WaveLiT: A Parameter-Efficient Architecture for Neural PDE Solvers

Shyam Sankaran
Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania
shyamss@seas.upenn.edu

Hanwen Wang
Graduate Program in Applied Mathematics and Computational Science, University of Pennsylvania
wangh19@sas.upenn.edu

Paris Perdikaris
Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania
pgp@seas.upenn.edu

Abstract

Building surrogate models for complex physical systems governed by partial differential equations (PDEs) requires models capable of accurately resolving fine-scale features without prohibitive computational costs. While transformer-based models are promising, the quadratic complexity of self-attention often restricts the input sequence length, limiting the resolution of tokenized inputs. We introduce WaveLiT, a neural PDE solver designed for high-resolution problems by utilizing an efficient linear attention mechanism. WaveLiT utilizes a wavelet transform for input tokenization, generating feature-rich tokens that are then processed by its linear attention core, enabling favorable scaling to long sequences. Crucially, to enhance performance for high-frequency details, we incorporate a wavelet-domain L_1 loss on the prediction error. This combination allows WaveLiT to achieve exceptional performance and parameter efficiency across multiple PDE benchmarks, with competitive training speeds. Our findings underscore the power of leveraging efficient attention mechanisms to process finer-grained inputs, complemented by targeted loss functions, offering a potent and scalable recipe for building neural PDE solvers.

1 Introduction

The emergence of deep learning approaches for PDE modeling represents a paradigm shift in scientific computing, offering data-driven surrogate models capable of accelerating simulations by orders of magnitude while maintaining acceptable accuracy [2]. Neural operators – architectures specifically designed to learn mappings between function spaces [3] – have demonstrated remarkable promise in this domain, effectively capturing the underlying physics while generalizing across initial conditions, boundary values, and system parameters. From DeepONets [4] and Fourier Neural Operators [5] to more recent Vision Trans-

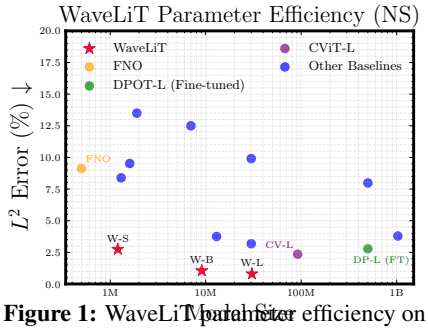

Figure 1: WaveLiT parameter efficiency on the PDEArena Navier-Stokes benchmark [1] (Appendix A). Lower L^2 error with fewer parameters (bottom-left) is better.

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

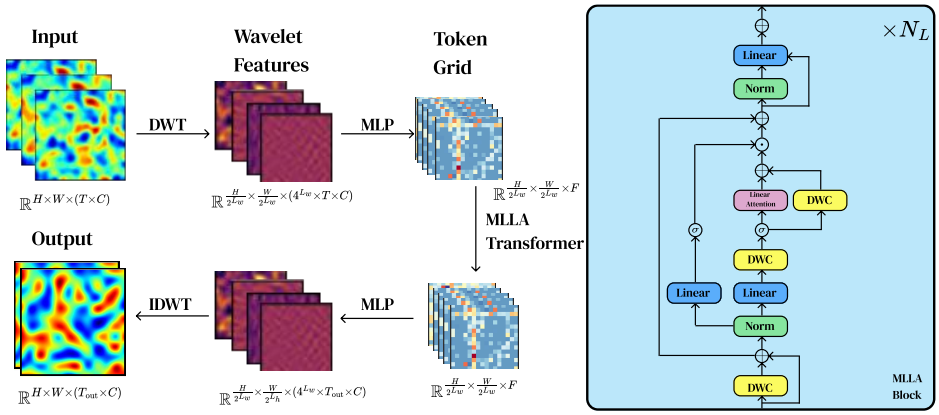

Figure 2: WaveLiT architecture: The core components of the WaveLiT architecture. Input fields are processed by a discrete wavelet transform and an MLP, passed through a series of MLLA blocks, and finally mapped to the output fields via a final MLP block and wavelet reconstruction.

former adaptations [6, 7, 8], these models have progressively pushed the boundaries of what is possible in data-driven scientific simulation.

Despite their powerful representational capabilities, transformer-based models face a fundamental challenge: the quadratic computational complexity with respect to input sequence length N. This limitation forces a critical compromise – either use coarse tokenization that loses crucial high-frequency physical details or accept computational costs that rapidly become infeasible as resolution increases. Recent studies in computer vision have demonstrated that increasing token resolution through smaller patch sizes consistently improves predictive performance, even up to pixel-level tokenization [9]. Here we argue this principle is even more critical for PDE operators, where resolving fine-scale structures is essential for accurate physical simulation. Specifically, our investigation confirms that while finer-grained tokenization significantly benefits transformer-based PDE operators, the quadratic attention cost quickly becomes prohibitive (Appendix C). Addressing this scalability challenge while preserving high-resolution details represents an important research direction. Recent advancements in efficient sequence modeling, such as linear attention mechanisms [10, 11] and State Space Models like Mamba [12], offer promising alternatives with linear or near-linear complexity. Concurrently, wavelet transforms provide a natural framework for multi-resolution analysis, effectively capturing features across different scales [13].

In this work, we bridge these advancements by introducing WaveLiT (Wavelet Linear Transformer), a lightweight neural PDE solver designed to efficiently process high-resolution inputs while maintaining state-of-the-art accuracy. WaveLiT achieves exceptional parameter efficiency, delivering superior results with significantly fewer parameters than existing methods (Figure 1). Our approach synergistically combines: (1) a wavelet transform for multi-scale input tokenization that preserves fine details, and (2) an efficient linear attention mechanism that scales favorably to long sequences. This combination, enhanced by a targeted wavelet-domain loss function, creates a powerful architecture for learning complex physical dynamics across multiple challenging benchmarks.

2 Wavelet Linear Transformer (WaveLiT)

WaveLiT is a scalable neural operator designed to efficiently model complex PDE systems while preserving high-resolution details. Built on the Vision Transformer paradigm but with crucial modifications for efficiency and accuracy, our architecture consists of three main components: (1) a Wavelet Patch Embedding layer, (2) a stack of Mamba-like Linear Attention (MLLA) blocks [14], and (3) an inverse wavelet transform for solution reconstruction. Figure 2 provides an overview of this architecture. While we demonstrate our approach on 2D problems, the pipeline extends naturally to higher-dimensional domains on Cartesian grids, but also to complex geometric domains via the wavelet lifting scheme [15]. The full details on the architectural design can be found in Appendix G.

Wavelet Patch Embedding. While standard transformer models typically use linear patch embedding techniques that could theoretically learn multi-scale representations, we explicitly incorporate

the multi-scale structure of wavelet transforms to provide a stronger inductive bias. This approach leverages the well-established mathematical properties of wavelets for analyzing signals across multiple scales while maintaining spatial locality – a critical feature for modeling physical systems.

Mamba-like Linear Attention (MLLA) Block. The second key component of WaveLiT is our attention mechanism, designed to efficiently process the wavelet-embedded tokens. The MLLA block [14] integrates linear attention with depth-wise convolutions to capture both global dependencies and local features efficiently. This enables linear scaling $\mathcal{O}(N)$ with respect to sequence length – a critical improvement over the quadratic complexity $\mathcal{O}(N^2)$ of standard self-attention.

The final component of WaveLiT is the inverse wavelet transform (IDWT) that maps the processed feature representation back to the original spatial domain. This transform mirrors the DWT process, using corresponding reconstruction filters and transposed convolutions to progressively restore the spatial resolution of the solution field. The IDWT provides an efficient way to generate high-resolution outputs while maintaining the multi-scale characteristics of the underlying physical system.

Wavelet Loss Function. Neural networks trained with mean squared error (MSE) typically exhibit spectral bias [16]; a well-documented phenomenon where high-frequency components of the target function are learned significantly later in training, even in over-parameterized networks. Drawing inspiration from computer vision, where perceptual losses have proven effective at preserving fine textures and local features beyond what MSE captures [17], we incorporate an auxiliary wavelet-domain loss that specifically aims to match underlying wavelet coefficients for the targets and the predictions leading to better representation of these high-frequency components. While recent approaches like Binned Spectral Power Loss [18] have addressed similar goals through frequency-space constraints, we leverage wavelet theory to provide a more principled and mathematically grounded approach. This motivation is detailed in Appendix B.

3 Experiments

Implementation Details. We evaluate several WaveLiT variants, differing in depth and width, to assess scalability. Configurations (WaveLiT-S, WaveLiT-B, WaveLiT-L) are detailed in Table 11 in Appendix G. All WaveLiT models are trained using the AdamW optimizer [19] using an exponential decay in learning rate. Additionally, we enforce gradient clipping to avoid large updates. We restrict ourselves to the use of a single level wavelet transform which leads to a maximal number of tokens for all cases except for the `helmholtz_staircase` example from TheWell (see below and Appendix F). We use the combined MSE and wavelet loss (Section 2). For the purpose of evaluation, we utilize the exponentially averaged parameters which often lead to improved stability and performance. Training is performed using JAX [20] and Flax [21]. For the wavelet transform, we leverage the `jax-wavelets` library [22]. Training progress and metrics are logged using Weights & Biases [23] and will be made available post publication. Further details about training have been detailed in Appendix F.

Evaluation Metrics. We use the Variance Scaled Root Mean Squared Error (VRMSE) staying consistent with the baselines that were reported in the original paper.

Experiments Performed For our evaluations, we consider challenging benchmarks from TheWell [24] utilizing eight different 2D cases that span a diverse range of physical phenomena. We compare three variants of WaveLiT, differing in model size (Small: WaveLiT/S $\approx$ 1.2M parameters; Base: WaveLiT/B $\approx$ 9M parameters; Large: WaveLiT/L $\approx$ 30.5M parameters), against established baselines, including the Fourier Neural Operator (FNO) [5], tensorized-FNO [25], U-Net [26], and a ConvNeXt-U-Net (CNextU-net) [27].

Table 1 demonstrates WaveLiT's strong performance on the TheWell datasets. Our experiments are performed using teacher forcing by taking in 4 input steps and outputting the next output step. Our approach achieves significant improvements over existing baselines, ranging from 25% to 97%, on seven out of eight benchmarks. Most impressively, WaveLiT achieves high accuracy on challenging problems like Gray-Scott reaction-diffusion (95.8% improvement), Rayleigh-Bénard convection (96.9%), and shear flow (97.8%). Even our smallest WaveLiT-S model (1.2M parameters) consistently outperforms larger established baselines on most tasks. These results are particularly

Table 1: Comparison of model performance (VRMSE) on selected 2D datasets from THEWELL. WaveLiT variants are Small (S, ≈1.2M params), Base (B, ≈9M params), and Large (L, ≈30.5M params). %Impr. rows give the percentage improvement of each WaveLiT variant over the best non-WaveLiT baseline, computed as $(\text{Prev. Best} - \text{WaveLiT})/\text{Prev. Best} \times 100\%$. Higher is better for %Impr.; lower is better for the error metric. Best results are made bold, and the second best results are underlined.

Model	AM	GSRD	RB	SF	TRL2D	VI	ASM	HS
FNO	0.3691	0.1365	0.8395	1.189	0.5001	0.7212	0.5062	**0.00046**
TFNO	0.3598	0.3633	0.6566	1.472	0.5016	0.7102	0.5057	0.00346
U-Net	0.2489	0.2252	1.4860	3.447	0.2418	0.4185	0.0351	0.01931
CNextU-Net	0.1034	0.1761	0.6699	0.8080	0.1956	0.2499	0.0153	0.02758
WaveLiT/S	0.0605	<u>0.0074</u>	0.0552	0.0606	0.1755	0.1856	0.0087	0.00064
% Impr. vs Best	*41.5%*	*94.6%*	*91.6%*	*92.5%*	*10.3%*	*25.7%*	*43.1%*	*-39.13%*
WaveLiT/B	**0.0293**	**0.0058**	<u>0.0261</u>	**0.0176**	<u>0.1346</u>	<u>0.1136</u>	<u>0.0040</u>	0.00044
% Impr. vs Best	*71.7%*	*95.8%*	*96.0%*	*97.8%*	*31.2%*	*54.5%*	*73.9%*	*4.35%*
WaveLiT/L	<u>0.0334</u>	**0.0058**	**0.0202**	<u>0.0311</u>	**0.1308**	**0.0821**	**0.0029**	<u>0.00044</u>
% Impr. vs Best	*67.7%*	*95.8%*	*96.9%*	*96.2%*	*33.1%*	*67.1%*	*81.0%*	*4.35%*

noteworthy considering we maintained consistent hyperparameters across all benchmarks and model sizes, without task-specific tuning, leaving room for further improvement via exploring principled hyperparameter transfer methodologies like μP [28].

Token Resolution vs Physical Correlation Length. Our results show that while WaveLiT excels on benchmarks with short correlation lengths, FNO's global fourier modes are better suited for the Helmholtz staircase (HS) problem, which exhibits long-range correlations. On this benchmark, a coarser 2-level WaveLiT paradoxically outperforms the finer 1-level version. We discuss this in detail in Appendix H.

Performance on Long Rollouts We evaluate the performance on long rollouts to assess model stability and error accumulation. The rollout is done autoregressively by repeatedly applying the model. Namely, the prediction for the next state would be appended to the current sequence of snapshots, and the earliest snapshot would be discarded so that the sequence length is fixed. The updated sequence is then fed into the model again for a prediction which is 2 time steps away. Continuing on, this process is applied K-times to generate a forecasting up to K time steps in the future. As shown in Table 2, WaveLiT largely outperforms original TheWell baselines (such as FNO, U-Net) at this longer horizon. However, they aren't as performant as more recent works that are specifically optimized for long-term stability[29, 30]. This believe this performance gap is primarily due to three distinct design facets:

(i) WaveLiT is trained as an end-to-end, next-step predictor using an autoregressive teacher forcing model. This is known to cause a train-test mismatch; over a long enough rollout, compounding errors accumulate, leading to error growth [31, 32].

(ii) Many state-of-the-art models utilize a two-stage process, first training a powerful autoencoder often with specialized perceptual or adversarial losses that performs *significant temporal compression*. For example, with a compression factor of 4, a 40-step physical rollout only requires 10 autoregressive steps in the latent space. In contrast, our next-step model must perform 40 sequential evaluations, providing 4x more opportunities for error to compound.

(iii) Our architecture lacks a selective memory mechanism or recurrent state (akin to a KV-cache or Mamba state) that holds context over past several windows, which can limit its understanding of long-term temporal dependencies.

Our future work will look into incorporating these facets, such as corrective training strategies[31, 33], latent space temporal compression, and efficient recurrent states to enhance performance on long-horizon rollouts.

Table 2: Comparison of the model performance (VRMSE) for autoregressive rollouts on selected 2D datasets from THEWELL, with two different time windows, from the 6th to the 12th of the rollout predictions, and from 13rd to 30th of the rollout predictions. The error metric is aboved over the prediction snapshots. WaveLiT variants are Small (S, ≈1.2M params), Base (B, ≈9M params), and Large (L, ≈30.5M params). Lower is better for the error metric. Best results are made bold, and the second best results are underlined.

Model	**FNO**		**TFNO**		**U-net**		**CNextU-net**		**WaveLiT/S**		**WaveLiT/B**		**WaveLiT/L**	
Time Window	6:12	13:30	6:12	13:30	6:12	13:30	6:12	13:30	6:12	13:30	6:12	13:30	6:12	13:30
ASM	1.06	1.72	1.13	1.23	0.56	0.92	0.78	1.13	**0.015**	0.025	0.015	<u>0.022</u>	0.020	0.044
AM	>10	>10	7.52	4.72	2.53	2.62	2.11	2.71	0.772	1.477	0.339	<u>0.945</u>	**0.336**	0.977
GSRD	0.89	>10	1.54	>10	0.57	>10	0.29	7.62	0.033	<u>0.094</u>	**0.023**	0.167	0.329	0.465
HS	0.002	0.003	0.011	0.019	0.057	0.097	0.11	0.194	0.001	0.002	**0.001**	<u>0.002</u>	0.011	0.024
RB	>10	>10	>10	>10	>10	>10	>10	>10	0.541	1.142	0.132	0.399	**0.101**	<u>0.367</u>
SF	>10	>10	>10	>10	>10	>10	2.33	>10	0.558	2.689	**0.187**	1.051	0.190	<u>0.483</u>
TRL2D	1.79	3.54	6.01	>10	0.66	1.04	0.54	1.01	0.687	1.023	0.562	0.874	**0.522**	<u>0.808</u>
VI	4.11	-	0.93	-	0.89	-	**0.52**	-	0.783	-	0.715	-	0.624	-

4 Discussion

Our introduction of WaveLiT demonstrates how strategically combining wavelet transformations, linear attention mechanisms, and targeted loss functions creates a neural PDE solver with exceptional parameter efficiency and competitive performance across diverse benchmarks.

Summary of Findings. WaveLiT's success stems from three synergistic design principles addressing core PDE modeling challenges: high-resolution tokenization via discrete wavelet transforms that preserve multi-scale features and sharp gradients; compute-optimal linear attention scaling as $O(N)$ instead of $O(N^2)$, enabling efficient processing of longer token sequences; and a theoretically-grounded wavelet-domain perception loss that counters spectral bias across the frequency spectrum. This combination delivers state-of-the-art results with remarkable parameter efficiency – WaveLiT-L achieves superior performance on the Navier-Stokes benchmark with only 30.5M parameters, outperforming DPOT-L (500M) and CViT-L (92M). These results demonstrate that principled architectural design can substantially outperform larger models by directly addressing the computational and representational challenges inherent in PDE modeling.

Limitations & Future Work. Despite its strong performance, WaveLiT has important limitations that present opportunities for future research. First, the current architecture operates exclusively on regular Cartesian grids, although an extension to complex geometries and irregular meshes is possible via the wavelet lifting scheme [15]. Second, while linear in complexity, our attention mechanism is not adaptive to varying information densities across the spatial domain. For extremely high-resolution problems (beyond 1024^2) or higher-dimensional domains, incorporating adaptive tokenization strategies such as token merging [34] could further optimize computational resources. Future work will also address long-term autoregressive stability, by exploring corrective training mechanisms and efficient recurrent states to mitigate error accumulation. Despite these limitations, WaveLiT's parameter efficiency and strong generalization capabilities make it a promising foundation for large-scale scientific foundation models [35, 36].

Conclusions. WaveLiT represents a significant advance in neural PDE solvers by demonstrating that compute-optimal architectures can outperform much larger models through principled design choices. Our work provides both practical tools for immediate application and architectural insights that can guide future developments in data-driven scientific simulation. Code, model weights, and all experimental logs will be made publicly available to facilitate further research.

Acknowledgements

The authors gratefully acknowledge the support of the U.S. Department of Energy (DOE), Office of Advanced Scientific Computing Research (ASCR) under award number DE-SC0024563.

References

[1] Jayesh K Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling. *arXiv preprint arXiv:2209.15616*, 2022.

[2] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[3] 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.

[4] 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.

[5] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[6] 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.

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

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

[9] Feng Wang, Yaodong Yu, Guoyizhe Wei, Wei Shao, Yuyin Zhou, Alan Yuille, and Cihang Xie. Scaling laws in patchification: An image is worth 50,176 tokens and more. *arXiv preprint arXiv:2502.03738*, 2025.

[10] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[11] 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.

[12] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[13] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.

[14] Dongchen Han, Ziyi Wang, Zhuofan Xia, Yizeng Han, Yifan Pu, Chunjiang Ge, Jun Song, Shiji Song, Bo Zheng, and Gao Huang. Demystify mamba in vision: A linear attention perspective. *arXiv preprint arXiv:2405.16605*, 2024.

[15] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM journal on mathematical analysis*, 29(2):511–546, 1998.

[16] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016.

[18] Dibyajyoti Chakraborty, Arvind T Mohan, and Romit Maulik. Binned spectral power loss for improved prediction of chaotic systems. *arXiv preprint arXiv:2502.00472*, 2025.

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[20] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[21] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2024.

[22] Katherine Crowson. jax-wavelets: The 2D discrete wavelet transform for JAX, 2022.

[23] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[24] Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Agocs, Miguel Beneitez, Marsha Berger, Blakesly Burkhart, Stuart Dalziel, Drummond Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

[25] Jean Kossaifi, Nikola Kovachki, Kamyar Azizzadenesheli, and Anima Anandkumar. Multi-grid tensorized fourier neural operator for high-resolution pdes. *arXiv preprint arXiv:2310.00120*, 2023.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[27] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[28] Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

[29] François Rozet, Ruben Ohana, Michael McCabe, Gilles Louppe, François Lanusse, and Shirley Ho. Lost in latent space: An empirical study of latent diffusion models for physics emulation. *arXiv preprint arXiv:2507.02608*, 2025.

[30] Tung Nguyen, Arsh Koneru, Shufan Li, and Aditya Grover. Physix: A foundation model for physics simulations. *arXiv preprint arXiv:2506.17774*, 2025.

[31] Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. *Advances in Neural Information Processing Systems*, 36:67398–67433, 2023.

[32] Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

[33] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

[34] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

[35] Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. Dpot: Auto-regressive denoising operator transformer for large-scale pde pre-training. *arXiv preprint arXiv:2403.03542*, 2024.

[36] Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

[37] Eugenio Hernández and Guido Weiss. *A first course on wavelets*. CRC press, 1996.

[38] Lukas Prantl, Jan Bender, Tassilo Kugelstadt, and Nils Thuerey. Wavelet-based loss for high-frequency interface dynamics. *arXiv preprint arXiv:2209.02316*, 2022.

[39] 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. *arXiv preprint arXiv:2010.11929*, 2020.

[40] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113, 2022.

[41] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[42] Albert Cohen, Ingrid Daubechies, and Jean-Christophe Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45:485–560, 1992.

[43] GitHub - crowsonkb/jax-wavelets: The 2D discrete wavelet transform for JAX — github.com. `https://github.com/crowsonkb/jax-wavelets?tab=readme-ov-file`. [Accessed 13-05-2025].

[44] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[45] WH Matthaeus, CW Smith, and JW Bieber. Correlation lengths, the ultrascale, and the spatial structure of interplanetary turbulence. In *AIP Conference Proceedings*, volume 471, pages 511–514. American Institute of Physics, 1999.

[46] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[47] 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.

A Results on PDEArena

Evaluation Metrics. For PDEArena [1], we report the relative L_2 error for rollout predictions.

Shallow-Water Equations (SWE). Shallow-water equations benchmark from PDEArena [1]. Table 3 compares different sizes of WaveLiT (WaveLiT-S, -B, -L) against competitive baselines, including those reported by CViT [6]. Our results highlight the parameter efficiency of our architecture. The largest variant, WaveLiT-L ($\approx$ 30.5M parameters), achieves a state-of-the-art relative L_2 error of 1.55% on 5-step rollouts. This result marginally surpasses the performance of CViT-L (92M parameters, 1.56% error), a strong transformer-based baseline, while utilizing approximately one-third of the parameters.

Table 3: SWE (PDEArena): 5-step rollout performance (Relative L^2 error $\downarrow$). Baselines from PDEArena and CViT.

Model	# Params	Rel. L^2 error
DilResNet	4.2 M	13.20%
U-Net$_{\text{att}}$	148 M	5.68%
FNO	268 M	3.97%
U-F2Net	344 M	1.89%
UNO	440 M	3.79%
CViT-S	13 M	4.47%
CViT-B	30 M	2.69%
CViT-L	92M	<u>1.56%</u>
WaveLiT-S	1.2M	8.66%
WaveLiT-B	9.1M	2.39%
WaveLiT-L	30.5M	**1.55%**

Table 4: Relative L^2 error of rollout predictions on the incompressible Navier-Stokes benchmark [1].

Model	# Params	Rel. L^2 error
FNO	0.5 M	9.12 %
FFNO	1.3 M	8.39 %
GK-T	1.6 M	9.52%
GNOT	1.8 M	17.20 %
Oformer	1.9 M	13.50 %
DPOT-Ti	7 M	12.50 %
DPOT-S	30 M	9.91 %
DPOT-L (Pre-trained)	500 M	7.98 %
DPOT-L (Fine-tuned)	500 M	2.78 %
DPOT-H (Pre-trained)	1.03 B	3.79 %
CViT-S	13 M	3.75 %
CViT-B	30 M	3.18 %
CViT-L	92 M	2.35 %
WaveLiT-S	1.2M	2.74 %
WaveLiT-B	9.1M	<u>1.05 %</u>
WaveLiT-L	30.5M	**0.80 %**

Navier-Stokes Equations (NS). We further assess WaveLiT on the challenging 2D incompressible Navier-Stokes benchmark from PDEArena using the setup established by DPOT [35] and compare against competitive baselines. The results, presented in Table 4, demonstrate WaveLiT's strong performance and efficiency. Our largest variant, WaveLiT-L (≈30.5M parameters), achieves a relative L^2 error of 0.80% in rollout predictions. This significantly outperforms all other reported baselines, including the heavily pre-trained and fine-tuned DPOT-L (500M params, 2.78% error from [35]) and the larger CViT-L (92M params, 2.35% error from [6]).

B Wavelet Loss

The key theoretical foundation for our wavelet loss comes from the established equivalence between wavelet coefficient norms and Sobolev norms. For a function f, the wavelet transform produces coefficients that effectively capture both spatial localization and frequency content. Specifically, let $\|\cdot\|_{L^{p,s}(\mathbb{R})}$ denote the Sobolev norm of order s, $\chi_{[a,b]}$ represent the characteristic function for interval $[a, b]$, and define:

$$\mathcal{W}_\psi^s(f)(x) = \left\{ \sum_{j\in\mathbb{Z}} \sum_{k\in\mathbb{Z}} |\langle f, \psi_{j,k}\rangle|^2 \left(1 + 2^{2js}\right) 2^j \chi_{[2^{-j}k,\, 2^{-j}(k+1)]}(x) \right\}^{1/2}, \tag{1}$$

Theorem 6.18 in [37] states that:

Theorem 1 *Let $\psi \in \mathcal{S}$ be a band-limited orthonormal wavelet. For $1 < p < \infty$ and $s = 1, 2, \cdots$, there exist two constants $A_{p,s}$ and $B_{p,s}$, $0 < A_{p,s} \le B_{p,s} < \infty$, such that*

$$A_{p,s}\|f\|_{L^{p,s}(\mathbb{R})} \le \left\|\mathcal{W}_\psi^s f\right\|_{L^p(\mathbb{R})} \le B_{p,s}\|f\|_{L^{p,s}},$$

for all $f \in L^{p,s}(\mathbb{R})$.

This establishes that the wavelet coefficient norm is equivalent to the Sobolev norm, which measures both function values and their derivatives. Unlike previous work [38] that employed wavelet losses in an ad-hoc manner, our approach is directly motivated by this theoretical foundation, enforcing constraints in the wavelet domain effectively serves as regularization on both the function and its derivatives, encouraging more physically accurate solutions.

Based on this insight, we formulate our wavelet-domain loss. Given the ground truth y and prediction $\hat{y}$, we compute their difference $d = y - \hat{y}$ and decompose it into wavelet coefficients using a wavelet transform:

$$\Phi^{(l)}(d) = \{d_{LL}, d_{LH}, d_{HL}, d_{HH}\}. \tag{2}$$

Each component captures different aspects of the error: d_{LL} represents low-frequency approximation errors, while d_{LH}, d_{HL}, and d_{HH} capture directional and high-frequency errors. In the general form, our wavelet loss can be expressed as:

$$\mathcal{L}_{\text{wavelet}}(d) = \sum_{\ell=0}^{L-1} \sum_{b\in\{\text{LL,LH,HL,HH}\}} w_{\ell,b} \left\|\Phi_b^{(\ell)}(d)\right\|_p, \tag{3}$$

where $w_{\ell,b}$ are weights that can modulate the contribution of different frequency bands. In our implementation, we specifically use a 1-level DWT and set all weights to 1, effectively creating an L^1 loss over the wavelet coefficients:

$$\mathcal{L}_{\text{wavelet}} = \|d_{LL}\|_1 + \|d_{LH}\|_1 + \|d_{HL}\|_1 + \|d_{HH}\|_1. \tag{4}$$

The total loss combines this wavelet-domain constraint with traditional MSE:

$$\mathcal{L}_{\text{total}} = w_{\text{mse}}\mathcal{L}_{\text{MSE}} + w_{\text{wavelet}}\mathcal{L}_{\text{wavelet}}. \tag{5}$$

In our experiments, we set $w_{\text{mse}} = w_{\text{wavelet}} = 1$, giving equal weight to both terms. While our current approach uses uniform weights, modulating $w_{\ell,b}$ could introduce an effective inductive bias towards specific frequency bands – for instance, emphasizing high-frequency details in turbulent flows or preserving low-frequency structures in diffusion problems. This represents a promising direction for future research. Our ablation studies (Appendix E) demonstrate that even our simple uniform weighting consistently improves performance across benchmarks, yielding models that more accurately capture the multi-scale dynamics of complex physical systems.

C Impact of Sequence Length and Attention Mechanism on Performance and Efficiency

We explore the interplay between model architecture, effective sequence length (modulated by wavelet decomposition levels), and computational cost.

Our primary goal is to identify model configurations that balance high accuracy with manageable computational cost. We compare standard Dot-Product Attention (DPA) with an enhanced linear attention mechanism (MLLA) [14]. To understand these trade-offs, we conducted evaluations on the Navier-Stokes (NS) benchmark from PDEArena [1], following the experimental setup established in DPOT [35]. We compared models using both DPA and MLLA across different model sizes ("WaveLiT-S" vs. "WaveLiT-B") and wavelet decomposition levels (1 vs. 2 levels, where fewer levels mean longer effective sequences). The results, including relative L_2 error and training time per 1000 iterations, are presented in Table 5 and visualized for key configurations in Figure 3. While detailed results are shown for NS, similar trends in the DPA vs. MLLA trade-off were observed across other datasets.

Table 5: Performance and efficiency comparison of Dot-Product Attention (DPA) and Enhanced Linear Attention (MLLA) across varying model sizes and wavelet levels. Training time is per 1000 iterations.

Attention	Model	Wavelet Levels	Params (M)	Training Time (s)	Relative L_2
DPA	WaveLiT-B	1	≈ 8.5	212.80	0.00458
MLLA	WaveLiT-B	1	≈ 8.5	60.00	0.00480
DPA	WaveLiT-B	2	≈ 8.7	26.68	0.00711
MLLA	WaveLiT-B	2	≈ 8.7	15.00	0.00750
DPA	WaveLiT-S	1	≈ 1.1	53.30	0.01848
MLLA	WaveLiT-S	1	≈ 1.1	18.00	0.01950
DPA	WaveLiT-S	2	≈ 1.2	7.12	0.02606
MLLA	WaveLiT-S	2	≈ 1.2	5.00	0.02700

Discussion. The data in Table 5 (visualized in Figures 3 and 4) reveals several key trends regarding model performance and efficiency. Firstly, for a given model size and attention type, decreasing the number of wavelet levels (thus increasing effective sequence length) generally improves performance (lower relative L_2 error), albeit at a higher computational cost, especially for DPA. This underscores the benefit of processing longer sequences for capturing complex dynamics, a point also linked to the characteristic correlation length of the data (see Appendix H). Secondly, increasing model parameters (e.g., from WaveLiT-S to WaveLiT-B) also tends to enhance performance. However, as detailed in Appendix F, input sequence length (determined by wavelet levels) often plays a more decisive role than raw parameter count alone, particularly for smaller models.

Most critically, these results highlight the compelling advantages of MLLA. While DPA occasionally achieves marginally lower error, MLLA consistently delivers comparable accuracy at a substantially reduced computational cost. For instance, with WaveLiT-B and 1 wavelet level, MLLA trains over 3.5 times faster than DPA for a very small trade-off in error. Therefore, MLLA emerges as a highly practical and scalable attention mechanism. It allows leveraging the performance benefits of longer effective sequences, crucial for resolving fine-scale features in PDE solutions, without incurring the prohibitive computational costs.

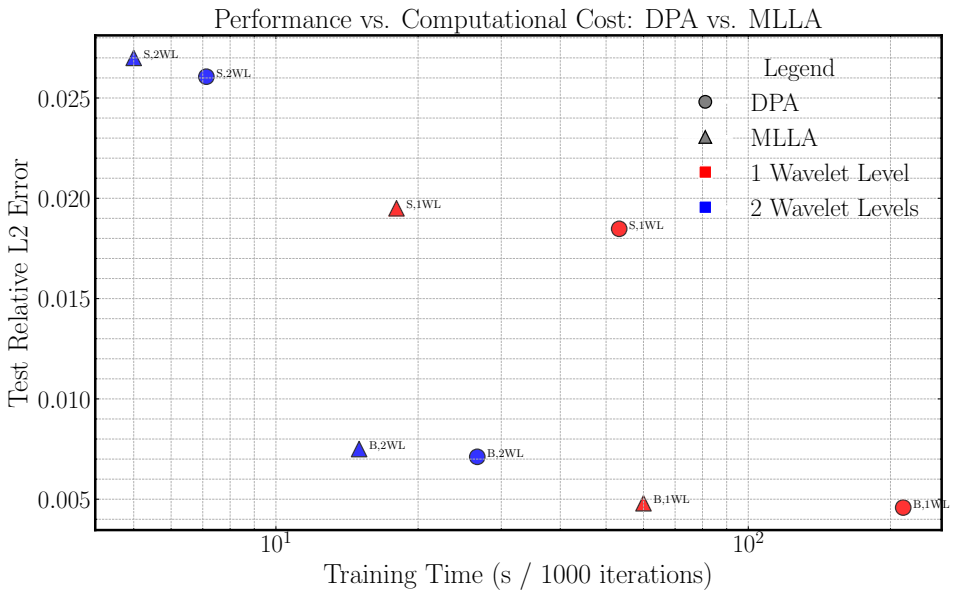

Figure 3: MLLA models (triangles) demonstrate significantly lower compute costs for comparable performance levels, especially at 1 wavelet level (longer sequences), compared to DPA models (circles)

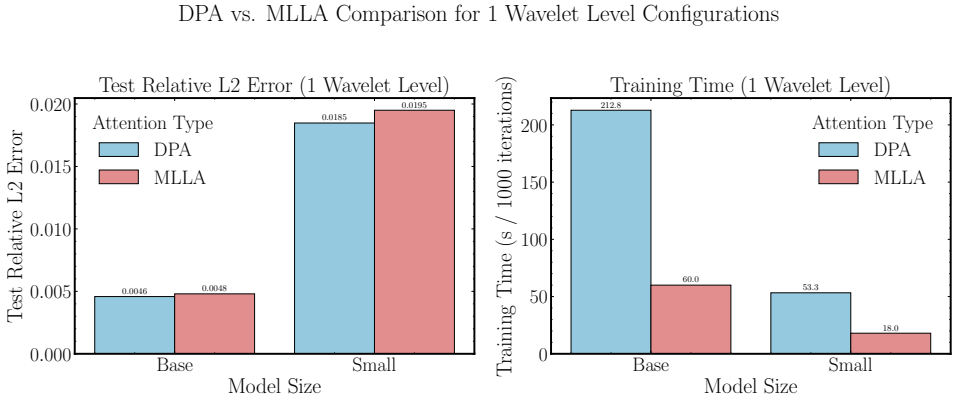

Figure 4: Comparative performance of Dot-Product Attention (DPA) and Enhanced Linear Attention (MLLA) for 1-wavelet-level (long sequence) configurations. *Left*: Test Relative L2 Error. *Right*: Training Time (sec/1000 iterations). MLLA demonstrates significant reductions in training time while maintaining competitive error rates compared to DPA across all model sizes.

D Wavelet Selection Choice

For the core experiments in this work, the Biorthogonal wavelet `bior2.2` was selected as the default choice for the wavelet transform layer. The `bior2.2` wavelet offers a good balance of properties, including symmetry and relatively compact support (filter length of 6), making it suitable for capturing features across various scales without excessive computational overhead.

To validate this choice and assess the model's robustness to the specific wavelet employed, we conducted an ablation study. We evaluated the performance of the WaveLiT-S model on the Navier-Stokes benchmark dataset provided by PDEArena [1] using three distinct wavelets: `haar`, `bior2.2`, and `bior4.4`. These wavelets represent a range of characteristics: `haar` is the simplest, discontinuous orthogonal wavelet (filter length 2); `bior2.2` is our chosen symmetric biorthogonal wavelet (filter length 6); and `bior4.4` is a smoother, higher-order symmetric biorthogonal wavelet (filter length 10). Our wavelet selection was guided by the `jax-wavelets` library [22], specifically choosing from families compatible with reflect padding, which we found to yield better performance than wrap padding in our experiments.

Discussion. The performance was measured using the relative-L_2 error for 1-step ahead predictions and 4-step rollouts. The results, presented in Table 6, demonstrate consistency across the different wavelet choices. This minimal variation in performance across wavelets with differing characteristics suggested that our architecture is reasonably robust to the specific choice of wavelet and we stick to the use of `bior2.2` as the default choice.

Table 6: Ablation study on wavelet choice for the `WaveLiT-S` model on the Navier-Stokes benchmark. Reported values are relative-L_2 error for 1-step ahead and 4-step rollout predictions. Lower is better.

Wavelet	1-Step Ahead	4-Step Rollout
`haar`	0.010669	0.023265
`bior2.2`	0.010618	0.023349
`bior4.4`	0.010896	0.023718

E Loss Term Ablations

Table 7 presents the results of experiments investigating the impact of different loss term weightings on model performance, specifically comparing Mean Squared Error (MSE) loss and an L_1 wavelet loss (L_1 wavelet). The experiments were conducted for two model sizes (Small and Base) and two different numbers of wavelet decomposition levels (1 and 2).

Table 7: Effect of MSE and L1 Wavelet Loss Terms on Model Performance. Performance metrics are relative L_2 error on the test set for single-step prediction (Test Rel. L_2) and 4-step rollout (Test Rel. L_2 Rollout). Best results for each ablation are highlighted in bold.

Model	Wavelet	Loss Weights		Test Performance	
	Levels	λ_{MSE}	λ_{L1}	Rel. L_2	Rel. L_2 Rollout (4 steps)
WaveLiT-B	1	0	1	**0.00416**	0.01028
WaveLiT-B	1	1	1	0.00453	**0.01009**
WaveLiT-B	1	1	0	0.00563	0.01295
WaveLiT-B	2	0	1	**0.00662**	**0.01634**
WaveLiT-B	2	1	1	0.00670	0.01636
WaveLiT-B	2	1	0	0.00837	0.01972
WaveLiT-S	1	0	1	0.01177	0.02792
WaveLiT-S	1	1	1	**0.01167**	**0.02702**
WaveLiT-S	1	1	0	0.01280	0.02945
WaveLiT-S	2	0	1	0.01407	**0.03283**
WaveLiT-S	2	1	1	**0.01406**	0.03295
WaveLiT-S	2	1	0	0.01515	0.03369

Discussion A clear trend emerges from the data: the inclusion of the L_1 wavelet loss term, either exclusively ($\lambda_{L1} = 1, \lambda_{MSE} = 0$) or in conjunction with the MSE loss ($\lambda_{L1} = 1, \lambda_{MSE} = 1$), consistently yields superior performance compared to using only the MSE loss ($\lambda_{L1} = 0, \lambda_{MSE} = 1$). This benefit of the wavelet loss is further demonstrated by examining the spectral characteristics of the prediction error. Figure 5 shows the Radially Averaged Power Spectral Density (RAPSD) of the prediction error for the WaveLiT-B model (1 wavelet level) when trained with the combined MSE and L_1 wavelet loss versus MSE loss alone. The plot indicates that the incorporation of the wavelet loss term leads to a reduction in error power across the entire frequency spectrum compared to the model trained without it. This reduction is evident at low frequencies, corresponding to large-scale structures, and extends to higher frequencies, representing finer details. Such broad spectral improvement suggests that the L_1 wavelet loss aids the model in more accurately resolving the multi-scale features inherent in the solution.

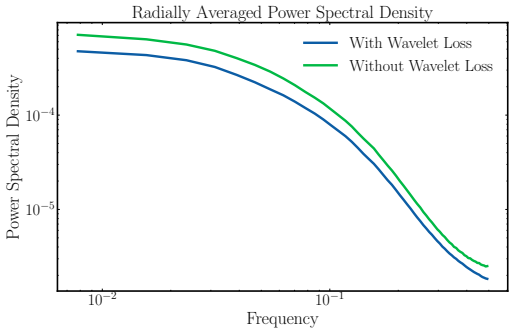

Figure 5: Radially Averaged Power Spectral Density (RAPSD) of the prediction error for the WaveLiT-B model (1 wavelet level), comparing training with both MSE and L_1 wavelet loss terms ("With Wavelet Loss") versus training with MSE loss alone ("Without Wavelet Loss"). The inclusion of the wavelet loss demonstrably reduces error power across all frequencies.

Given these observations, we find that setting both $\lambda_{MSE} = 1$ and $\lambda_{L1} = 1$ provides robust, high-quality results. Therefore, for all experiments in this paper, we adopted this balanced weighting, setting both the MSE loss weight and the L_1 wavelet loss weight to 1.

F Training Details

All models are trained using the AdamW optimizer [19], using a weight decay rate of 1×10^{-4}. We employ an exponential decay learning rate scheduler with a linear warm-up phase: the learning rate is linearly scaled up from 1×10^{-7} to 1×10^{-3} over the first 5,000 steps, after which it is exponentially decayed with a decay rate of 0.99 every 2,000 transition steps. To ensure training stability, gradients are globally clipped to a norm of 1. The batch size is set to 8 for all benchmarks, with exceptions for WaveLiT-B on the VI and SF benchmarks, where a batch size of 2 was used due to memory constraints. We default to a Discrete Wavelet Transform (DWT) level of 1 for most benchmarks to maximize input sequence length and provide the model with fine-grained details. An exception is made for the Helmholtz Staircase (HS) dataset, where the DWT level is set to 3. This choice for HS is informed by the autocorrelation analysis presented in Section H, which indicated a longer characteristic correlation length for this dataset, making a coarser input resolution more suitable.

Table 8 outlines the projected total training times for 500,000 steps for various WaveLiT models across different benchmarks, without intermediate evaluation. It is important to note that these times are estimates from running on a shared NVIDIA H200 server and can vary depending on factors such as data loading efficiency, software library versions, and computational facility workload; thus, they may not perfectly and consistently reflect the raw computational cost.

Table 8: Projected total training times (HH:MM) for each WaveLiT model and benchmark.

Benchmark	Sequence Length	WaveLiT-S	WaveLiT-B	WaveLiT-L
AM	16384	09:01	18:30	32:03
GSRD	4096	02:29	09:41	09:55
RB	16384	07:28	33:13	34:25
SF	32768	07:10	11:59	18:36*
TRL2D	12288	05:15	12:34	24:04
VI	65536	21:52	53:44	31:42*
ASM	16384	06:49	16:31	33:51
HS**	4096	05:56	07:04	10:38
SWE	4608	01:02	04:02	09:57
NS	4096	01:54	08:54	24:04

*Batch size is set to 2.

**DWT level is set to 3, for effectively patch size 8.

Impact of Model Size and Wavelet Levels. To evaluate the impact of model scale and input resolution (controlled by DWT levels) on performance and computational cost, we define several WaveLiT configurations. We adopt the common -Small (S), -Base (B), and -Large (L) naming convention [39, 40], with hyperparameters adapted for PDE modeling, as detailed in Table 9. The number of DWT levels directly influences the input sequence length L and thus the computational FLOPs, as shown in Table 10. These FLOP estimates were obtained via JAX AOT analysis [20] for a $128 \times 128 \times 3$ input, and training FLOPs are typically $\approx 3\times$ forward FLOPs [41].

Table 9: WaveLiT model configurations evaluated. N: Number of Layers, d_e: Embedding Dimension, h: Number of Heads, d_{ff}: Feed-Forward Network Dimension ($4 \times d_e$). Head dimension is fixed at 32. Total parameters are estimated for an input leading to $L = 4096$ tokens after DWT embeddings.

Model Name	Layers (N)	Emb. Dim (d_e)	Heads (h)	FFN Dim (d_{ff})	Total Parameters
WaveLiT-S	4	128	4	512	$\approx$ 1.2M
WaveLiT-B	8	256	8	1024	$\approx$ 9.1M
WaveLiT-L	12	384	12	1536	$\approx$ 30.5M

Performance trade-offs are evaluated on the Navier-Stokes benchmark dataset from PDEArena [1], following the setup reported in [35]. Figure 6 summarizes the mean test relative L_2 error across model sizes and DWT levels. Generally, for a fixed model size, performance improves with fewer DWT levels (longer sequences). This trend suggests that providing more input tokens helps the model capture finer details. However, this is not universally optimal; for datasets with large intrinsic

Table 10: Estimated Forward Pass GFLOPs vs. Number of DWT Levels for a $128 \times 128 \times 3$ input. Sequence length $L = (128/2^{\text{levels}})^2$. Estimates via JAX AOT analysis. Throughput measured on NVIDIA H200 GPU.

Model Variant	**DWT Levels**	**Seq. Length** (L)	**Fwd GFLOPs** (per sample)	**Train Throughput**[*] (samples/sec)
WaveLiT-S	1	4096	0.221	73.1
WaveLiT-S	2	1024	0.055	163.6
WaveLiT-S	3	256	0.014	199.3
WaveLiT-S	4	64	0.004	193.7
WaveLiT-B	1	4096	0.868	15.6
WaveLiT-B	2	1024	0.214	48.7
WaveLiT-B	3	256	0.054	108.7
WaveLiT-B	4	64	0.014	142.2
WaveLiT-L	1	4096	1.941	5.8
WaveLiT-L	2	1024	0.478	19.4
WaveLiT-L	3	256	0.120	46.2
WaveLiT-L	4	64	0.030	48.3

[*]Throughput benchmarked with batch size 8 on a single NVIDIA H200 GPU

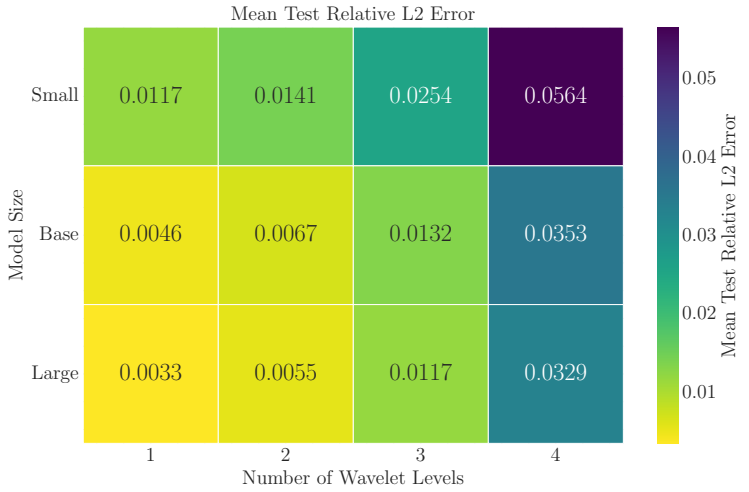

Figure 6: Mean test relative L_2 error (lower is better) vs. model size and DWT levels on the Navier Stokes benchmark from PDEArena[1]. While fewer DWT levels (longer sequences) generally yield lower error, optimal DWT levels depend on the dataset's characteristic scales (cf. Section H). Larger model sizes also tend to reduce error.

correlation lengths, such as HS (see Section H), a moderate number of DWT levels can yield better or comparable performance with significantly reduced computational cost by matching the effective patch size to these characteristic scales. For most other datasets examined, which exhibit finer-scale features, minimizing DWT levels proves beneficial. Similarly, for a fixed number of DWT levels, larger models tend to perform better, although the benefit of increased input resolution often plays a more dominant role in achieving the lowest error.

Discussion. These observations collectively suggest that while model capacity is important, its effective utilization is highly dependent on the quality and resolution of the input sequence. For many PDE problems exhibiting fine-scale details, investing computational resources in processing longer sequences (achieved through fewer DWT levels) appears to be a more efficient strategy for enhancing predictive accuracy than merely increasing model parameters with a low-resolution input. However, as demonstrated by our autocorrelation analysis (Section H) and the HS benchmark results, if the dominant physical phenomena occur at larger scales, matching the DWT levels to these characteristic scales can provide an optimal balance of accuracy and computational efficiency.

G Wavelit Architecture

Wavelet Patch Embedding. Given an input field $u(x, y)$, we apply a l-level DWT using a chosen wavelet basis. The DWT decomposes the input into a hierarchy of approximation and detail coefficients, effectively capturing both coarse structures and fine details. For all experiments, we use the `bior2.2` wavelet [42], which provides a good balance between compactness and smoothness. Our ablation studies (Appendix D) confirm this choice, also showing that comparable results are achievable with alternatives such as the Haar wavelet.

Formally, the DWT operates as follows. For a 2D signal X of shape $H \times W$, we first construct a filter bank from 1D decomposition filters: a low-pass filter L_d for approximation and a high-pass filter H_d for detail extraction. These filters are combined using outer products to create a 2D filter bank:

$$LL = L_d \otimes L_d,\ HH = H_d \otimes H_d,\ LH = L_d \otimes H_d,\ HL = H_d \otimes L_d. \tag{6}$$

Applying this filter bank with a stride of 2 yields four sub-bands, X_{LL}, X_{LH}, X_{HL}, and X_{HH}, each with dimensions $H/2 \times W/2$, forming the first-level transform $X^1 \in \mathbb{R}^{H/2 \times W/2 \times 4}$:

$$X \in \mathbb{R}^{H \times W} \xrightarrow{\text{DWT}} [X * LL, X * LH, X * HL, X * HH] \in \mathbb{R}^{H/2 \times W/2 \times 4}. \tag{7}$$

Each component serves a distinct purpose: X_{LL} provides a coarse approximation, X_{HH} captures high-frequency details, while X_{LH} and X_{HL} represent mixed-frequency components along different spatial directions. For deeper levels of decomposition, we recursively apply the same process to the approximation coefficients X_{LL}, while retaining the detail coefficients from previous levels.

For multi-channel inputs with temporal dimensions of shape $B \times T \times H \times W \times C$ (where B, T, and C represent batch, temporal, and channel dimensions), we apply the transform independently to spatial dimensions, yielding:

$$X \in \mathbb{R}^{B \times T \times H \times W \times C} \xrightarrow{l\text{-level DWT}} X' \in \mathbb{R}^{B \times T \times H/2^l \times W/2^l \times C \cdot 4^l}. \tag{8}$$

Each spatial location in this transformed representation becomes a token processed by our attention mechanism. For implementation, we leverage the efficient `jax-wavelets` package [43].

Mamba-like Linear Attention (MLLA) Block The MLLA block [14] incorporates additional blocks that enhance the performance of a naive-linear attention block through residual connections and Mamba-inspired gating allow the block to have linear complexity $O(N)$ with respect to sequence length N.

The MLLA block starts with depth-wise convolution. Unlike the original ViT, where the inter-token communication is solely done by the self attention mechanism, MLLA block includes Depth-Wise 3×3 Convolution (DWC) before and after the attention layer, promoting localized information exchange. Similar to the self attention, DWC does not mix information channel-wisely. Given an input token grid x of size $H \times W \times C$, the convolutional positional embedding can be expressed as

$$\tilde{x} = x + \text{DWC}(x), \tag{9}$$

$$x_1 = \text{LayerNorm}(\tilde{x}), \tag{10}$$

$$x_g = \sigma(\text{Dense}(x_1)),\ y = \sigma(\text{DWC}(\text{Dense}(x_1))). \tag{11}$$

The linear attention (LA), as the key component of MLLA, can be expressed as

$$\text{LA}(Q, K, V)_i = \frac{\text{RoPE}(\phi(Q_i)) \sum_j \text{RoPE}(\phi(K_i)) V_j}{\text{RoPE}(\phi(Q_i)) \sum_j \text{RoPE}(\phi(K_i))}, \tag{12}$$

where $\text{LA}(Q, K, V)_i$ denotes i-th query result, and K_j, V_j are the j-th key and value, respectively. ϕ is the feature map for the linear kernel attention. We follow the convention of setting $\phi(x) = \text{elu}(x) + 1$, elementwisely, as in [10]. Additionally we apply the rotary positional embedding (RoPE)[44] to the queries and keys after the feature map ϕ. We extend the linear attention to multi-head linear attention as in the multi-head dot product attention, and denote it as MHLA. The self-MHLA can be expressed as

$$z = \text{MHLA}(\text{Dense}(y), \text{Dense}(y), y) + \text{DWC}(y). \tag{13}$$

The self-MHLA is followed by a forget gate x_g, another level of DWC, and a feed-forward layer i.e.,

$$\tilde{z} = \text{Dense}(z \odot x_g) + \tilde{x}, \tag{14}$$

$$z_{\text{out}} = \tilde{z} + \text{Dense}(\sigma(\text{Dense}(\text{LayerNorm}(\tilde{z})))). \tag{15}$$

Table 11: WaveLiT model configurations evaluated. N_L: Number of Layers, D_{model}: Embedding Dimension, N_H: Number of Heads in MLLA. Head dimension for MLLA is set to 32. FFN Dim is $4 \times D_{\text{model}}$.

Model Name	**Layers (N_L)**	**Emb. Dim (D_{model})**	**Heads (N_H)**	**Approx. Parameters**
WaveLiT-S	4	128	4	1.2M
WaveLiT-B	8	256	8	9.1M
WaveLiT-L	12	384	12	30.5M

H Autocorrelation Analysis

The computation costs of any transformer-based model scale proportional to the number of tokens utilized. Consequently, we would like to use *as few tokens as the underlying physics allows*. We can achieve this by roughly matching the patch size to the *characteristic correlation length* ℓ_c of the target physical fields.

Physical intuition. If correlations decay over a short length scale (local regime) then each small region of the domain contains largely independent information and must be resolved individually necessitating many patches. Conversely, when the dominant physics is long-ranged, i.e., $\ell_c \sim \mathcal{O}(L)$, where L is the domain size, neighboring samples are highly redundant: large patches (more wavelet levels) and thus a much shorter input sequence suffice, allowing compute to be spent on model depth or width rather than sequence length.

Two–point autocorrelation. We estimate ℓ_c from the data via the radially averaged two-point autocorrelation function,

$$C(r) = \big\langle C(\mathbf{r}) \big\rangle_{|\mathbf{r}|=r}, \qquad \text{where} \quad C(\mathbf{r}) = \frac{\sum_{\mathbf{x}} \big(u(\mathbf{x}) - \bar{u}\big)\big(u(\mathbf{x}+\mathbf{r}) - \bar{u}\big)}{\sum_{\mathbf{x}} \big(u(\mathbf{x}) - \bar{u}\big)^2}. \tag{16}$$

Here $\bar{u}$ denotes the spatial mean of u.

Definition of the characteristic length. Following standard practices in the turbulence literature [45], we define the correlation length as the first radial distance at which the normalised autocorrelation decays to $1/e$ of its maximum value:

$$\ell_c := \min r \mid C(r) \le e^{-1}, \tag{17}$$

where e is the base of the natural logarithm. For fields that never drop below e^{-1} within the numerical domain, we take ℓ_c to be the largest resolved r. We use the distance between the neighboring pixels as the unit, denoted as px.

Datasets Utilized. From TheWell dataset [24], we pick two representative PDE benchmarks with markedly different correlation scales: `helmholtz_staircase` (HS) exhibits a larger correlation length, whereas `rayleigh_benard` (RB) contains small-scale features and thus a much shorter correlation length (Table 12). Figure 7 showcases the different sample behaviors of the two selected datasets.

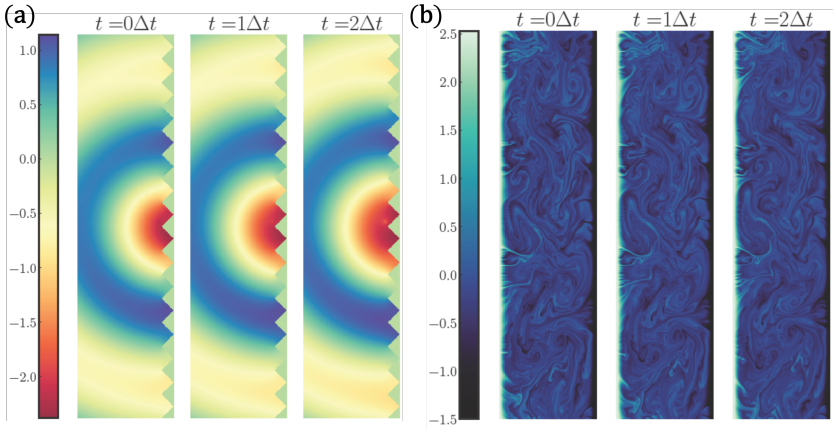

Figure 7: Sample images of the datasets with different characteristic length scale. (a) `helmholtz_staircase`, $\bar{\ell}_c = 59.38$ px. (b) `rayleigh_benard`, $\bar{\ell}_c = 13.08$ px.

Table 12: Correlation lengths (measured in pixels) of the selected data samples (average over 256 random snapshots per dataset).

Dataset	$\bar{\ell}_c$ [px]
`helmholtz_staircase`(HS)	59.38
`rayleigh_benard` (RB)	13.08

Experiments Performed. We vary the number of wavelet levels utilized from $\{1, 2, 3, 4\}$, consequently yielding varying sequence lengths for the WaveLiT-S model ($\approx 9 - 10M$). All other hyper-parameters (optimizer, learning-rate schedule, training steps) are held fixed using the choices we employed during training. The final VRMSE error for each configuration is listed in Table 13.

Table 13: Impact of varying wavelet levels (and thus effective patch size) on model performance (VRMSE). For RB, % change in error is relative to Level 1. For HS, changes are relative to Level 2 (HS's optimal performance).

Dataset	Wavelet Levels	Final Error VRMSE	% Change in Error
`rayleigh_benard` (RB)	1	**0.02605**	—
($\bar{\ell}_c \approx 13$ px)	2	0.03219	$+23.57\%$ (vs L1)
	3	0.04777	$+83.10\%$ (vs L1)
	4	0.07486	$+187.14\%$ (vs L1)
`helmholtz_staircase` (HS)	1	0.01956[†]	$+4152.17\%$ (vs L2[*])
($\bar{\ell}_c \approx 59$ px)	2	**0.00044**	—
	3	0.00055	$+19.57\%$ (vs L2[*])
	4	0.00065	$+41.30\%$ (vs L2[*])

[†]Optimization difficulties observed, likely due to very long sequence length.
[*]HS Level 2 represents the optimal performance (error 0.00046) for this dataset.

Discussion. The results in Table 13 highlight the critical interplay between the field's characteristic correlation length ℓ_c, the effective patch size determined by wavelet levels, and the model's predictive accuracy.

For the `rayleigh_benard` (RB) dataset, characterized by a short correlation length ($\bar{\ell}_c \approx 13$ px), increasing wavelet levels (and thus patch size) consistently degrades performance. Level 1, with the smallest patches, achieves the lowest error (0.02605). Subsequent increases to Levels 2, 3, and 4 result in error increases of approximately 24%, 83%, and 187% respectively, relative to Level 1. This highlights that for fields with fine-scale features, resolving these details with smaller patches is crucial.

In contrast, the `helmholtz_staircase` (HS) dataset, with its larger correlation length ($\bar{\ell}_c \approx 59$ px), exhibits a different behavior. Level 1 performance (error 0.01956) was hampered by optimization difficulties associated with its very long input sequence. A significant improvement occurred at Level 2, where the error plummeted to 0.00046. This suggests that the larger patches at Level 2 adequately capture the long-range correlations of HS, while the shorter sequence length aids optimization. Crucially, while further increasing wavelet levels to Level 3 and Level 4 did lead to a degradation in performance relative to this optimal Level 2 result, this fall-off in accuracy is notably less sharp than that observed for the RB dataset. Figure 8 visually confirms that the rate of performance degradation for HS beyond its optimum is notably less severe than for RB. This indicates that while patches can become too coarse even for HS, its inherent long-range correlations make it more robust to increases in patch size beyond the optimum, compared to the RB dataset whose small-scale features are more quickly lost.

These findings confirm our hypothesis: matching the effective patch size with the field's ℓ_c is crucial. Short ℓ_c fields (RB) demand fine resolution (fewer wavelet levels) and show a rapid performance decline with coarser patches. Long ℓ_c fields (HS) are more robust to coarser representations (more wavelet levels) with accuracy declining more gracefully.

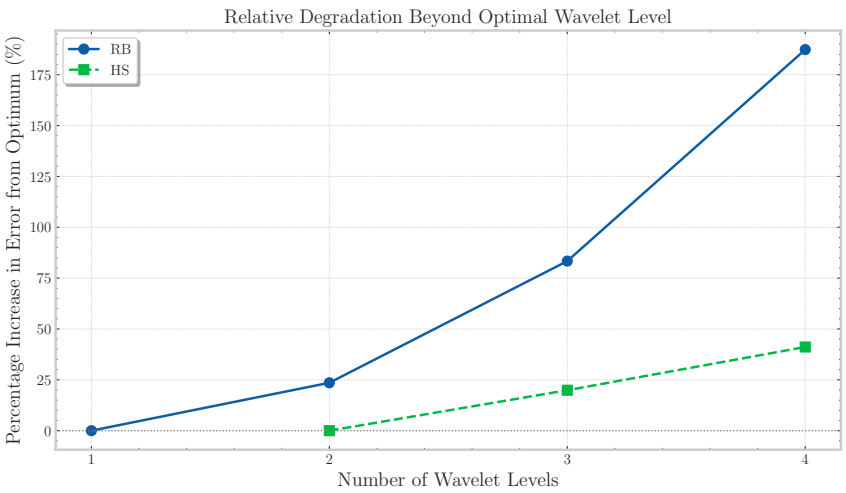

Figure 8: Percentage increase in model error from the dataset-specific optimal wavelet level.

Looking ahead, these insights into the importance of local interactions for short ℓ_c fields suggest promising avenues for future investigation. Exploring the integration of attention mechanisms designed for locality, such as sparse attention [46] or windowed attention schemes like those found in Swin Transformers [47], could offer further enhancements in computational efficiency and performance, particularly for systems dominated by fine-scale, local phenomena.