Supplementary material

Alex Colagrande[1], Paul Caillon[1], Eva Feillet[1], Alexandre Allauzen[1,2]
[1] Miles Team, LAMSADE, Université Paris Dauphine-PSL, Paris, France
[2] ESPCI PSL, Paris, France
{name}.{surname}@dauphine.psl.eu

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) [7] 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 pretrained on large-scale datasets such as ImageNet-21k [6] or LVD-142M [22], 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 [7].

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 [29] projects keys and values into a low-dimensional subspace ($k \ll N$), reducing per-head complexity from $O(N^2)$ to $O(Nk)$ while retaining competitive accuracy. Performer [5] uses a randomized feature map to approximate $\mathrm{softmax}(QK^\top) \approx \Phi(Q)\Phi(K)^\top$, achieving true $O(N)$ 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 [25] 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 $O(NC)$ instead of $O(N^2)$, 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) [31] 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 $O(N \cdot N/r^2)$. PVT matches CNN backbones in detection and segmentation.

Swin Transformer [16, 17] introduces *window-based MSA* over non-overlapping $M \times M$ patches (e.g., 7×7), reducing complexity to $O\left(\frac{N}{M^2} \times M^4\right)$. 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 [32] 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) [26] 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 [27] 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 [35] 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 [33] leverages a multiresolution decomposition of attention weights using wavelet transforms to capture both coarse and fine-scale dependencies.

FMMformer [21] builds on the Fast Multipole Method (FMM) to hierarchically group tokens and reduce attention complexity by summarizing distant interactions.

Fast Multipole Attention (FMA) [9] similarly applies FMM-inspired grouping but in a more generalizable attention framework.

ERWIN [34] 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 approache assumes the knowledge of the underlying PDE, like PINNs [10, 19, 23]. 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 [12, 24]. In a "semi-supervised" way, the recent approach of Boudec et al. [2] 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 [14, 18]. 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 [4] 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 solutions shares the linear complexity with our work but not the same expressivity. Still in the simplfyiing trend, LOCA (Learning Operators with Coupled Attention) [15] 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. [15] introduces an efficient transformer for the operator learning setting was proposed based on kernel theory. Recently in [3] was proposed an interesting way to see attention in the continuos setting and in particular the continuum patched attention. In Universal Physics Transformer [1] framework for efficient scaling was proposed based on a coarsoning of the input mesh. In [30] 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)** [13] extends multipole ideas to irregular domains via message-passing on graph hierarchies. Finally, **V-MGNO**, **F-MGNO**, and **W-MGNO** [20] 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 [11]. 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. [18], who leveraged a universal approximation theorem for nonlinear operators and paved the way for numerous extensions. Fourier

Neural operators [14] 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) [28] introduces wavelet-based multiscale localization, enabling kernels that simultaneously capture global structures and fine-grained details. The Multiwavelet Neural Operator (MWNO) [8] 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 1 summarizes the architectural and training hyperparameters used in our model. Below, we provide brief comments on each of them. he first block in Table 1 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 pretrained 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.

Hyperparameter	Value
`Patch size`	4
`Input channels`	3
`Embedding dimension` (`embed_dim`)	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` (`drop_rate`)	0.0
`Projection-drop rate` (`proj_drop_rate`)	0.0
`Attention-drop rate` (`attn_drop_rate`)	0.0
`Drop-path rate` (`drop_path_rate`)	0.2
`Activation layer`	`gelu`
`Normalization layer` (flag)	`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 1. MANO Hyperparameters for image classification

B.2. Architecture Hyperparameters for Darcy Flow

Table 2 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; 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.

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[6]:

- 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:

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 2. MANO Hyperparameters for Darcy flow

- `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.

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

[2] 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. 2

[3] Edoardo Calvello, Nikola B Kovachki, Matthew E Levine, and Andrew M Stuart. Continuum attention for neural operators. *arXiv preprint arXiv:2406.06486*, 2024. 2

[4] Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021. 2

[5] Krzysztof Choromanski, Valerii Likhosherstov, 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. 1

[6] 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. 1, 4

[7] 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

[8] Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. *Advances in neural information processing systems*, 34:24048–24062, 2021. 3

[9] Yanming Kang, Giang Tran, and Hans De Sterck. Fast multipole attention: A divide-and-conquer attention mechanism for long sequences, 2024. 2

[10] George Karniadakis, Yannis Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, pages 1–19, 2021. 2

[11] 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. 2

[12] 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. 2

[13] 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. 2

[14] 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, 3

[15] 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. 2

[16] 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. 1

[17] 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. 1

[18] 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

[19] 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. 2

[20] Leon Migus, 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. 2

[21] 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. 2

[22] 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

[23] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations, 2017. 2

[24] 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. 2

[25] 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. 1

[26] 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

[27] 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

[28] Tapas Tripura and Souvik Chakraborty. Wavelet neural operator: a neural operator for parametric partial differential equations. *arXiv preprint arXiv:2205.02191*, 2022. 3

[29] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 1

[30] 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. 2

[31] Wenhai 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. 1

[32] 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. 1

[33] Zhanpeng Zeng, Sourav Pal, Jeffery Kline, Glenn M Fung, and Vikas Singh. Multi resolution analysis (mra) for approximate self-attention, 2022. 2

[34] Maksim Zhdanov, Max Welling, and Jan-Willem van de Meent. Erwin: A tree-based hierarchical transformer for large-scale physical systems. *CoRR*, 2025. 2

[35] Zhenhai Zhu and Radu Soricut. H-transformer-1d: Fast one-dimensional hierarchical attention for sequences, 2021. 2