.96%	38.36%	87.14%
MANO-tiny	28M	$O(N)$	87.52%	85.08%	<u>89.00%</u>	82.48%	<u>65.68%</u>	88.31%

Table 1. Linear probing accuracies for several image classification datasets. MANO matches or even outperforms the performances of state-of-the-art models. The results for TinyViT are taken from [58], while all other baselines are fine-tuned via linear probing using pretrained backbones¹. For each model, we report the number of parameters and the asymptotic complexity of its attention block with respect to the number of patches N . As a plug-and-play replacement for the attention mechanism, our attention can be applied after patching, similar to Swin, allowing our vision-specific MANO variant to scale linearly with the number of patches.

Training. We train all the considered models for 50 epochs with AdamW optimizer and cosine learning rate scheduler. The initial learning rate is in the order of 10^{-4} . We use the dataset open-sourced in [25], comprised of input–output pairs (a, u) at resolutions $n \times n$ for $n \in \{16, 32, 64\}$. The model is trained to minimize the mean squared error (MSE) on the training set and evaluated on a held-out test set using the relative MSE error, where \hat{u} is the model prediction and u the ground truth solution: $\frac{\|\hat{u} - u\|_2}{\|u\|_2}$.

5. Results

5.1. Image Classification Results

Table 1 presents the top-1 accuracy of three models of similar parameter counts (ViT (21M parameters), SwinV2-T (28M), ViT-base (86M), DeiT-small (22M) and MANO (28M)) across six downstream image classification datasets. The reported results for TinyViT are taken from [11, 58] while the results for SwinV2-T are taken from [36].

First, MANO consistently outperforms TinyViT on all benchmarks where results are available. It also surpasses DeiT-small and SwinV2-T when these models are fine-tuned via linear probing using ImageNet-1k pretrained weights, across an expanded set of datasets. Compared with the bigger ViT-base, our model performs better in all the benchmark except than Flowers-102 that is the dataset with less training data among the one considered. the dataset with the smallest training set among those considered. This suggests that in low-data regimes, models with a higher parameter count may have an advantage due to their increased capacity to memorize or adapt to limited supervision.

The improvement ranges from about 1–2 % on easier tasks like Oxford–IIIT Pet to nearly 5/7 points on more challenging datasets such as CIFAR–100 and Tiny-ImageNet. Even compared to ViT-base, which has more than twice the number of parameters, MANO achieves gains of roughly 3–10 points accross all the benchmarks expect for Flowers-102, demonstrating that multiscale hierarchical attention produces significantly more transferable features without increasing too much model size.

Second, the advantage of MANO becomes especially pronounced on fine-grained classification tasks. On Flowers–102 and Stanford Cars, SwinV2-T achieves only 56.5% and 38.4% accuracy, respectively, while TinyViT recovers to 82.4% and 61.7%. In both cases, MANO further improves performance to 89.0% on Flowers–102 and 65.7% on Cars, indicating that combining local details (e.g., petal shapes or headlight contours) with global context (overall flower appearance or car silhouette) is critical for distinguishing highly similar classes.

Third, on medium-difficulty datasets such as Tiny–ImageNet–202 and CIFAR–100, MANO again holds a clear lead. It outperforms SwinV2-T by approximately 7 points on Tiny–ImageNet–202 and by around 10 points on CIFAR–100. These results suggest that attending to multiple resolutions—capturing both fine textures and broader scene structures—yields better representations than the single-scale windowed attention used in SwinV2-T.

Finally, although MANO and SwinV2-T share the same parameter count (28M), MANO delivers a consistent 5–10 point advantage on mid-level benchmarks and maintains a smaller lead on easier tasks like Oxford–IIIT Pet. TinyViT’s 21M parameters are insufficient to match either 28M model, underscoring that hierarchical multiscale attention makes more effective use of model capacity than either pure global self-attention (ViT) or fixed-window local attention (SwinV2-T).

5.2. Darcy Flow Simulation Results

Model	16×16	32×32	64×64
FNO	0.0195	0.0050	0.0035
ViT patch_size=8	0.0160	<u>0.0038</u>	0.0021
ViT patch_size=4	0.0179	0.0039	<u>0.0019</u>
ViT patch_size=2	0.0169	0.0049	0.0026
Local Attention	<u>0.0133</u>	0.0188	0.0431
MANO	0.0080	0.0020	0.0013

Table 2. Benchmark on Darcy flow simulations. Relative MSE. A given model is evaluated and tested on the same resolution, either 16×16 , 32×32 or 64×64 .

Table 2 presents the relative MSE of various models trained from scratch on Darcy flow at different resolutions. The Fourier Neural Operator (FNO) [33] is a neural operator designed to learn mappings between function spaces, such as the coefficient-to-solution map for PDEs. By operating in the Fourier domain, the FNO captures long-range dependencies across the entire domain with near-linear complexity $O(N^2 \log N)$ (for an $N \times N$ grid), making it effective for a wide range of PDE-based tasks, leading to state-of-the-art results on tasks such as Darcy Flow Simulation. It achieves MSEs of 0.0195, 0.0050, and 0.0035 as the grid is refined, showcasing its strength in capturing global spectral components but its limited ability to resolve fine-scale details at coarser resolutions.

For ViT, we evaluate patch sizes of 8, 4, and 2: the patch-4 variant attains the best errors (0.0179, 0.0039, 0.0019), whereas smaller patches (size 2) slightly worsen performance at low resolution and fail to match patch-4 at 64^2 . This sensitivity indicates that vanilla ViT’s pure global attention is capable of approximating the solution operator but depends heavily on patch granularity. In contrast, a pure local-attention model (fixed window) degrades sharply at 32^2 and 64^2 , since local windows cannot propagate long-range dependencies across the domain.

By combining fine-grid attention (to capture local conductivity channels) with progressively coarser resolutions (to model global pressure fields), MANO consistently achieves the lowest errors, roughly halving the MSE of both FNO and standard ViT at every scale and overcoming the locality limitations inherent in fixed-window attention.

Lastly, we examine in Figure 2 the quality of the reconstructed images, demonstrating that MANO’s multi-scale modeling recovers both sharp transitions and smooth boundaries with high fidelity, even on a coarse grid. By contrast, images reconstructed by ViT exhibit noticeable patching artifacts.

In summary, MANO’s multiscale hierarchical attention achieves state-of-the-art performance on both Darcy flow simulations and image classification tasks. Its design makes it well suited to the corresponding data, as it captures fine-scale detail and broad-scale context simultaneously.

Hyperparameters. A detailed table of hyperparameters is provided in the Appendix; below, we outline our main design choices.

The “number of levels” specifies how many hierarchical scales are included in the multipole attention; with a window size of 8 for the windowed attention, we achieve

¹Checkpoints are available at <https://huggingface.co/google/vit-base-patch16-224> (ViT-base), <https://huggingface.co/facebook/deit-small-patch16-224> (DeiT-Small), and https://huggingface.co/timm/swinv2-tiny_window8_256.ms_in1k (SwinV2-T).

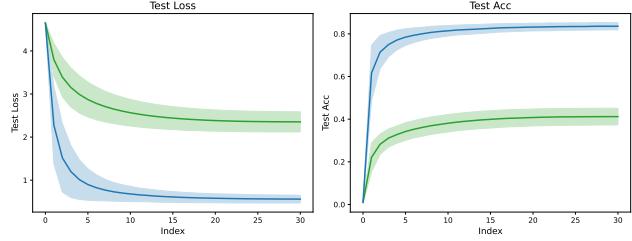


Figure 3. Ablation study comparing average pooling (green) and learnable convolutions (blue) for the sampling step in MANO. We report the Cross-Entropy validation loss (left) and accuracy (right) on CIFAR-100. Mean and standard deviation over fifteen runs are reported, varying the learning rate between 10^{-3} and 10^{-4} .

the best performance using the maximum of 3 levels. Due to Swin’s built-in downsampling between stages, this corresponds to 3 levels across the first two layers, 2 levels for the next two layers, and a single level for the remaining eight layers. To coarsen the input grid, we compared average pooling versus learned convolutions—convolutions consistently outperformed pooling. For upsampling, transpose convolutions outperformed nearest-neighbor interpolation. When used, `kernel_size` and `stride` refer to these convolutional operations.

As shown in Figure 3, using learned convolutions for both down- and up-sampling significantly improves expressivity: even with pretrained attention weights, convolution-based sampling enables a windowed attention trained at one resolution to transfer effectively to another. Note that a single convolutional kernel is reused for all downsampling operations, and a separate kernel is reused for all upsampling operations.

6. Discussion

Hierarchy depth in vision vs. Physics. In our image classification experiments, we follow SwinV2-T’s architecture and compute attention at three hierarchical levels in early stages, two in the middle, one at the end of the encoder). For Darcy flow grids, we set downsampling steps so that the coarsest scale is 2×2 , the smallest possible when using a 2×2 window for the attention. We found that increasing the number of levels consistently improves performance, however we note that in physics simulation, the number of levels can be treated as a hyperparameter and tuned based on input resolution and the desired balance between local and global interactions.

Limitations. MANO’s current design uses a fixed, static hierarchy and attention parametrization. While effective, it could benefit from learnable scale selection and explicit cross-level interactions to better capture multi-scale couplings. Additionally, we assume a uniform grid to discretize

the input. This simplifies implementation but fails to capture regions with steep gradients or intricate boundaries in physics simulations. Introducing adaptive meshing would thus improve accuracy and efficiency for localized phenomena that otherwise demand a higher resolution simulation, but would require redefining attention over nonuniform spatial supports. Finally, our current implementation could be accelerated by a hardware-optimized GPU kernel to further reduce runtime.

Future Directions. Beyond steady-state Darcy flow, MANO can naturally extend to time-dependent PDEs by integrating with recurrent timestepping or operator-splitting schemes, preserving its ability to capture both spatial multiscale structure and temporal evolution. We also plan to extend our method to unstructured meshes and irregular domains common in real-world physics simulations. On the computer-vision side, applying MANO to dense prediction tasks—such as semantic segmentation or image inpainting—is promising, since its multiscale attention could dynamically balance local details and global context more effectively than standard U-Net architectures.

7. Conclusion

We propose **MANO**, an efficient attention-based architecture inspired by n -body methods, which interprets attention as interactions among mesh points. By introducing a distance-based, multiscale attention mechanism, MANO achieves linear time and memory complexity per head while preserving a global receptive field. Across several image classification benchmarks and Darcy flow simulations, MANO matches the accuracy of full-attention models yet substantially reduces runtime and peak memory. Unlike patch-based approximations, it avoids discontinuities and retains long-range dependencies inherent to physical systems. Our results demonstrate that MANO is a scalable alternative for both vision tasks and mesh-based simulations. Future work includes applying MANO to dense vision tasks such as semantic segmentation, and to extend it to irregular meshes and a broader class of physical simulations.

Acknowledgments

This work was supported by the SHARP ANR project ANR-23-PEIA-0008 in the context of the France 2030 program and by the HPC resources from GENCI-IDRIS (grants AD011015154 and A0151014627).

References

- [1] Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. *Advances in Neural Information Processing Systems*, 37:25152–25194, 2024. [2](#), [3](#), [12](#)
- [2] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015. [1](#)
- [3] Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan A. Weyn, Haiyu Dong, Jayesh K. Gupta, Kit Thambiratnam, Alexander T. Archibald, Chun-Chieh Wu, Elizabeth Heider, Max Welling, Richard E. Turner, and Paris Perdikaris. A foundation model for the earth system. *Nature*, 641(8065):1180–1187, 2025. [2](#)
- [4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101-mining discriminative components with random forests. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014. [5](#)
- [5] Lise Le Boudec, Emmanuel de Bezenac, Louis Serrano, Ramon Daniel Regueiro-Espino, Yuan Yin, and Patrick Gallinari. Learning a neural solver for parametric PDEs to enhance physics-informed methods. In *The Thirteenth International Conference on Learning Representations*, 2025. [12](#)
- [6] R.N. Bracewell. *The Fourier Transform and Its Applications*. McGraw Hill, 2000. [1](#)
- [7] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138):333–390, 1977. [3](#)
- [8] William L Briggs, Van Emden Henson, and Steve F McCormick. *A multigrid tutorial*. SIAM, 2000. [3](#)
- [9] Edoardo Calvello, Nikola B Kovachki, Matthew E Levine, and Andrew M Stuart. Continuum attention for neural operators. *arXiv preprint arXiv:2406.06486*, 2024. [3](#), [12](#)
- [10] Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021. [3](#), [12](#)
- [11] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021. [6](#)
- [12] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. [2](#), [11](#)
- [13] P.G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002. [1](#)
- [14] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM J. Res. Dev.*, 11(2):215–234, 1967. [1](#)
- [15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [2](#), [5](#), [11](#), [14](#)
- [16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

- Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 2, 6, 11
- [17] Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics, 2024. 3
- [18] Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. A mathematical perspective on transformers, 2024. 3
- [19] Leslie Greengard and Vladimir Rokhlin. A new version of the fast multipole method for the laplace equation in three dimensions. *Acta Numerica*, 6:229–269, 1997. 3, 4
- [20] Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. *Advances in neural information processing systems*, 34:24048–24062, 2021. 12
- [21] Wolfgang Hackbusch. *Multi-grid methods and applications*. Springer Science & Business Media, 2013. 3
- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1
- [23] Yanming Kang, Giang Tran, and Hans De Sterck. Fast multipole attention: A divide-and-conquer attention mechanism for long sequences, 2024. 3, 12
- [24] George Karniadakis, Yannis Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, pages 1–19, 2021. 1, 12
- [25] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023. 1, 2, 5, 6, 12
- [26] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5
- [27] Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, pages 26548–26560. Curran Associates, Inc., 2021. 12
- [28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. 5
- [29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. 1
- [30] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5
- [31] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 1
- [32] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:6755–6766, 2020. 3, 12
- [33] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations, 2021. 2, 7, 12
- [34] Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36:35836–35854, 2023. 3, 12
- [35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2, 11
- [36] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022. 2, 5, 6, 11
- [37] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019. 2, 12
- [38] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021. 12
- [39] Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM review*, 63(1):208–228, 2021. 1
- [40] Leon Miguës, Yuan Yin, Jocelyn Ahmed Mazari, and Patrick Gallinari. Multi-scale physical representations for approximating pde solutions with graph neural operators. In *Topological, Algebraic and Geometric Learning Workshops 2022*, pages 332–340. PMLR, 2022. 3, 12
- [41] Tan M. Nguyen, Vai Suliafu, Stanley J. Osher, Long Chen, and Bao Wang. Fmmformer: Efficient and flexible transformer via decomposed near-field and far-field attention, 2021. 3, 12
- [42] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 5
- [43] G. Of. An efficient algebraic multigrid preconditioner for a fast multipole boundary element method. *Computing*, 82(2):139–155, 2008. 3
- [44] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al.

- Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 2, 11
- [45] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 5
- [46] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017. 1, 12
- [47] Tim De Ryck, Florent Bonnet, Siddhartha Mishra, and Emmanuel de Bezenac. An operator preconditioning perspective on training in physics-informed machine learning. In *The Twelfth International Conference on Learning Representations*, 2024. 12
- [48] Michael E. Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention, 2022. 3
- [49] Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention for transformer models. In *International conference on machine learning*, pages 10183–10192. PMLR, 2021. 2
- [50] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021. 11
- [51] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 32–42, 2021. 2, 11
- [52] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *European conference on computer vision*, pages 516–533. Springer, 2022. 2, 6, 11
- [53] Tapas Tripura and Souvik Chakraborty. Wavelet neural operator: a neural operator for parametric partial differential equations. *arXiv preprint arXiv:2205.02191*, 2022. 12
- [54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 1, 4
- [55] Sinong Wang, Belinda Z Li, Madiyan Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2, 11
- [56] Sifan Wang, Jacob H Seidman, Shyam Sankaran, Hanwen Wang, George J Pappas, and Paris Perdikaris. Cvit: Continuous vision transformer for operator learning. *arXiv preprint arXiv:2405.13998*, 2024. 3, 12
- [57] Wenhui Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 11
- [58] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Tinyvit: Fast pretraining distillation for small vision transformers. In *European conference on computer vision*, pages 68–85. Springer, 2022. 2, 6, 11
- [59] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020. 2
- [60] Zhanpeng Zeng, Sourav Pal, Jeffery Kline, Glenn M Fung, and Vikas Singh. Multi resolution analysis (mra) for approximate self-attention, 2022. 3, 12
- [61] Maksim Zhdanov, Max Welling, and Jan-Willem van de Meent. Erwin: A tree-based hierarchical transformer for large-scale physical systems. *CoRR*, 2025. 3, 12
- [62] Zhenhai Zhu and Radu Soricut. H-transformer-1d: Fast one-dimensional hierarchical attention for sequences, 2021. 3, 11

A. Extended related work section

In this section, we provide a more comprehensive overview of the related literature, expanding upon the works briefly mentioned in the main text.

Vision Transformers (ViTs) (ViTs) [16] divide each input image into fixed-size patches (e.g., 16×16), flatten them into tokens, add positional embeddings, and process the resulting sequence with a Transformer encoder. When pre-trained on large-scale datasets such as ImageNet-21k [15] or LVD-142M [44], ViTs achieve performance on par with or surpassing that of convolutional neural networks (CNNs) on standard image classification benchmarks.

Despite these advances, ViTs face several limitations:

1. quadratic computational complexity $\mathcal{O}(N^2)$ with respect to the number of input patches (N , where typically $N \approx 196$ for a 224×224 image).
2. Absence of built-in locality and translation equivariance, in contrast to CNNs, which makes ViTs more dependent on large training datasets.
3. High computational and memory demands—for instance, ViT-Large/16 contains roughly 300 million parameters and requires thousands of GPU-hours to train [16].

These drawbacks have spurred the development of more efficient ViT variants.

A.1. Efficient Vision Transformer Variants

Several efficient alternatives to the standard attention have been proposed in the literature to address the limitations of Vision Transformers. While they differ in methodology, they have collectively inspired this work.

Linear-Attention Transformers: Linformer [55] projects keys and values into a low-dimensional subspace ($k \ll N$), reducing per-head complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(Nk)$ while retaining competitive accuracy. Performer [12] uses a randomized feature map to approximate softmax(QK^\top) $\approx \Phi(Q)\Phi(K)^\top$, achieving true $\mathcal{O}(N)$ time and memory with bounded error. When applied to ViT backbones, these methods handle larger images with much lower memory cost.

All-MLP Architectures: MLP-Mixer [50] differs from both CNNs and ViTs by alternating *token-mixing* MLPs (mixing across N spatial tokens) and *channel-mixing* MLPs (mixing across C channels). This yields per-layer complexity $\mathcal{O}(NC)$ instead of $\mathcal{O}(N^2)$, and achieves 84% top-1 on ImageNet-1K (with ImageNet-21k pretraining), demonstrating that dense MLPs can approximate spatial interactions effectively.

Pyramid/Hierarchical ViTs: Pyramid Vision Transformer (PVT) [57] builds a multi-scale pyramid by progressively downsampling tokens: early stages operate on high-resolution grids ($\frac{H}{4} \times \frac{W}{4}$), and deeper stages use “patch merging” to halve spatial dimensions at each level. Within each stage, *Spatial-Reduction Attention (SRA)* pools keys/values by a factor r , reducing sequence length from N to N/r^2 and complexity to $\mathcal{O}(N \cdot N/r^2)$. PVT matches CNN backbones in detection and segmentation.

Swin Transformer [35, 36] introduces *window-based MSA* over non-overlapping $M \times M$ patches (e.g., 7×7), reducing complexity to $\mathcal{O}(\frac{N}{M^2} \times M^4)$. Each stage ends with a *patch merging* layer that concatenates 2×2 tokens and projects them, halving resolution and doubling channels. Crucially, Swin alternates “standard” and “shifted” window partitions: shifted windows (offset by $\lfloor M/2 \rfloor$) overlap adjacent regions, enabling cross-window context without global attention. Swin-B attains 87.3% top-1 on ImageNet-1K, with near-linear inference latency.

Distilled and Compact ViTs: TinyViT [58] uses *pretraining-stage distillation* from a large teacher (e.g., Swin-B/L trained on ImageNet-21k). By caching teacher logits and applying neural architecture search under FLOPs/parameter constraints, TinyViT produces 11M–21M parameter models that achieve 84.8–86.5% top-1 on ImageNet-1K—close to much larger ViTs.

Data-Efficient Image Transformers (DeiT) [51] add a learnable *distillation token* that learns from a CNN teacher’s soft logits (e.g., ResNet-50) while training on ImageNet-1K alone. Combined with aggressive augmentation (RandAugment, Mixup, CutMix) and regularization (Label Smoothing, Stochastic Depth), DeiT-Small (22M) reaches 83.1% top-1 (vs. 77.9% for vanilla ViT), and DeiT-Base (86M) hits 85.2% in under three GPU-days, matching ResNet-152. Later work [52] adds self-supervised distillation and token pruning for further efficiency.

Collectively, these efforts—linear-attention, MLP-only designs, hierarchical token pyramids, window-based local attention, and distillation—have greatly extended ViT applicability across resource-constrained tasks. However, the inherent hierarchical structure of images remains only partially integrated into existing attention mechanisms, potentially hindering the overall performance.

Multiscale neural architectures. Several transformer architectures have been proposed in the one-dimensional setting of Natural Language Processing (NLP) that are closely related to the multiscale principles underlying our method.

H-Transformer-1D [62] introduces a hierarchical attention scheme that restricts full attention to local windows

while allowing global information to flow through a tree-like structure.

MRA-Attention [60] leverages a multiresolution decomposition of attention weights using wavelet transforms to capture both coarse and fine-scale dependencies.

FMMformer [41] builds on the Fast Multipole Method (FMM) to hierarchically group tokens and reduce attention complexity by summarizing distant interactions.

Fast Multipole Attention (FMA) [23] similarly applies FMM-inspired grouping but in a more generalizable attention framework.

ERWIN [61] proposes a multilevel window-based transformer with recursive interpolation between coarse and fine spatial scales in the setting of graph attention.

A.2. Neural Operators

The challenge in solving PDEs is the computational burden of conventional numerical methods. To improve the tractability, a recent line of research investigates how machine learning and especially artificial neural networks can provide efficient surrogate models. A first kind of approach assumes the knowledge of the underlying PDE, like PINNs [24, 38, 46]. With this knowledge, the neural network is optimized by solving the PDE, which can be considered as a kind of unsupervised learning. However, the difficult optimization process requires tailored training schemes with many iterations [27, 47]. In a "semi-supervised" way, the recent approach of Boudec et al. [5] recasts the problem as a *learning to learn* task, leveraging either, the PDE and simulations or observations data. While this method obtained promising results, its memory footprint may limit its large scale usage. In this work, we focus neural operators, which learn directly the solution operator from data [33, 37]. In this line of work, the challenge lies in the model architecture rather than in the optimization process and different kind of models were recently proposed.

Transformer neural operators In [10] the classical transformer was adapted for the first time to operator learning problems related to PDEs. The paper explores two variants, based on Fourier transform and Galerkin method. The latter one uses a simplified attention based operator, without softmax normalization. This solution shares the linear complexity with our work but not the same expressivity. Still in the simplifying trend, LOCA (Learning Operators with Coupled Attention) [34] maps the input functions to a finite set of features and attends to them by output query location.

Based on kernel theory, Li et al. [34] introduces an efficient transformer for the operator learning setting was proposed based on kernel theory. Recently in [9] was proposed an interesting way to see attention in the continuous setting and in particular the continuum patched attention. In Uni-

versal Physics Transformer [1] framework for efficient scaling was proposed based on a coarsening of the input mesh. In [56] the Continuous vision transformer was proposed as a operator-learning version of the more classical ViT.

In the context of operator learning and graph-structured data, the **Multipole Graph Neural Operator (MGNO)** [32] extends multipole ideas to irregular domains via message-passing on graph hierarchies. Finally, **V-MGNO**, **F-MGNO**, and **W-MGNO** [40] propose variations of MGNO to improve stability.

These works highlight the growing interest in multiscale and hierarchical schemes to improve efficiency and generalization, both in sequence modeling and operator learning. Our work builds on this line by proposing a spatially structured multipole attention mechanism adapted to vision and physical simulation tasks.

Our model is explicitly designed to function as a neural operator [25]. To qualify as a neural operator, a model must satisfy the following key properties. First, it should be capable of handling inputs and outputs across arbitrary spatial resolutions. Second, it should exhibit discretization convergence — that is, as the discretization of the input becomes finer, the model's predictions should converge to the true underlying operator governing the physical system. This pose a new challenge to the computer vision community, namely not just learn an image to image function but the underlying operator independently of the resolution. This field saw its first proof of concept with Lu et al. [37], who leveraged a universal approximation theorem for nonlinear operators and paved the way for numerous extensions. Fourier Neural operators [33] rely on a translation-equivariant kernel and discretize the problem via a global convolution performed computed by a discrete Fourier transform. Building on this foundation, the Wavelet Neural Operator (WNO) [53] introduces wavelet-based multiscale localization, enabling kernels that simultaneously capture global structures and fine-grained details. The Multiwavelet Neural Operator (MWNO) [20] further extends this approach by incorporating multiple resolution components, leading to improved convergence with respect to discretization.

B. Detailed hyperparameters

B.1. Architecture Hyperparameters for Image classification

Table 3 summarizes the architectural and training hyperparameters used in our model. Below, we provide brief comments on each of them. The first block in Table 3 corresponds to the standard configuration of the pretrained SwinV2-Tiny model, which we adopt as our backbone.

- **Patch size:** Size of non-overlapping image patches. A value of 4 corresponds to 4×4 patches.
- **Input channels:** Number of input channels, set to 3 for

RGB images.

- **Embedding dimension (`embed_dim`):** Dimensionality of the token embeddings, controlling model capacity.
- **Global pooling:** Global average pooling is used instead of a [CLS] token at the output.
- **Depths (layers per stage):** Number of transformer blocks in each of the four hierarchical stages, e.g., [2, 2, 6, 2].
- **Number of heads (per stage):** Number of attention heads per stage; increases with depth to maintain representation power.
- **Window size:** Local attention is applied in windows of size 8×8 .
- **MLP ratio:** Ratio between the hidden dimension in the feed-forward MLP and the embedding dimension (e.g., $4.0 \times 96 = 384$).
- **QKV bias:** Whether learnable biases are used in the query/key/value projections (set to True).
- **Dropout rates (`drop_rate`, `proj_drop_rate`, `attn_drop_rate`):** All standard dropout components are disabled (set to 0).
- **Drop-path rate (`drop_path_rate`):** Stochastic depth with rate 0.2 applied to residual connections for regularization.
- **Activation layer:** GELU is used as the non-linearity in MLP layers.
- **Normalization layer:** Layer normalization is applied throughout the network.
- **Pretrained window sizes:** Set to [0, 0, 0, 0] as no pre-trained relative position biases are used.
- **Attention sampling rate:** The input to the attention mechanism is downsampled by a factor of 2, allowing for increased expressivity without a relevant additional computational cost.
- **Attention down-sampling:** A convolutional layer with kernel size 2 and stride 2 is used to downsample features between the levels of the multipole attention.
- **Attention up-sampling:** Transposed convolution (kernel size 2, stride 2) is used to upsample the features after the windowed attention at each hierarchical level.
- **Number of levels:** Specifies the number of multipole attention levels used at each stage. We found it beneficial to use the maximum number of levels permitted by the spatial resolution.

B.2. Architecture Hyperparameters for Darcy Flow

Table 4 reports the main architectural hyperparameters used in our MANO model for solving the Darcy flow problem. Below, we provide a brief description of each.

- **channels:** Number of input channels; set to 3 because we concatenate the two spatial coordinate with the permeability coefficient.
- **patch size:** Patch size used to partition the input grid;

Hyperparameter	Value
Patch size	4
Input channels	3
Embedding dimension (<code>embed_dim</code>)	96
Global pooling	avg
Depths (layers per stage)	[2, 2, 6, 2]
Number of heads (per stage)	[3, 6, 12, 24]
Window size	8
MLP ratio	4.0
qkv bias (boolean)	True
Dropout rate (<code>drop_rate</code>)	0.0
Projection-drop rate (<code>proj_drop_rate</code>)	0.0
Attention-drop rate (<code>attn_drop_rate</code>)	0.0
Drop-path rate (<code>drop_path_rate</code>)	0.2
Activation layer	gelu
Normalization layer (<code>flag</code>)	True
Pretrained window sizes	[0, 0, 0, 0]
Attention sampling rate	2
Attention down-sampling	conv
kernel size	2
stride	2
Attention up-sampling	conv transpose
kernel size	2
stride	2
number of levels	[3, 2, 1, 1]

Table 3. MANO Hyperparameters for image classification

set to 1 to retain full spatial resolution, ideal for dense prediction tasks.

- **domain dim:** Dimensionality of the input domain; set to 2 for 2D PDEs like Darcy flow.
- **stack regular grid:** Indicates whether the input discretization is regular and should be stacked; set to `true`.
- **dim:** Embedding dimension of the token representations.
- **dim head:** Dimensionality of each individual attention head.
- **mlp dim:** Hidden dimension of the MLP layers following attention.
- **depth:** Total number of transformer blocks.
- **heads:** Number of self-attention heads in each attention block.
- **emb dropout:** Dropout rate applied to the input embeddings.
- **Attention sampling rate:** The input to the attention mechanism is downsampled by a factor of 2, allowing for increased expressivity without a relevant additional computational cost.
- **Attention down-sampling:** A convolutional layer with kernel size 2 and stride 1 is used to downsample features between the levels of the multipole attention.
- **Attention up-sampling:** Transposed convolution (kernel size 2, stride 1) is used to upsample the features after the windowed attention at each hierarchical level.
- **att dropout:** Dropout rate applied within the attention block.
- **Window size:** Local attention is applied in windows of size 2×2 .

- **local attention stride**: Stride with which local windows are applied; controls overlap in attention.
- **positional encoding**: Whether explicit positional encodings are added; set to `false` in our setting.
- **learnable pe**: Whether the positional encoding is learnable; also disabled here.
- **pos enc coeff**: Scaling coefficient for positional encodings, if used; `null` since not applicable.

Hyperparameter	Value
channels	3
patch size	1
domain dim	2
stack regular grid	true
dim	128
dim head	32
mlp dim	128
depth	8
heads	4
emb dropout	0.1
Attention sampling rate	2
Attention down-sampling	conv
kernel size	2
stride	1
Attention up-sampling	conv transpose
kernel size	2
stride	1
att dropout	0.1
window size	2
local attention stride	1
positional encoding	false
learnable pe	false
pos enc coeff	null

Table 4. MANO Hyperparameters for Darcy flow

C. Implementation details

All our experiments are implemented in PyTorch.

C.1. Model checkpoints

Our experiments in image classification use the following pre-trained models from HuggingFace on ImageNet[15]:

- ViT-base available at <https://huggingface.co/google/vit-base-patch16-224>
- DeiT-small available at <https://huggingface.co/facebook/deit-small-patch16-224>
- SwinV2 available at https://huggingface.co/timm/swinv2_tiny_window8_256.ms_in1k

We initialize our MANO model by loading the full weights of the pretrained SwinV2-Tiny.

D. Data Augmentation

During training, in the case of image classification, we apply standard data augmentations to improve generalization. Specifically, the training pipeline includes:

- Resize to a fixed resolution, matching the input size expected by the pretrained models;
- RandomCrop with a crop size equal to the resized resolution, using a padding of 4 pixels;
- RandomHorizontalFlip;
- ToTensor conversion;
- Normalize using dataset-specific mean and standard deviation statistics.

At test time, images are resized (if necessary), converted to tensors, and normalized using the same statistics as in training.

For numerical simulations, we do not apply any data augmentation.