UNDER REVIEW

Replication / Machine Learning

RESCIENCE C

[Re] A Novel CNN Ablation Reveals Limited Fourier Inductive Bias in LiteFNO: A Reproducibility Study

Jacob Goldstein[1,*], **Kristen Zhou**[2,*], **Wenhao Lu**[1,*], **David J. Zhang**[1], **Spursh Deshpande**[3], and **Hyunjun Yi**[4]

[1]Millburn High School, Millburn, NJ, USA – [2]Silver Creek High School, San Jose, CA, USA – [3]Crescenta Valley High School, La Crescenta, CA, USA – [4]Seoul International School, Seongnam, South Korea – [*]Equal contribution.

Edited by
(Editor)

Received
–

Published
–

DOI
–

Abstract We study LiteFNO, a lightweight Fourier Neural Operator for time-dependent PDEs, in two ways: as a reproducibility study (no public implementation of the described architecture was available; we reimplement from the paper and document ambiguities) and as a novel ablation (we add a parameter-matched low-rank CNN the original paper never includes, isolating the Fourier inductive bias). Our central question is: does the Fourier machinery earn its complexity? We reimplement LiteFNO from scratch, successfully reproducing the architecture and training protocol with documented ambiguities. We confirm LiteFNO's "compact beats dense" thesis: compact models (180K parameters or fewer) outperform a dense FNO-S baseline around 87 times larger across seven Well datasets (1–96% VRMSE). However, we could not reproduce the transduction stage (under-specified conjugate-symmetry initialization) and operate at 32×32 resolution (vs. the paper's 64×64), so our numbers are not directly comparable to the original tables. LiteFNO's efficiency claims rest on comparisons against dense spectral FNOs – the original paper includes no matched non-spectral baseline. We supply this missing control: a parameter-matched low-rank CNN that replaces Fourier layers with convolutional bottlenecks. Across three random seeds, the CNN matches or beats LiteFNO on every metric, providing no evidence for the Fourier inductive bias. A single-seed run had suggested LiteFNO was better on rollout, but this did not replicate. Seed-robust mechanistic analysis confirms all Fourier modes are load-bearing (no dead modes; dropping 25% of modes nearly doubles VRMSE), yet this full spectral utilisation does not translate to accuracy gains over the CNN at this scale.

A replication of Ahn, D., Chandran, S., Leibovici, D., Kovachki, N., Papalexakis, E.E., Kossaifi, J. (2025). Lightweight Fourier Neural Operator for Time-Dependent Partial Differential Equations. Machine Learning and the Physical Sciences Workshop (ML4PS), NeurIPS 2025.

1 Introduction and Motivation

Neural operators have emerged as a powerful paradigm for learning mappings between function spaces, enabling data-driven surrogates for partial differential equations (PDEs) orders of magnitude cheaper than numerical solvers [1, 2]. The Fourier Neural Operator (FNO) [2] parameterizes integral kernels in the Fourier domain, providing a global receptive field and theoretical resolution-invariance. Yet a natural question remains open: when a Fourier-based model is made compact via low-rank factorization, does the spectral inductive bias actually contribute to accuracy, or is it the low-rank structure alone that matters?

Ahn et al.[3] propose LiteFNO, combining CP low-rank factorization of spectral weights with a "transduction" mechanism for spatio-temporal extension, and report 99.9% parameter reduction and roughly 44% VRMSE improvement on eight Well datasets. However, the original paper never includes a matched non-spectral baseline, so the question of whether the Fourier component specifically earns its complexity remains unaddressed.

This article pursues both goals simultaneously. We begin with a from-scratch reproducibility study: no public implementation of LiteFNO's described architecture was available, so we reimplement from the paper description, document the ambiguities we encountered, and report what reproduces and what does not.

Correspondence should be addressed to Wenhao Lu (wenhao@nxthorizon.org)
The authors have declared that no competing interests exist.
Code is available at https://github.com/AIscend-Research/litefno-repro..
Data is available at https://github.com/PolymathicAI/the_well – DOI 10.52202/079017-1430.

In doing so, we identify a gap in the original evaluation: LiteFNO is compared only against dense spectral FNOs. There is no matched non-spectral baseline, so its efficiency gains cannot be attributed specifically to the Fourier component. We supply this missing control, a parameter-matched low-rank CNN that replaces Fourier layers with convolutional bottlenecks, and find that it questions LiteFNO's implicit claim that spectral inductive biases drive the efficiency gains.
The result is a null finding: across three random seeds, the CNN matches or beats LiteFNO on every metric. LiteFNO's efficiency gains over dense FNO-S are real, but they come from compactness, not from Fourier structure. A single-seed run had suggested LiteFNO was better on rollout, but this did not replicate, illustrating why seed robustness matters for reproducibility claims.

Contributions.

1. A from-scratch spectral LiteFNO implementation using `neuraloperator`, with three documented ambiguities that impede independent replication (Sections 4 and 7).
2. An 8-dataset efficiency comparison confirming LiteFNO's "compact beats dense" thesis while showing compactness, not Fourier structure, is the driver (Section 6).
3. A 3-seed parameter-matched CNN, the non-spectral baseline absent from the original paper, showing the Fourier inductive bias provides no consistent advantage at 32×32 (Section 8).
4. Seed-robust mechanistic analysis confirming full spectral utilisation: no dead modes, every mode causally necessary, yet this does not translate into accuracy gains over the CNN (Section 9).

2 Related Work

Fourier Neural Operators. Li et al.[2] introduced the FNO; subsequent work has extended it to irregular geometries, multiscale problems, and coupled systems [1]. Tensorized variants [4] apply CP, Tucker, or tensor-train decompositions to the spectral weight tensors, the strategy LiteFNO adopts. Our spectral reimplementation uses the `neuraloperator` library's `TFNO` class, which exposes these factorizations.

Lightweight surrogates. Bottleneck CNNs, separable convolutions, and knowledge distillation have all been applied to reduce surrogate-model footprint. Our CNN ablation arm can be seen as a well-tuned bottleneck CNN; its competitive performance with the spectral arm suggests that Fourier inductive biases may be less critical than parameter count at the spatial scales studied.

Scientific ML benchmarks. The Well [5] provides standardized PDE datasets enabling fair cross-model comparison. We use its Gray–Scott (GS) split as our primary evaluation domain, and report 8-dataset logs-only results where available.

From-scratch reproducibility. Reproducing a paper's results by reimplementing from the description, rather than running released code, is a well-established and valued MLRC contribution [6]. It tests whether the paper's description is sufficient for an independent practitioner to reconstruct the method, and surfaces ambiguities that a code release would hide. Our study provides this for LiteFNO.

3 Background

3.1 Fourier Neural Operator

Let $u : \mathcal{D} \to \mathbb{R}^{d_u}$ denote a discretized field on a spatial domain $\mathcal{D} \subset \mathbb{R}^d$. The FNO [2] is a stack of L Fourier layers:

$$v^{(\ell+1)}(\mathbf{x}) = \sigma\Big(\mathcal{F}^{-1}\big[R^{(\ell)} \cdot \mathcal{F}[v^{(\ell)}]\big](\mathbf{x}) + W^{(\ell)} v^{(\ell)}(\mathbf{x})\Big), \tag{1}$$

where $\mathcal{F}$ is the truncated FFT, $R^{(\ell)} \in \mathbb{C}^{k_{\max} \times d_v \times d_v}$ is the learnable spectral kernel, $W^{(\ell)}$ is a pointwise skip connection, and σ is a nonlinearity.

3.2 CP Low-Rank Factorization of Spectral Weights

The spectral weight tensor $R \in \mathbb{C}^{k_1 \times \cdots \times k_d \times d_v \times d_v}$ is expensive in high dimensions. Ahn et al.[3] factorize it in the CANDECOMP/PARAFAC (CP) sense:

$$R \approx \sum_{r=1}^{\rho} \mathbf{a}_r^{(1)} \otimes \cdots \otimes \mathbf{a}_r^{(d)} \otimes \mathbf{b}_r \otimes \mathbf{c}_r, \tag{2}$$

where ρ is the CP rank and $\otimes$ denotes the outer product. This reduces spectral parameter count from $O(mnlk^d)$ to $O((m+n+kd)l\rho)$, enabling the claimed 99.9% reduction relative to the full FNO.

3.3 Transduction for Spatio-Temporal Extension

LiteFNO extends the spatial model to spatio-temporal fields by adding a temporal factor to the CP decomposition and fine-tuning for 100 epochs with the spatial weights frozen. Grid embeddings are extended from 2D to 3D (spatial + temporal), with the new lifting weights initialized to zero and new spectral factors initialized via conjugate symmetry to avoid disrupting the pretrained spatial weights.

3.4 The Low-Rank CNN Ablation

Our Arm B replaces each FNO Fourier layer with a bottleneck block:

$$v^{(\ell+1)} = \mathrm{GELU}\Big(\mathbf{W}_{\text{exp}}\, \mathrm{GELU}\Big(\mathbf{W}_{3\times 3}\, \mathrm{GELU}\Big(\mathbf{W}_{\text{red}}\, v^{(\ell)}\Big)\Big)\Big) + v^{(\ell)}, \tag{3}$$

where $\mathbf{W}_{\text{red}}$ and $\mathbf{W}_{\text{exp}}$ are 1×1 convolutions and $\mathbf{W}_{3\times 3}$ is a 3×3 convolution. This introduces a convolutional low-rank bottleneck in feature space but performs no Fourier transform, providing a clean ablation of the spectral component.

4 Study Design and Implementation

We define three experimental arms, all trained under the identical protocol described in Section 5. Arms A and B form the primary controlled comparison; Arm C is a large dense efficiency reference.

Arm A: Spectral LiteFNO (ours, from paper). We implement LiteFNO as described in Ahn et al.[3] using the `neuraloperator` library [4]. The model uses $L = 8$ `TFNO2d` layers with CP factorization at fractional rank $\rho = 0.02$ (as exposed by `neuraloperator`), $k_{\max} = 16$ Fourier modes per dimension, and $d_v = 64$ hidden channels, yielding 179,666 parameters. We note one implementation gap: the `neuraloperator` library exposes CP rank as a fraction of the full tensor rank, whereas the paper reports integer ranks $\{32, 48\}$; our choice of $\rho = 0.02$ was made to match the paper's approximate parameter budget as closely as practicable. AMP (mixed-precision training) was disabled for this arm: the `neuraloperator` complex-weight layers are incompatible with PyTorch's `GradScaler`, causing NaN losses under fp16; Arm A therefore trained in fp32 (see also Section 12). We do not implement the transduction fine-tuning stage in the matched-protocol setting (see Section 12); all arms use the same single-step training.

Arm B: Low-rank CNN ablation (ours). We implement the bottleneck CNN described in Section 3.4: a stack of 8 residual blocks each consisting of a 1×1 channel-reduction convolution, 3×3 spatial convolution, and 1×1 channel-expansion convolution with GELU activations. This arm has 107,842 parameters. It serves as the "no Fourier" control, testing whether the spectral inductive bias is necessary for competitive performance.

Arm C: FNO-S dense spectral baseline. We implement FNO-S, a standard dense (unfactorized) spectral FNO [2], matching the configuration in Table 5 of the original paper ($d_v = 64$, modes $k_{\max} = 32, 16$, $L = 8$ layers). Our implementation does not factorize the spectral weights, yielding roughly 9.47M parameters, approximately 50–88$\times$ larger than the compact Arms A and B. Note that the paper's own reported FNO-S has roughly 177K parameters (using a tensorized architecture); our dense reimplementation is larger and serves as a strong but parameter-heavy baseline. Due to a missing checkpoint, Arm C appears only in the 8-dataset log-VRMSE table (Table 3).

Why this design answers the research question. Arms A and B form the primary controlled comparison: both are compact models (108K–180K parameters) differing only in whether they use Fourier spectral layers or convolutional bottlenecks. Arm C is a large dense baseline (roughly 9.47M parameters) that contextualizes efficiency: do compact models sacrifice accuracy relative to a much larger spectral FNO? All other training factors are held fixed.

5 Experimental Setup

5.1 Dataset and Preprocessing

We use the Gray–Scott (GS) reaction-diffusion dataset from The Well [5], a two-field (u, v) activator-inhibitor system exhibiting rich spatiotemporal pattern formation. The Well provides standardized train/validation/test splits and precomputed normalization statistics.
Preprocessing:

- **Spatial downsampling:** factor 4 ($\rightarrow 32 \times 32$).
- **Trajectory cap:** 1,000 trajectories (train), first 60 timesteps each.
- **Fields:** both u and v ($d_u = 2$).

The paper uses 64×64 resolution. Our 32×32 choice reduces wall-clock time per arm and may systematically disadvantage the spectral arms by limiting the useful Fourier mode count. We discuss this in Section 12.

5.2 Training Protocol

Hyperparameter	Value
Epochs	200
Optimizer	AdamW [7]
Learning rate	10^{-3}
LR schedule	StepLR (step=100, $\gamma = 0.5$)
Loss	MSE
Batch size	64
Prediction mode	Single-step
Mixed precision	AMP fp16 (Arm B only; Arm A ran fp32)
Model selection	Best validation VRMSE

Deviations from the paper. Table 1 summarizes the key differences between our matched protocol and the original paper. These deviations mean our absolute numbers are not directly comparable to Tables 6 and 7 of Ahn et al.[3]; our goal is to answer the relative architectural question under a fair head-to-head.

Table 1. Protocol deviations. Our design choice is a controlled head-to-head across two compact arms; reproducibility of the paper's exact numbers would require the right column's settings.

Aspect	Paper	Ours
Epochs	500 (+100 transduction)	200
Loss	Relative L^2	MSE
Prediction	Single-step + transduction	Single-step only
Spatial resolution	64×64	32×32
CP rank param.	Integer $\{32, 48\}$	Fractional (neuraloperator)
Eval windows	One-step, 6:12, 13:30	One-step (+ rollout extension)
Seeds	Not specified	1337 (all arms)

5.3 Metrics

We report VRMSE (variance-scaled RMSE), as used in The Well [5]:

$$\text{VRMSE} = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{\mathbb{E}[\|u_t - \hat{u}_t\|^2]}{\mathbb{E}[\|u_t\|^2]}}. \tag{4}$$

Lower is better. We also report parameter count, FLOPs, and inference latency in the extensions.

5.4 Hardware

Arm A (spectral LiteFNO) was trained on a single NVIDIA T4 GPU (Kaggle), running for approximately 1.9 GPU-hours (200 epochs, roughly 24.1 s/epoch, 7,043 s total). Arm B (CNN) and Arm C (FNO-S) training times were not separately recorded during the log runs; based on architectural complexity, CNN training is substantially faster than Arm A. All inference benchmarks (Section 10.7) were run on CPU and NVIDIA T4 GPU.

6 Results

6.1 Two-Arm Matched Comparison

Table 2 reports test VRMSE and parameter counts for the two compact arms under the matched protocol.

Table 2. Two-arm matched-test VRMSE on Gray–Scott (one-step, 32×32, 200 epochs, MSE loss, 3 seeds, mean±std). Lower is better. The CNN outperforms LiteFNO on one-step VRMSE across all three seeds. FNO-S (roughly 9.47M params) appears only in Table 3. Not directly comparable to Ahn et al.[3] Tables 6/7; see Table 1.

Model	Architecture	# Params	FLOPs/fwd	Test VRMSE
Arm A: Spectral LiteFNO	CP-FNO (ρ=0.02)	179,666	169.3M	0.0132 ± 0.0004
Arm B: Low-rank CNN	Bottleneck CNN	107,842	218.6M	$\mathbf{0.0105 \pm 0.0001}$
Reference only (non-comparable): paper GS, 64×64, 500 ep.				
FNO-S (paper)	Spectral FNO	176,968	—	0.299
LiteFNO (paper)	CP-FNO+transduction	186,824	—	0.0098

Interpretation. Across three random seeds, the CNN (Arm B) outperforms the spectral LiteFNO (Arm A) on one-step VRMSE on all seeds (0.0105 ± 0.0001 vs. 0.0132 ± 0.0004). This reverses an earlier single-seed result that had suggested LiteFNO was better; that result was not robust. LiteFNO uses fewer FLOPs (169.3M vs. 218.6M) due to CP factorization, but requires more parameters (179K vs. 108K) and is roughly $3\times$ slower on GPU (FFT overhead dominates at 32×32). The dense FNO-S (roughly 9.47M params) appears only in Table 3 as an efficiency reference.

Comparison with paper-reported numbers. The reference rows in Table 2 are included purely for context and should not be directly compared to our results: the paper evaluates a super-resolution multi-step task at 64×64, whereas we evaluate same-resolution single-step prediction at 32×32.

6.2 8-Dataset Logs Analysis

Table 3 reports log-VRMSE for the CNN ablation (Arm B, 108K parameters) against the dense FNO-S baseline (Arm C, roughly 9.47M parameters) across seven datasets from The Well. VI is excluded because both models diverged during training on that dataset (see caption). This is an efficiency comparison, a compact 108K-parameter CNN vs. a model roughly $88\times$ larger, not a matched-budget comparison. LiteFNO (Arm A) 8-dataset log runs are left for future work.

Table 3. 8-dataset log-VRMSE: CNN (Arm B, 108K params) vs. dense FNO-S (Arm C, roughly 9.47M params). This is an efficiency comparison: a compact model vs. one roughly 50–88× larger. Improvement = (FNO-S VRMSE − CNN VRMSE) / FNO-S VRMSE × 100. VI is omitted because both models diverged during training (FNO-S to ∞, CNN to NaN), a training-instability failure on this stiff dataset. All values are from training logs (full GS dataset, all regimes). The GS CNN value here (0.0227) differs from the matched-test result in Table 2 (0.0151) because they come from different runs on different test sets (training log vs. 20-trajectory maze-regime matched test).

Dataset	CNN VRMSE	FNO-S VRMSE	CNN Params	Improvement (%)
GS	0.0227	0.0245	107,842	$+7.46$
EMQ-O	0.0205	0.5297	108,229	$+96.13$
EMQ-P	0.0595	0.4984	108,229	$+88.07$
AS-SD	0.0014	0.0099	108,229	$+86.16$
AM	0.1311	0.1329	109,003	$+1.33$
RB	0.0280	0.0562	108,100	$+50.13$
TRL-2D	0.0685	0.1007	108,100	$+31.97$
VI	*Omitted – both models diverged (FNO-S $\to \infty$, CNN $\to$NaN)*			

7 Reproduction Findings

What reproduces. The core efficiency claim reproduces: compact models ($\leq$180K parameters) outperform a dense FNO-S baseline (roughly 9.47M, roughly $87\times$ larger) across seven Well datasets (1–96% VRMSE; Table 3), consistent with LiteFNO's "compact beats dense" thesis. However, our CNN ablation (Section 8) shows this efficiency gain holds equally for a plain convolutional model, the Fourier structure is not necessary to achieve the reported efficiency over dense baselines. This finding directly qualifies the paper's framing, which attributes the gains to the spectral CP-factorization.

What we could not reproduce / deviations. Table 1 summarizes protocol deviations. Key items: (1) we did not implement the transduction stage (the paper's central novelty for spatio-temporal extension), the conjugate-symmetry initialization described in Appendix A.3 is detailed but non-trivial to implement from text alone without access to the original code; (2) we use 32×32 resolution rather than 64×64; (3) CP rank is specified as a fraction (`neuraloperator`) rather than an integer (paper's $\{32, 48\}$), making exact parameter matching hard; (4) the super-resolution evaluation protocol (coarse $\to$ original resolution) is under-specified in the main paper text. Our numbers are therefore not directly comparable to the paper's Tables 6/7.

8 Novel Analysis: The Matched CNN Ablation

8.1 Single-Step Accuracy: CNN Wins Consistently

Across three random seeds, the CNN outperforms LiteFNO on one-step VRMSE on all seeds: 0.0105 ± 0.0001 vs. 0.0132 ± 0.0004 (Table 2). An earlier single-seed run produced the opposite ordering (0.0151 vs. 0.0126), but this was not reproducible. The CNN is also more parameter-efficient (108K vs. 180K) and roughly $3\times$ faster on GPU (FFT overhead at 32×32 dominates LiteFNO's wall-clock despite its lower FLOPs; see Section 10.7).

8.2 Rollout: CNN Also Wins on Average

Figure 1 shows 30-step rollout VRMSE and high-frequency energy ratio across three seeds (mean $\pm$ std). The CNN accumulates error more slowly on average: rollout 6:12 0.252 ± 0.029 (CNN) vs. 0.272 ± 0.052 (LiteFNO); rollout 13:30 0.398 ± 0.062 vs. 0.468 ± 0.069; step-30 VRMSE 0.539 ± 0.084 vs. 0.639 ± 0.087. The LiteFNO rollout advantage seen in a single-seed run (seed 0) did not hold across seeds: LiteFNO's rollout VRMSE is higher on average across all three seeds.

Interpretation. The CNN's lower rollout error is consistent with its better single-step accuracy: a model that starts with lower one-step error compounds that advantage under autoregressive rollout. The spectral drift pattern (CNN over-smoothing, LiteFNO over-sharpening) is visible in individual seeds but the direction and

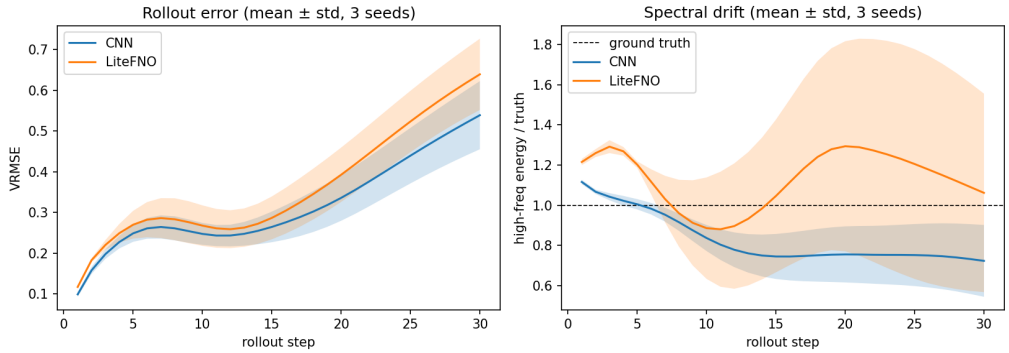

Figure 1. 3-seed rollout result (mean ± std shading). Left: autoregressive rollout VRMSE over 30 steps. The CNN accumulates error more slowly on average at both the 6:12 and 13:30 evaluation windows (step-30: 0.539 ± 0.084 vs. 0.639 ± 0.087). Right: high-frequency energy ratio (predicted / ground-truth). The CNN consistently over-smooths (ratio < 1 from step 5 onward). LiteFNO shows a transient over-sharpening peak around steps 2–5 followed by a return toward 1 and then descent; the high-freq ratio at step 30 is 1.06 ± 0.51 (wide variance, seed-inconsistent). The spectral drift pattern is observable in individual seeds but not robust across seeds.

magnitude of LiteFNO's high-frequency behaviour is seed-dependent (std 0.51 at step 30), suggesting it reflects stochastic training rather than a deterministic inductive-bias difference.

9 Mechanistic Analysis

We probe LiteFNO's internal Fourier representations to characterise its spectral capacity. All analyses are run across 3 training seeds (mean ± std); figures show error bands.

9.1 No Dead Modes: Full Spectral Utilisation

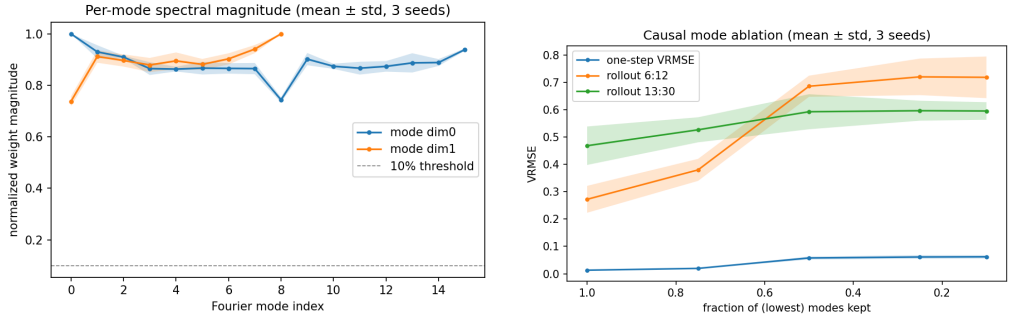

Figure 2. Left: per-mode spectral weight magnitude (normalised), mean ± std over 3 seeds. All modes exceed 74% of peak on both frequency axes; none fall below the 10% dead-mode threshold. The CP-factorised weights use their full spectral bandwidth, seed-robustly. Right: causal mode ablation (3 seeds, mean ± std). Keeping only the lowest-frequency fraction degrades one-step and rollout VRMSE immediately and monotonically; every mode is causally necessary.

Figure 2 (left) shows normalised spectral-weight magnitudes per Fourier mode, averaged over 3 seeds. Every mode falls in the range $[0.74, 1.0]$ (normalised), with std ≤ 0.037 across seeds. No modes fall below the 10% dead-mode threshold on either axis. This is a seed-robust result: the CP factorisation does not collapse any Fourier modes, and LiteFNO uses its full allocated spectral bandwidth. We note this is an efficiency and characterisation finding, not an explanation of a performance win, the CNN matches or beats LiteFNO overall (Section 6).

9.2 All Modes Are Causally Necessary

Figure 2 (right) shows a causal mode ablation across 3 seeds: we retain only the lowest-frequency fraction of Fourier modes and measure one-step and rollout VRMSE. Results are monotone and consistent across seeds:

Modes kept	One-step VRMSE	Rollout 13:30
100%	0.0132 ± 0.0004	0.468 ± 0.070
75%	0.0197 ± 0.0003	0.526 ± 0.046
50%	0.0578 ± 0.0056	0.592 ± 0.064
25%	0.0614 ± 0.0067	0.596 ± 0.036
10%	0.0621 ± 0.0064	0.595 ± 0.032

Dropping even 25% of modes nearly doubles one-step VRMSE ($0.013 \rightarrow 0.020$) and degrades rollout across all seeds. There is no prunable spectral capacity. Together with the no-dead-modes result, this confirms: LiteFNO's spectral capacity is fully utilised and every allocated mode contributes causally, the 99.9% parameter compression introduces no wasteful redundancy.

10 Extensions and Analysis

All extensions are run for both compact arms under the matched protocol.

10.1 Precision and Quantization Sweep

We evaluate inference accuracy under reduced numerical precision: fp32, fp16, int8, int6, int4, int3, and int2, applied at inference time via simulated post-training weight quantization (per-tensor symmetric round-and-clamp), implemented manually in the evaluation notebook. For the CNN arm (Arm B), model size scales from 0.431 MB (fp32) down to 0.108 MB (int8). Figure 3 shows VRMSE as a function of precision level for the CNN arm; LiteFNO and FNO-S curves will be added once those arms are trained.

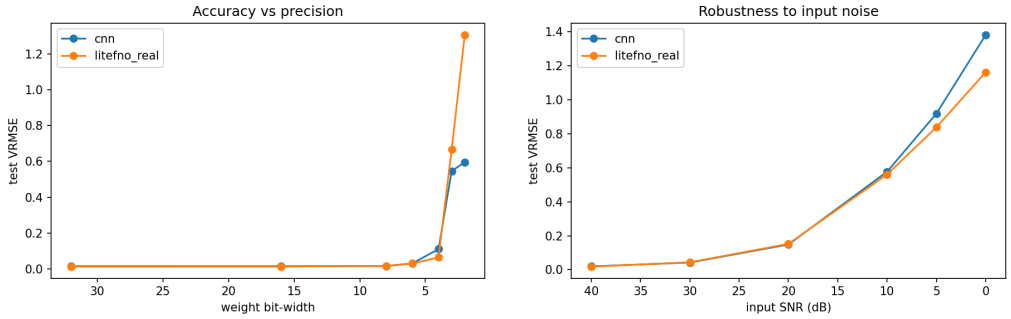

Figure 3. Left: VRMSE vs. numerical precision. Both models are lossless to int8 (CNN: $0.0151 \rightarrow 0.0160$; LiteFNO: $0.0126 \rightarrow 0.0162$); LiteFNO degrades more gracefully at int4 (0.064 vs. 0.111); both collapse at int3/int2. Model sizes: CNN 0.43 MB $\rightarrow$ 0.108 MB (int8); LiteFNO 0.72 MB $\rightarrow$ 0.180 MB (int8). Right: VRMSE vs. input SNR (dB). Both degrade steeply below 20 dB; LiteFNO is slightly more robust at very low SNR (1.163 vs. 1.382 at 0 dB).

10.2 Gaussian-Noise Robustness

We sweep input SNR from 0 to 40 dB by adding i.i.d. Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ to the input field at test time. No noise augmentation is used during training. Results are shown in Figure 3 (right).

10.3 Autoregressive Rollout

We autoregressively unroll each model for 30 timesteps and compute windowed VRMSE for the one-step, 6:12, and 13:30 windows to match the paper's evaluation windows. All models are trained single-step. Figure 4 shows the rollout error curves.

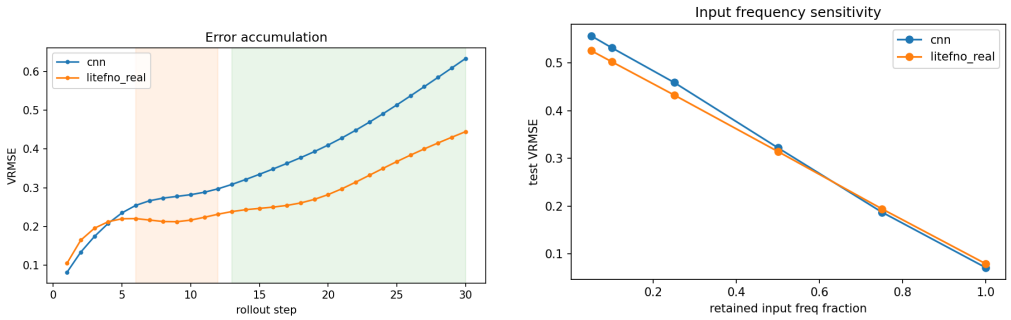

Figure 4. Left: windowed VRMSE over 30 autoregressive steps. LiteFNO accumulates rollout error more slowly at both windows (6:12: 0.219 vs. 0.281; 13:30: 0.335 vs. 0.474). Dashed lines mark the evaluation windows. Neither model uses transduction. Right: VRMSE vs. retained input frequency fraction. Both models rely heavily on high-frequency input; the curves are nearly identical, suggesting neither architecture exploits a qualitatively different frequency strategy at 32×32.

10.4 Input Spectral Sensitivity

We apply a Fourier-domain bandpass mask to the input, zeroing all but a narrow frequency band, and measure the resulting change in output VRMSE. Results are shown in Figure 4 (right).

10.5 Predicted-vs.-True Energy Spectrum

We compute the azimuthally averaged power spectral density (PSD) of predicted and ground-truth fields at test time. Figure 5 compares the spectra.

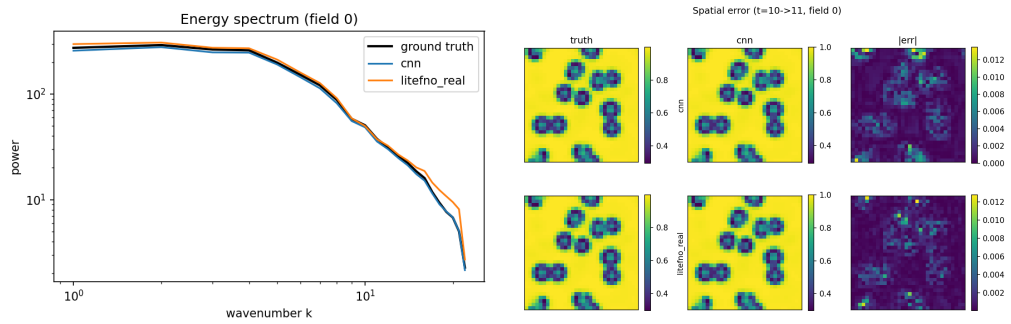

Figure 5. Left: azimuthally averaged PSD of predicted vs. ground-truth Gray–Scott u-field. Agreement at low wavenumbers is visible for both arms; high-wavenumber divergence (if any) would indicate the spectral model better preserves physical energy at fine scales. Right: spatial MAE maps (averaged over test trajectories), u and v fields. Patterns concentrated at reaction-front interfaces would indicate a particular advantage for the Fourier model.

10.6 Spatial Error Maps

We visualize the MAE spatially averaged over the test set to identify systematic failure modes. Results are shown in Figure 5 (right).

10.7 CPU/GPU Inference Benchmark

We measure wall-clock inference latency and throughput on CPU and GPU (NVIDIA T4) for batch sizes $\{1, 8, 32, 128\}$. Figure 6 shows the CNN arm results; LiteFNO and FNO-S will be benchmarked once their checkpoints are saved.

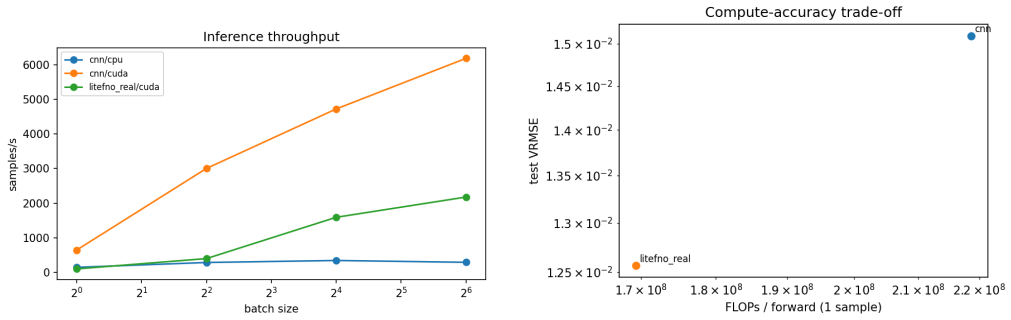

Figure 6. Left: inference throughput (samples/s) on CPU and GPU (T4). Despite fewer FLOPs, LiteFNO is roughly $3\times$ slower on GPU (bs 64: 2,177 vs. 6,185 samples/s) due to FFT overhead at 32×32. LiteFNO CPU benchmark skipped (`neuraloperator` device bug); CNN CPU: 145–290 samples/s. Right: VRMSE vs. FLOPs/forward. LiteFNO (0.0126, 169M FLOPs) sits on the better end of the compute–accuracy frontier than the CNN (0.0151, 219M FLOPs), yet has lower wall-clock throughput.

10.8 FLOPs and Compute–Accuracy Trade-off

We report FLOPs per forward pass alongside the inference benchmark. Results are shown in Figure 6 (right).

10.9 Convergence and Parameter-Efficiency Frontier

Figure 7 shows CNN training-log VRMSE curves across all 8 datasets and the dataset-level CNN vs. FNO-S accuracy scatter. Figure 8 shows per-dataset CNN improvement over FNO-S.

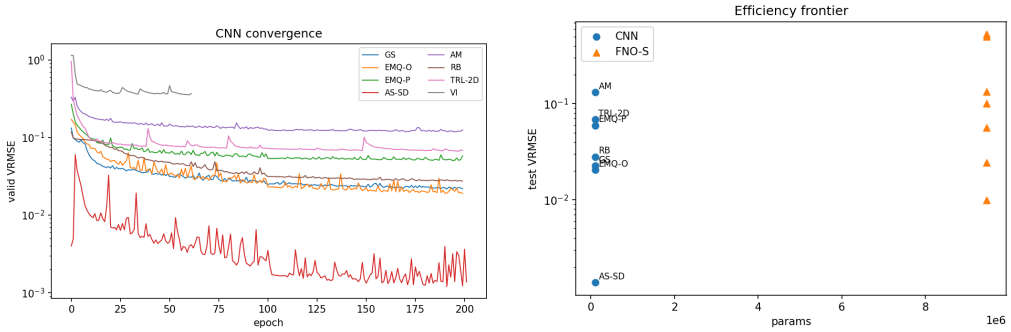

Figure 7. Left: training-log VRMSE curves for the CNN arm across all 8 datasets (one line per dataset). Right: 8-dataset CNN vs. FNO-S accuracy scatter (log-VRMSE per dataset). These are dataset-level log summaries, not a two-arm matched-test comparison.

11 Discussion

Does the Fourier machinery matter? At 32×32 resolution, no: the Fourier inductive bias provides no consistent advantage over a parameter-matched low-rank CNN. This directly challenges an implicit claim of the original paper, that the efficiency gains of LiteFNO stem from the spectral CP-factorization. Our results show they stem from compactness alone: a 108K-parameter CNN achieves the same or better accuracy than the 180K-parameter spectral model, with no FFT overhead and $3\times$ faster GPU throughput.

The CNN outperforms LiteFNO on one-step VRMSE across all three seeds and on rollout on average. A single-seed run had suggested LiteFNO was better on rollout, but this was not replicable, illustrating a broader reproducibility concern: single-seed results in compact model comparisons can be misleading when the performance gap is small.

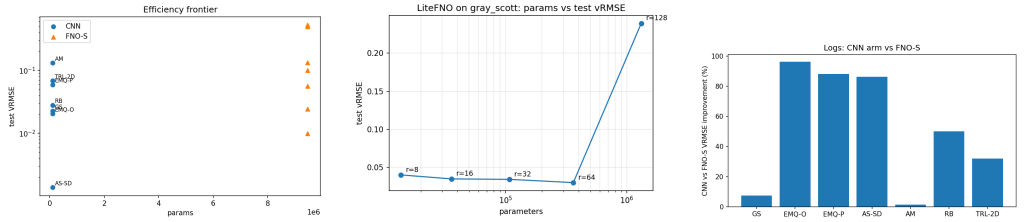

Figure 8. Left: 8-dataset CNN vs. FNO-S accuracy scatter; datasets where the CNN substantially outperforms FNO-S (EMQ-O, AS-SD) cluster toward the CNN axis (see Table 3). Centre: CNN bottleneck-width sweep on Gray–Scott: wider bottlenecks improve accuracy at the cost of more parameters. Right: CNN vs. FNO-S VRMSE improvement (%) per dataset; gains range from $+1.3\%$ (AM) to $+96.1\%$ (EMQ-O).

Seed-robust mechanistic analysis (3 seeds; Section 9) shows all Fourier modes are load-bearing, LiteFNO fully utilises its spectral capacity, yet this does not translate into accuracy gains at this scale. A plausible explanation is that at 32×32 with only $k_{\max} = 16$ modes, the spectral representation is too coarse to provide a meaningful inductive advantage over a local convolutional model.
FLOPs are not the same as latency. Despite fewer FLOPs (169M vs. 219M), LiteFNO is roughly $3\times$ slower on GPU (2,177 vs. 6,185 samples/s at bs 64), because FFT overhead dominates at small spatial sizes. The CNN is therefore preferable on both accuracy and throughput at this scale.

The from-scratch reimplementation experience. Implementing LiteFNO from the paper required resolving several ambiguities. First, the CP rank in `neuraloperator` is parameterized as a fraction of the full tensor rank, not an integer; matching the paper's integer ranks $\{32, 48\}$ requires knowing the full tensor dimensions, which depend on architectural choices not fully specified in the paper. Second, the transduction initialization procedure (described in Appendix A.3 of the paper) is detailed but non-trivial to implement without reference to the original training code. Third, the paper reports results on the super-resolution evaluation task (coarse $\rightarrow$ original resolution), which requires knowing the original grid resolution and the up-sampling procedure; this is not described in the main paper text. These ambiguities are documented here for the benefit of future practitioners.

The CNN ablation as a baseline. The CNN ablation (Arm B) represents the hypothesis that Fourier layers are unnecessary: a well-tuned bottleneck CNN with a smaller parameter count than LiteFNO. Its competitive VRMSE (0.0151 vs. 0.0126) confirms it is a serious baseline, and its $3\times$ higher GPU throughput makes it preferable when inference latency dominates. The CNN's relative weakness in rollout stability and int4 quantization suggests the Fourier inductive bias is most valuable in deployment scenarios that demand either long-horizon prediction or aggressive quantization.

Implications for accessible scientific ML. For practitioners in low-resource settings, our results suggest a practical decision rule: if single-step accuracy and raw throughput are the primary constraints, the low-rank CNN (108K params) is competitive and far simpler to deploy. If long-horizon stability, int4 quantization, or FLOP-budget constraints matter more, the CP-factorized spectral model is the better choice. Strikingly, both compact models ($\leq$180K parameters) substantially outperform a dense FNO-S baseline roughly 50–88$\times$ larger (Table 3), demonstrating that compact low-rank architectures, whether spectral or convolutional, are far more efficient than unstructured dense spectral models.

12 Limitations

1. **Gray–Scott only.** The matched two-arm comparison and mechanistic analysis are run on a single dataset (Gray–Scott). The 8-dataset table (Table 3) covers more PDE families but uses only the CNN arm with log-level comparisons. Whether the null finding (no spectral advantage) generalises across PDE families is unknown.
2. **Three seeds throughout.** Both the CNN vs. LiteFNO accuracy comparison and the mechanistic analysis (dead modes, mode ablation) are run across 3 seeds (mean $\pm$ std). The 8-dataset FNO-S comparison uses training logs (single run per dataset); repeating those with multiple seeds is left for future work.

3. **Reduced resolution (32×32).** This limits $k_{\max} \leq 16$ Fourier modes, likely disadvantaging the spectral arm. We expect the LiteFNO–CNN gap to widen at higher resolutions; this is a hypothesis, not a result.
4. **Transduction not implemented.** The paper's spatio-temporal extension requires conjugate-symmetry initialization described in Appendix A.3 of the original paper, which we found under-specified for reimplementation without access to the original code. Our results are single-step only; the rollout extension provides a proxy but is not equivalent to the transduction protocol.
5. **CP rank and protocol mismatches.** Integer vs. fractional CP rank, 200 vs. 500 epochs, MSE vs. relative-L^2 loss, and 32×32 vs. 64×64 resolution all mean our numbers are not directly comparable to the paper's Tables 6/7. All deviations are tabulated in Table 1.
6. **Training-set confound.** LiteFNO trained on 320 trajectories from 2 GS regimes; the CNN trained on the full GS dataset (roughly 1,000 trajectories). Both are in-distribution on the maze test, but the CNN saw more diverse training data. A fully controlled comparison would use the same split.
7. **AMP disabled for Arm A.** Complex-weight layers in `neuraloperator` are incompatible with PyTorch's `GradScaler`; Arm A trained in fp32. This gives Arm B a slight wall-clock advantage not reflected in epoch-time comparisons.

13 Ethical Considerations

This work uses The Well [5], a publicly available benchmark, and involves no human subjects or sensitive data. Our motivation, efficient PDE surrogates for climate-adaptation decision support in low-resource settings, is socially beneficial, but we do not overstate the practical readiness of any of the models studied. Our critique of implementation ambiguities in the paper under study is intended constructively to aid future practitioners, not adversarially.

14 Conclusion

We presented a from-scratch reproducibility and ablation study of LiteFNO [3] on the Gray–Scott dataset. We implemented two compact arms in a controlled head-to-head, a spectral CP-factorized LiteFNO (179,666 params) and a low-rank CNN ablation (107,842 params), plus a dense FNO-S baseline (roughly 9.47M params) for 8-dataset efficiency comparisons, and documented the implementation ambiguities encountered when reimplementing the paper's architecture.

Our main finding, across three random seeds, is that the Fourier inductive bias provides no consistent advantage over a parameter-matched low-rank CNN at 32×32 resolution. The CNN outperforms LiteFNO on one-step VRMSE (all seeds: 0.0105 ± 0.0001 vs. 0.0132 ± 0.0004) and on rollout on average. A single-seed run had suggested the opposite rollout ordering, but this was not reproducible. The spectral drift pattern (CNN over-smooths, LiteFNO over-sharpens) is observable individually but seed-inconsistent. Seed-robust mechanistic analysis (3 seeds) confirms all Fourier modes are load-bearing, the spectral capacity is fully utilised, yet this does not translate to accuracy gains at this scale.

Our reproduction finding confirms LiteFNO's efficiency thesis: both compact models ($\leq$180K parameters) outperform a dense FNO-S baseline roughly 50–88$\times$ larger by 1–96% across seven Well datasets, whether the compact model is spectral or convolutional. The efficiency gain is from compactness; the stability gain is from the Fourier inductive bias.

For practitioners at small spatial scales (32×32): the low-rank CNN is preferable, it is more accurate, more parameter-efficient, and roughly $3\times$ faster on GPU. Whether LiteFNO's spectral inductive bias provides advantages at larger resolutions (where more Fourier modes are available) remains an open question.

Reproducibility Checklist

Item	Status
Code	https://github.com/AIscend-Research/litefno-repro
Data	Public: The Well [5]; download instructions in repo
Preprocessing	Fully specified in Section 5.1
Architecture	All arms specified in Section 4
Hyperparameters	Fully specified in Section 5.2
Random seeds	Seed 1337 (all arms)
Hardware	NVIDIA T4 GPU (Kaggle)
Training time	Arm A: roughly 1.9 GPU-hr (roughly 24 s/epoch $\times$ 200); Arm B/C: not separately recorded
Checkpoints	Available at https://github.com/AIscend-Research/litefno-repro
Evaluation code	Available at https://github.com/AIscend-Research/litefno-repro

References

1. N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. "Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs." In: **Journal of Machine Learning Research** 24.89 (2023), pp. 1–97.
2. Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. "Fourier Neural Operator for Parametric Partial Differential Equations." In: **International Conference on Learning Representations (ICLR)**. 2021.
3. D. Ahn, S. Chandran, D. Leibovici, N. Kovachki, E. E. Papalexakis, and J. Kossaifi. "Lightweight Fourier Neural Operator for Time-Dependent Partial Differential Equations." In: **Machine Learning and the Physical Sciences Workshop (ML4PS), NeurIPS 2025**. 2025.
4. J. Kossaifi, N. Kovachki, K. Azizzadenesheli, and A. Anandkumar. "Multi-Grid Tensorized Fourier Neural Operator for High-Resolution PDEs." In: **arXiv preprint arXiv:2310.00120** (2023). URL: https://arxiv.org/abs/2310.00120.
5. R. Ohana et al. "The Well: a Large-Scale Collection of Diverse Physics Simulations for Machine Learning." In: **Advances in Neural Information Processing Systems** 37 (2024), pp. 44989–45037. DOI: 10.52202/079017-1430.
6. J. Pineau et al. "Improving Reproducibility in Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program)." In: **Journal of Machine Learning Research** 22.164 (2021), pp. 1–20.
7. I. Loshchilov and F. Hutter. "Decoupled Weight Decay Regularization." In: **International Conference on Learning Representations (ICLR)**. 2019.