The Alternation Depth Principle for Neural Operator Design

Haoze Song[1] **Zhilu Lai**[1,2,†] **Wei Wang**[1,2,†]
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology
hsong492@connect.hkust-gz.edu.cn
{zhilulai, weiwcs}@ust.hk

Abstract

Despite the success of Neural Operators (NOs), architectural hyperparameters such as depth are still typically selected by empirical grid search, with limited connection to the governing PDE. We introduce the *alternation depth*, α_F, a structural quantity of the PDE right-hand side that counts the minimum number of non-pointwise/pointwise coupling stages needed to build its leading nonlinear spatial structure. We use $L_{\min} = \alpha_F + 1$ as a *depth-efficiency lower-bound guideline* for IVP solution maps: a starting depth, not a predictor of the empirical optimum L^*. We provide partial formal support: (*sufficiency*) an L-block architecture can approximate any operator in a formally defined L-alternation class under compactness and regularity assumptions; (*separation*, proof sketch) for a constructed hard family at finite resolution and polynomial activations, we argue that using fewer blocks forces width to grow as a power of the discretization size. Empirically, a study over nine in-scope IVP datasets plus one Darcy stress test, using FNO-family variants and one kernel-integral baseline (KIN), is consistent with depth improvements appearing at or above $L_{\min}$ in the tested settings. The observed optimum can be larger because of spectral approximation, pointwise approximation difficulty, data regime, and optimization. We summarize the resulting practitioner guidance in four capacity-allocation rules.

1 Introduction

A practitioner's puzzle. Why can some operator-learning problems be handled with a single propagation block, while Burgers or 2D Navier–Stokes reliably benefit from additional depth? More broadly: given a novel PDE, how should one choose a reasonable starting depth before launching a large architecture sweep, and which part of that choice should be attributed to the PDE rather than to optimization or parameter budget?

Neural operators [22, 21, 8, 36, 34] are rapidly becoming the standard data-driven paradigm for learning PDE solution maps, driving advances from weather forecasting [28] to industrial simulations [36]. Yet, the depth L of these models is routinely chosen through brute-force grid search. The field currently lacks a simple structural rule for choosing a reasonable minimum depth *a priori*. This gap is costly, as hyperparameter search spaces scale multiplicatively with every added architectural dimension.

The alternation structure. Our starting point is a structural analysis of the feature updates constructed within a single neural operator block. In standard architectures, each block comprises two stages:

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1. A *non-pointwise linear* stage $\mathcal{K}$ (e.g., spectral convolutions, graph kernels, or attention) that aggregates information across the spatial domain.
2. A *pointwise nonlinear* stage $\sigma \circ W$ (e.g., an activation function applied to a channel-mixed representation) that generates local nonlinear features.

Neither stage alone captures nonlinear spatial operators: $\mathcal{K}$ is spatial but linear, while $\sigma \circ W$ is nonlinear but local. Their alternation constructs terms such as $u\,\partial_x u$, and repeated alternations build progressively richer operator classes. This suggests that the needed depth should be tied to PDE structure rather than tuned blindly.

Main Contributions.

- **(C1) The depth-efficiency framework.** We formalize pointwise/non-pointwise alternation as a shared structural primitive across NOs. This yields a PDE-derived guideline, $L_{\min} = \alpha_F + 1$, supported by an upper-bound approximation theorem for alternation classes and a finite-resolution hard-family separation result, given with a proof sketch.
- **(C2) The capacity-allocation principle.** The theory and experiments jointly suggest that, in the tested nonlinear settings, depth is the first lever for realizing additional coupling stages, while channel width and mode counts are second-order levers refining approximation within a fixed stage budget. For linear or additive semilinear problems, extra depth is not structurally required, although it may still help finite-width approximation or optimization.
- **(C3) Broad empirical evaluation.** We study nine IVP datasets plus one Darcy stress test, seven FNO-family variants plus a kernel baseline (KIN), five training-set sizes, and depths up to $L{=}40$. Across the in-scope IVP datasets, the best tested depth never falls below $L_{\min}$, and Burgers sample scaling moves L^* closer to $L_{\min}$ as data grows (Section H.7).
- **(C4) A Practitioner's Guide.** We separate theorem-level statements, structural heuristics, and empirical observations, and translate the resulting picture into four actionable design rules that can narrow depth search before expensive tuning (Section 5).

2 Preliminaries

2.1 Neural Operator Architecture

A neural operator maps an input function $a \in \mathcal{A}$ to an output function $u \in \mathcal{U}$ through a pointwise lifting $\mathcal{P}$, L propagation blocks, and a pointwise recovery $\mathcal{Q}$. We use two standard block forms:

Parallel form: $\sigma(\mathcal{K}v + Wv)$. Used by FNO [22], GNO [21], and MWT [8]:

$$v_{\ell+1}(x) = \sigma\big((\mathcal{K}_\ell\, v_\ell)(x) + W_\ell\, v_\ell(x)\big). \tag{1}$$

Sequential form: $\sigma(W \cdot \mathcal{K}v)$. Used by Transolver [36] and LNO [34]:

$$v_{\ell+1}(x) = \sigma\big(W_\ell\, (\mathcal{K}_\ell\, v_\ell)(x)\big). \tag{2}$$

Both forms contribute one non-pointwise/pointwise stage per block, so they have the same alternation count. At finite width, the parallel form is easier to optimize because σ sees both non-pointwise and local features directly; the sequential form must route local information through $\mathcal{K}$ (Section H.12).

The PDE setting. We consider initial-value problems

$$\partial_t u = F(u), \quad u(0) = u_0, \tag{3}$$

where F is a (possibly nonlinear) operator on a function space over Ω. The goal is to learn the solution map

$$\mathcal{G}\colon u_0 \mapsto u(T) \tag{4}$$

using a neural operator.

2.2 How Depth Controls Compositional Complexity

The class of representable operators expands with depth. With one block, σ can only act on linear spatial features of the lifted input. With two blocks, $\mathcal{K}$ can act on features already made nonlinear by the first block. That extra stage is the key structural resource behind the depth effect.

For Burgers, one block can form $u\,\partial_x u$, so $\alpha_F = 1$. Section 3 formalizes this idea.

3 The Alternation Depth Principle

3.1 Alternation Depth

Section 2.2 established that each block combines a non-pointwise linear transform with pointwise nonlinear coupling. We now formalize the minimal number of such coupling stages needed to build a given right-hand side, using a grammar of operator classes.

Definition 1 (Alternation classes)**.** *Let $\mathcal{A}$ be a class of non-pointwise* bounded *linear operators on $L^2(\Omega)$, with the operators used in Theorem 2 additionally assumed compact (e.g., compact integral operators with smooth kernels, Fourier multipliers with finitely supported symbol, or spectral truncations of differential operators such as ∂_x or Δ). Raw unbounded differential operators are not elements of $\mathcal{A}$; in the architecture they are represented through spectral truncations (cf. Remark 5). We define the alternation class hierarchy inductively:*

- $\mathcal{B}_0 = \big\{ u \mapsto f(u(x)) + \sum_{i=1}^{m} B_i\,(K_i\,u)(x) \;\big|\; f \text{ locally Lipschitz},\ K_i \in \mathcal{A},\ B_i \in \mathbb{R}^{q\times r_i} \big\}$.
- $\mathcal{B}_{\ell+1} = \big\{ u \mapsto \phi\big(u(x),\, A_1 h_1(x),\, \ldots,\, A_m h_m(x)\big) \;\big|\; h_i \in \mathcal{B}_\ell,\ A_i \in \mathcal{A},\ \phi \text{ locally Lipschitz} \big\}$.

The alternation depth *of an operator F is*

$$\alpha_F \;=\; \min\big\{ \ell \ge 0 : F \in \mathcal{B}_\ell \big\}.$$

$\mathcal{B}_0$ contains additive combinations of a purely pointwise nonlinear term and linear non-pointwise terms. It does *not* allow nonlinear coupling between local values $u(x)$ and spatial features $K_i u(x)$; such terms first appear in $\mathcal{B}_1$. $\mathcal{B}_{\ell+1}$ adds one more round in which a non-pointwise operator acts on outputs of $\mathcal{B}_\ell$ and the result is fused pointwise.

Example: Burgers ($\alpha_F = 1$). The advection term is $F(u) = -u\cdot\partial_x u + \nu\nabla^2 u$. We verify $\alpha_F = 1$:

- $F \in \mathcal{B}_1$: Take $h_1 \in \mathcal{B}_0$ to be the identity map $h_1(u) = u$ (an element of $\mathcal{B}_0$ via $f = \mathrm{id}$, $m = 0$), and $A_1 = \partial_x$ (a spectral-truncated derivative, in $\mathcal{A}$). Then $(A_1 h_1)(u)(x) = (\partial_x u)(x)$, and $F(u)(x) = \phi(u(x),\, (A_1 h_1)(u)(x)) + \nu(\nabla^2 u)(x)$, where $\phi(a, b) = -a \cdot b$. The first term couples $u(x)$ with the non-pointwise feature $A_1 h_1(x)$; the second is a $\mathcal{B}_0$ additive term. So $F \in \mathcal{B}_1$.
- $F \notin \mathcal{B}_0$: $\mathcal{B}_0$ allows only additive combinations $f(u(x)) + \sum B_i K_i u(x)$. The term $u \cdot \partial_x u$ is a *multiplicative* coupling of $u(x)$ with $(\partial_x u)(x)$—not an additive combination of a purely pointwise term and a purely linear spatial term. Hence $F \notin \mathcal{B}_0$.

Therefore $\alpha_F = 1$. The identity $u \cdot \partial_x u = \frac{1}{2}\partial_x(u^2)$ further reveals that the dynamics require differentiating a nonlinear quantity—a structural signature of $\alpha_F \ge 1$.

Example: Allen–Cahn ($\alpha_F = 0$). $F(u) = u - u^3 + \varepsilon\nabla^2 u = f(u(x)) + B_1\,(\nabla^2 u)(x)$, where $f(a) = a - a^3$ is locally Lipschitz and ∇^2 is non-pointwise linear. This is exactly the additive form $f(u) + BKu$ defining $\mathcal{B}_0$, so $\alpha_F = 0$. The key: the nonlinear reaction ($u - u^3$) and the linear diffusion ($\varepsilon\nabla^2 u$) are *added*, not multiplicatively coupled.

PDE term	Structure	α_F	Reason
$\nu\nabla^2 u$ (Heat)	BKu (linear spatial)	0	additive $\mathcal{B}_0$: no coupling
$u - u^3$ (Allen–Cahn)	$f(u)$ (pointwise)	0	additive $\mathcal{B}_0$: no coupling
$u \cdot \partial_x u$ (Burgers)	$\phi(u, Ku)$ (coupled)	1	multiplicative coupling of u with Ku
$u \cdot A(u \cdot Au)$ (D8)	nested $\mathcal{K}$-then-σ	2	two rounds: $\mathcal{K}_1$ gives Au, σ_1 forms $u\cdot Au$; $\mathcal{K}_2$ gives $A(u\cdot Au)$, σ_2 forms product

Interpretation. The counting is about the minimal number of nested non-pointwise/pointwise rounds in a representation of F, not the literal instruction order inside a single block. Burgers needs one such round; D8 needs two.

Computing α_F. (i) Write the PDE as $\partial_t u = F(u)$ and fix the state variables and the operator class $\mathcal{A}$; (ii) parse F into a tree whose internal nodes are pointwise maps or operators in $\mathcal{A}$; (iii) $\alpha_F = 0$ if F is an additive combination of a pointwise term and linear non-pointwise terms; (iv) each round in which an operator from $\mathcal{A}$ acts on an already nonlinear feature, fused pointwise, adds one level; (v) take the minimum over all valid parses (Definition 1). For example, the advective form $u\,\partial_x u$ and the conservative form $\frac{1}{2}\partial_x(u^2)$ of Burgers both give $\alpha_F = 1$. α_F depends on the chosen state variables and on $\mathcal{A}$: it is invariant under re-parsing the same F, but not necessarily under reformulations that introduce auxiliary fields.

3.2 Depth-Efficiency Guideline

Depth-efficiency guideline. *For IVP solution maps, we use $L_{\min} = \alpha_F + 1$ as a leading-structure guideline.*

Status: This is a structural heuristic, not a theorem for exact PDE solution operators. In particular, the "+1" is an empirically supported starting-depth rule rather than a proved bound, and the block count is the same for residual and non-residual blocks. It is most naturally interpreted through short-time / first-order views such as

$$u(T) \;=\; u_0 + T\,F(u_0) + O(T^2),$$

a first Picard iterate, or a learned one-step discretization. Higher-order compositions, finite-width effects, spectral truncation, and optimization can all push the empirical optimum above $L_{\min}$.

For additive semilinear problems such as Allen–Cahn, $F \in_0$ and the guideline gives $L_{\min} = 1$: extra depth may still help, but not because the leading alternation structure forces it. For Burgers, the identity $u\,\partial_x u = \frac{1}{2}\partial_x(u^2)$ illustrates why one level is higher: a spatial operator must interact with a nonlinear quantity, so $\alpha_F = 1$ and $L_{\min} = 2$.

3.3 Main Results

We state two formal results: an approximation theorem (non-polynomial σ, Part (i)) and a depth-separation result (polynomial σ, Part (ii)) whose proof is a sketch. These use different activation assumptions and provide complementary evidence—sufficiency from above and hardness from below. They also concern different operator classes: the hard family uses a flat-spectrum A_N that does not satisfy assumption (A2) of Theorem 2.

Theorem 2 (Alternation-class approximation, fixed depth)**.** *Consider the following assumptions:*

- *(A1) The target operator $G \in_L$ for some $L \geq 1$, mapping a compact subset $\mathcal{K}_{\text{in}} \subset H^s(\Omega)$ (with $s > d/2$) into $L^2(\Omega)$.*
- *(A2) The non-pointwise operators in the $_L$ representation of G are compact with singular values satisfying $s_j \leq C\,j^{-\beta/d}$ for some $\beta > d/2$ and universal constant $C > 0$.*
- *(A3) The activation σ is non-polynomial and Lipschitz.*
- *(A4) The depth L is treated as a fixed structural constant (determined by α_F, not by ε); the maximum per-stage Lipschitz constant $M = \max_\ell \operatorname{Lip}(\varphi_\ell \circ \mathcal{A}_\ell)$ is likewise fixed.*

Then for any $\varepsilon > 0$, there exists an L-block neural operator (1) *with spatial rank $r = O(\varepsilon^{-d/\beta})$ modes and sufficiently large channel width $d_v = d_v(\varepsilon, r)$ satisfying*

$$\sup_{u_0 \in \mathcal{K}_{\text{in}}} \left\| \hat{\mathcal{G}}(u_0) - G(u_0) \right\|_{L^2(\Omega)} \;\leq\; \varepsilon.$$

Each block realizes one $\mathcal{K}$-then-σ stage: $\mathcal{K}$ is approximated by finite-rank truncation (retaining r modes; operator-norm error $\|K - K^{(r)}\|_{\text{op}} \leq C\,r^{-\beta/d}$ by A2), and σ approximates the pointwise coupling by universal approximation with d_v channels. If the pointwise couplings additionally satisfy a quantitative single-hidden-layer approximation bound under σ, the same construction yields a polynomial-in-$1/\varepsilon$ width bound for fixed L (proof: Section A.1).

Table 1: **Claim status.** What is defined, proved, assumed, heuristic, or observed.

Claim	Status	Evidence
α_F and the alternation classes $_\ell$	Definition	Definition 1
An L-block NO approximates operators in $_L$	Proved (A1–A4)	Theorem 2
$k<L$ blocks need width $\geq cN^{c'/(L-k)}$ (hard family)	Stated, proof sketch	Theorem 3
A given PDE solution map lies in $_L$	Assumption	Remark 15
$L_{\min} = \alpha_F + 1$ as starting depth	Heuristic	Section 3.2
Gains appear at or above $L_{\min}$ (FNO family, KIN)	Observed	Table 2, Figure 1
Widening below $L_{\min}$ plateaus	Observed	Table 3
L^* moves toward $L_{\min}$ as data grows	Observed (Burgers)	Table 25
Deep-vanilla collapse is fixed by skips, not by clipping	Observed (Burgers)	Section I.1
Depth threshold beyond FNO (GNO, attention)	Observed; structural counting	Section I.2

Table 2: $L_{\min}$ **as a lower bound.** Across the in-scope IVP datasets, the best tested depth never falls below $L_{\min}$. L^* is from single-seed scans; the 3-seed best depths (Table 10) differ for D1, D2, D9 and D10.

Dataset	α_F	$L_{\min}$	L^*
In-scope: IVP solution maps $u_0 \mapsto u(T)$.			
D6 Linear	0	1	**1**
D1 Heat	0	1	6
D2 Allen–Cahn	0	1	4
D3 Burgers	1	2	4
D7 Synth. ord 2	1	2	4
D8 Synth. ord 3	2	3	5
D4 NS 2D	1	2	7
D9 KdV	1	2	4
D10 Sine-Gordon	0	1	4
Out-of-scope stress test.			
D5 Darcy	—	—	13

Theorem 3 (Finite-resolution separation; proof sketch)**.** *There exists a hard family $\{H_L\}_{L\geq 1}$ of operators with $H_L \in_L \setminus_{L-1}$ (alternation depth exactly L) such that the following holds. For any neural operator with $k < L$ blocks and polynomial activation of degree p, operating on an N-point spatial grid, approximation of H_L to constant relative accuracy requires channel width*

$$d_v \;\geq\; c\, N^{c'/(L-k)},$$

where $c, c' > 0$ are constants depending on p. This is a finite-resolution hard-family result under polynomial activations: we use it as evidence that insufficient depth can force large width, not as a universal lower bound for all PDE solution maps or GELU models (proof sketch: Section A.3).

Remark. Theorem 2 and Theorem 3 are complementary but limited: the approximation theorem is an upper bound for targets already known to lie in $_L$, while the separation result (proof sketch) is a lower bound for a constructed finite-resolution hard family. Together with the depth-efficiency guideline (Section 3.2), they provide partial formal support for the alternation-depth viewpoint; they do not by themselves prove $L_{\min}$ for general PDE solution maps.

Proofs, quantitative qualifications, and block-form details are deferred to Sections A.1, A.3 and H.12.

3.4 Scope and Experimental Alignment

Table 2 shows how α_F and $L_{\min}$ align with empirically optimal depth L^* across nine IVP datasets plus one Darcy stress test. The gap $L^* - L_{\min}$ comes from finite-width spectral approximation, higher-order solution terms, dimensionality, and, for Darcy, inverse-operator convergence rather than alternation complexity.

We next evaluate these predictions experimentally on nine IVP datasets plus the Darcy stress test.

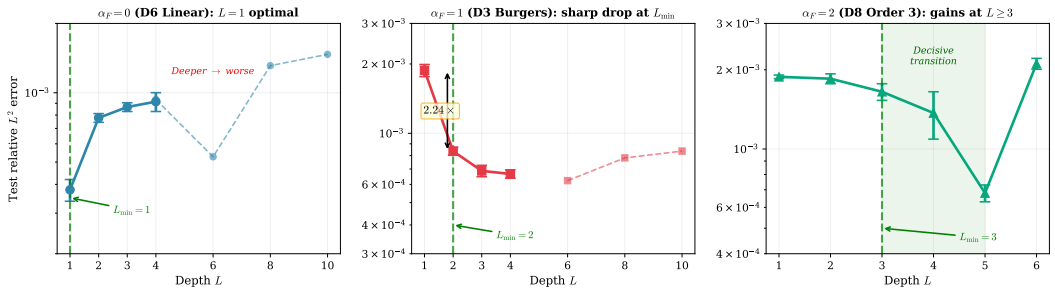


Figure 1: **Controlled depth scans.** For $\alpha_F = 0, 1, 2$, the first clear gains appear at or above the predicted $L_{\min} = \alpha_F + 1$. Exact seed counts are given in the appendix.

Table 3: **Width plateaus below the depth threshold.** At fixed L=1, widening helps only modestly; adding depth at similar or smaller parameter count yields the decisive improvement.

Dataset (α_F, $L_{\min}$)	d_v=32	d_v=64	d_v=128	d_v=256	Best at $L \geq L_{\min}$
D3 Burgers (α_F=1, L=1)	.00297	.00159	.00116	.00132	**.000671** (L=2, d_v=64)
D7 Synth. ord 2 (α_F=1, L=1)	.00236	.00187	.00167	.00173	**.000641** (L=4, d_v=128)
D8 Synth. ord 3 (α_F=2, L=1)	.00261	.00188	.00184	.00193	**.000644** (L=3, d_v=128)

4 Experiments

We evaluate the alternation-depth hypothesis across training runs spanning nine IVP datasets plus one Darcy stress test, 7 architecture variants, depths L=1–40, and 5 training-set sizes. Full reproducibility details appear in Section H.

4.1 Setup

We test on three synthetic datasets (D6–D8) with known alternation complexity $\alpha_F \in \{0, 1, 2\}$ and seven physical PDE datasets (D1–D5, D9–D10), treating Darcy (D5) as an out-of-scope stress test rather than part of the main IVP claim (Table 5). The base architecture is Vanilla FNO [22] with d_v=64, 32 Fourier modes (1D) or 12×12 modes (2D), trained for 500 epochs with Adam [14] (lr=10^{-3}, cosine annealing). All reported errors are relative L^2 on held-out test sets. Statistical significance is assessed via Welch's t-test over 10 independent seeds where indicated.

4.2 Depth Phase Transition

Figure 1 and Table 2 present the depth scan across the full dataset suite. Three controlled cases isolate the role of alternation complexity. D6 (α_F=0) is the negative control: in 10-seed validation, L=1 is best and L=2 is significantly worse ($p = 5.7 \times 10^{-15}$). D3 Burgers (α_F=1) shows the transition: error drops 2.24× from L=1 to L=2 ($p = 3.8 \times 10^{-11}$), exactly at $L_{\min}$=2, with the best vanilla depth at L^*=4. D8 (α_F=2) shows only mild change below L=3 and clearer gains once depth reaches the predicted threshold.

Two anomalies clarify the difference between a lower-bound threshold and the empirical optimum. Heat (α_F=0) continues improving to L=6 because extra blocks refine the fixed-width spectral approximation of the semigroup. Darcy, treated here only as a stress test, prefers much larger depth because the inverse operator has a slowly converging Neumann expansion.

4.3 Depth-vs-Width Separation: Width Plateaus Below $L_{\min}$

The strongest evidence for the depth-efficiency framework is not that "deeper is better," but that *depth beats width* when the alternation budget is insufficient. Table 3 shows this directly: at $L < L_{\min}$, sweeping d_v over an 8× range yields a plateau, while adding even one block at modest width breaks through.

At L=1, widening by $8\times$ yields only limited gains on D3, D7, and D8, and the improvement often reverses beyond d_v=128. By contrast, adding one block at smaller or comparable parameter count breaks through the plateau: on D3, L=2, d_v=64 beats L=1, d_v=128 with roughly half the spectral parameter budget, and on D8 the first clear break occurs only once depth reaches L=3. This is the practical version of the depth-vs-width story: width refines an available stage budget, but does not reliably replace a missing one.

4.4 Architecture Comparison

Across five tested FNO-family variants, the D3 Burgers L=1→L=2 ratio stays in the narrow range 1.79–2.42×, so the threshold location is not an artifact of one particular block design (Table 21). Residual FNO is the strongest practical variant in this sweep and remains stable far beyond vanilla's collapse depth; full architecture tables and matched-depth curves are deferred to Section H.5. The residual and AttnRes runs in this sweep use gradient clipping while vanilla does not; a controlled 2×2 ablation on D3 Burgers (Section I.1) shows that vanilla collapses at $L\geq16$ with and without clipping, while residual without clipping remains stable to L=20.

4.5 Additional Findings

The appendix provides the rest of the empirical picture. On Burgers sample scaling, L^* decreases from 10 to 3 as N_{train} grows from 100 to 1800, moving toward $L_{\min}$=2 (Section H.7). At matched parameter count, narrow-deep beats wide-shallow on the tested problems (Section H.8). KIN shows the same qualitative depth trend on D3/D8 (Section H.9). Additional 2–3 seed sweeps across resolution, width, sample size, and activation keep L^* at or above $L_{\min}$, while mechanism sweeps show that the threshold location is more stable than the transition steepness (Sections H.2, H.3 and H.13).

Having established the main empirical picture, we now translate it into actionable design rules.

5 Practitioner's Guide

We distill the theory and experiments into four rules.

Rule 1 (Set depth-efficiency threshold from the PDE). Compute the alternation depth α_F of the target PDE's right-hand side (Definition 1 and table 2) and set $L \geq \alpha_F + 1$. This bound assumes the leading nonlinear structure of F determines depth requirements and should be read as a starting point for IVP-like solution maps. It is a *lower bound*, not an optimum: the practical optimum additionally depends on spectral complexity, data regime, optimization dynamics, and spatial dimensionality.

Rule 2 (Diagnose depth starvation via width plateau). If widening d_v yields diminishing returns while error remains high, the network lacks depth, not capacity. Our width-screening experiment (Section 4.3) shows that more parameters yield diminishing returns at $L < L_{\min}$ in the tested configurations. The diagnostic signature is a flat error curve across $d_v \in \{32, 64, 128, 256\}$ at a fixed shallow depth.

Rule 3 (Depth-first allocation at fixed budget). Under a fixed parameter budget on nonlinear PDEs, prefer narrow-deep over wide-shallow configurations. Concretely, L=6, d_v=56 outperforms L=2, d_v=96 by 7–17% at matched parameter count (∼1.4M) across three PDEs (Section 4.5). For linear PDEs (α_F=0), L=1 is expressively sufficient.

Rule 4 (Prefer residual connections). In our architecture comparison, the identity skip $v_{\ell+1} = v_\ell + \sigma(\mathcal{K}_\ell v_\ell)$ is the most reliable way to prevent depth collapse (Section 4.4). Among five tested depth-mixing strategies, this simplest residual wins every dataset (Section H.5). A controlled clipping $\times$ architecture ablation on D3 Burgers (Section I.1) indicates that this stabilization comes from the skip connection, not from gradient clipping.

Related work. Depth-expressivity theory for standard networks [32, 6, 5] proves that deeper circuits express exponentially more complex functions via alternating linear and nonlinear maps. Neural operator theory [15–17] has established approximation rates and universal approximation for infinite-dimensional maps: FNOs are universal [15], and even minimal nonlocality (an integral average) combined with nonlinearity suffices for universality [18]. These results address whether an

operator can be approximated with sufficient capacity; our question is how much depth is needed to approximate it efficiently. Our work connects these lines by proposing that the number of required spatial-aggregation/nonlinearity alternations in the PDE sets a depth-efficiency threshold. Broader architectural scope is discussed in Sections C and G.

These rules provide a practical starting point; we now summarize the broader contributions and scope of our results.

6 Conclusion

We introduced the alternation depth α_F—a structural property of the PDE's right-hand side that counts the minimum number of non-pointwise/pointwise coupling stages in a minimal representation of its leading spatially nonlinear structure—and proposed $L_{\min} = \alpha_F + 1$ as a *depth-efficiency lower-bound guideline* (not a predictor of the empirical optimum L^*). Our formal support is partial: an upper-bound theorem for a defined alternation class at fixed depth, and a finite-resolution hard-family separation for polynomial activations (proof sketch). An empirical study across nine IVP datasets plus one Darcy stress test, seven FNO-family variants, and one KIN baseline is consistent with a depth-efficiency threshold at or above $L_{\min}$ in the tested settings; on Burgers sample-scaling sweeps, L^* moves closer to $L_{\min}$ as data grows. These results provide a starting point for depth selection derived from PDE structure. Beyond FNO-family variants and KIN, depth scans of a GNO and a Transolver-style attention operator on Burgers and D8 (Section I.2) give best depths at or above $L_{\min}$ in three of four cells; the attention operator's D8 curve is flat across depths and therefore uninformative about depth. This extension is empirical and not covered by our theorems; the broader compatibility argument is documented in Section E.

Limitations and scope. The main caveats are straightforward: the sufficiency theorem is for alternation classes rather than exact PDE solution maps, the separation result is a proof sketch for a hard family under polynomial activations, and the "$+1$" rule is motivated by short-time / learned-update views rather than exact mild-solution theory. Practical L^* still depends on spectrum, width, optimization, and higher-order solution structure; Darcy and attention-based operators remain structural edge cases. Extending the framework to broader activations, larger time horizons, and branch-trunk architectures remains open.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, 2019.

[3] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[4] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

[5] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pages 698–728. PMLR, 2016.

[6] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, pages 907–940. PMLR, 2016.

[7] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning*, pages 399–406, 2010.

[8] Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In *Advances in Neural Information Processing Systems*, volume 34, pages 24048–24062, 2021.

[9] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2nd edition, 2006.

[10] Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. GNOT: A general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 12556–12569, 2023.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[12] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

[13] Kimi Team, Guangyu Chen, Yu Zhang, Jianlin Su, Weixin Xu, et al. Attention residuals. Technical Report arXiv:2603.15031, Moonshot AI, 2026.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[15] Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for Fourier neural operators. *Journal of Machine Learning Research*, 22(290): 1–76, 2021.

[16] 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.

[17] Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. Error estimates for DeepONets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1), 2022.

[18] Samuel Lanthaler, Zongyi Li, and Andrew M. Stuart. Nonlocality and nonlinearity implies universality in operator learning. *Constructive Approximation*, 62(2):261–303, 2025. doi: 10.1007/s00365-025-09718-3.

[19] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.

[20] Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *TMLR*, 2023.

[21] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

[22] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

[23] 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.

[24] Huakun Luo, Haixu Wu, Hang Zhou, Lanxiang Xing, Yichen Di, Jianmin Wang, and Mingsheng Long. Transolver++: An accurate neural solver for PDEs on million-scale geometries. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 41432–41449, 2025.

[25] Robert I McLachlan and G Reinout W Quispel. Splitting methods. *Acta Numerica*, 11:341–434, 2002.

[26] Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

[27] Matteo Pagliardini, Amirkeivan Mohtashami, François Fleuret, and Martin Jaggi. DenseFormer: Enhancing information flow in transformers via depth weighted averaging. In *Advances in Neural Information Processing Systems*, 2024.

[28] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. FourCastNet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[29] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2007.

[30] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[31] Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968.

[32] Matus Telgarsky. Benefits of depth in neural networks. In *Conference on Learning Theory*, pages 1517–1539. PMLR, 2016.

[33] Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized Fourier neural operators. In *International Conference on Learning Representations*, 2023.

[34] Tian Wang and Chuang Wang. Latent neural operator for solving forward and inverse PDE problems. In *Advances in Neural Information Processing Systems*, 2024.

[35] Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. U-FNO—an enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022.

[36] Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 53681–53705, 2024.

[37] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, 2019.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction separate theorem-level results, the $L_{\min} = \alpha_F + 1$ structural guideline, and the empirical evidence. Scope restrictions and out-of-scope cases (notably Darcy) are stated explicitly in Section 3.4 and the Conclusion.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Conclusion lists explicit limitations on the theorem scope, activation assumptions, interpretation of the "+1" rule, architectural coverage, Darcy as an out-of-scope stress test, and attention-based operators as structural rather than theorem-level cases.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [No]

   Justification: The approximation theorem (Theorem 2) states all assumptions, with a full proof in Section A.1. The separation result (Theorem 3) is given with a proof sketch (Section A.3): its width-bound step builds on the generic CP-versus-Hierarchical-Tucker separation of [5], but the reduction from k-block neural operators, the bound for the specific hard-family tensor, and the exponent are not derived. The depth-efficiency guideline ($L_{\min} = \alpha_F + 1$) is presented as a principled structural heuristic, not a rigorous theorem for general PDE solution operators; this distinction is stated explicitly in the main text.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section B provides complete dataset specifications, hyperparameters, training protocols, and hardware details. All synthetic datasets (D6–D8) are fully specified including random seeds.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplementary material includes code, configuration files, LICENSE information, and reproduction instructions for the reported experiments.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All details are in Section B: optimizer (Adam), learning rate (10^{-3}), scheduler (cosine annealing), epochs (500), batch sizes, loss function, activation, Fourier modes, and the use of gradient clipping in the non-vanilla variants.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The two controlled phase-transition comparisons (D3 Burgers and D6 Linear) are validated with 10 independent seeds and Welch's t-tests. Additional depth scans and factor sweeps report 2–3 seed aggregates where available, and single-seed screening experiments are labeled as such.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section B reports the hardware, software stack, per-run timing, total wall-clock time, and total GPU-hours: $4\times$ NVIDIA RTX 4090 (24 GB each), roughly 2 minutes per 1D run, 20–30 minutes per 2D run, and about 7 GPU-days total.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work studies theoretical and empirical properties of neural network architectures for scientific computing. It does not involve human subjects, sensitive data, or potential for misuse.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is foundational research on neural network architecture theory for PDE solving. We do not foresee direct negative societal impacts. Positive impacts include more efficient scientific computing by reducing hyperparameter search costs.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release pre-trained models or scraped datasets. The code implements standard neural operator training on synthetic/public PDE datasets.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The Burgers, Darcy, and NS datasets from Li et al. [22] are cited, and the supplementary material documents the associated license or usage information for the reused assets.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The synthetic datasets D6–D8 are documented with the operator, damping, solver, resolution, and seed-generation details in Section B, and the supplementary package includes the corresponding code/configuration assets.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This work does not involve human subjects research.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs only contribute to the writing on this paper.

A Full Proofs

A.1 Proof of Theorem 2 (Alternation-Class Approximation)

We first state the regularity assumptions on the operator class and the architecture.

Assumption 4 (Kernel regularity)**.** *Each non-pointwise operator $\mathcal{K}$ appearing in the alternation decomposition of a target $\mathcal{G} \in_L$ is a* compact *operator on $L^2(\Omega)$ with singular values satisfying*

$$s_j(\mathcal{K}) \;\le\; C_{\mathcal{K}}\, j^{-\beta/d}, \qquad j = 1, 2, \ldots,$$

where $d = \dim(\Omega)$ and $\beta > d/2$ (a Sobolev smoothing rate).

Remark 5 (Compactness and differential operators)**.** *The compactness assumption (Assumption 4) applies to the integral kernel representation of spatial operators used within the neural architecture (e.g., spectral convolutions with finitely many modes, graph kernels with bounded range). Raw differential operators (∂_x, Δ) are unbounded on L^2; in the neural operator setting, they are represented through their spectral truncations, which* are *compact. The theory analyzes the architecture's internal operators, not the PDE's differential operators directly.*

Assumption 6 (Fixed pointwise coupling dimension)**.** *At each alternation stage ℓ, the pointwise coupling function φ_ℓ depends on a fixed number p_ℓ of inputs—the local channel values and the non-pointwise features required at that stage—where p_ℓ is determined by the structural complexity of $\mathcal{G}$ and is* independent *of the approximation accuracy ε.*

Assumption 7 (Activation for Theorem 2)**.** *The activation $\sigma : \mathbb{R} \to \mathbb{R}$ is locally Lipschitz, continuous, and non-polynomial. Non-polynomiality is the standard condition ensuring universal approximation in single hidden layers [19].*

Proof of Theorem 2. We constructively build an L-block neural operator that approximates any target $\mathcal{G} \in_L$ to accuracy ε in L^2.

Step 1: Theoretical block structure. We work with a slightly enriched block that includes a post-activation linear mixing:

$$v_{\ell+1}(x) \;=\; B_\ell\, \sigma\big((\mathcal{K}_\ell\, v_\ell)(x) + W_\ell\, v_\ell(x) + b_\ell\big), \tag{5}$$

where $B_\ell \in \mathbb{R}^{d_v \times d_v}$ is a pointwise-linear mixing matrix. At each spatial point x, the map $z \mapsto B_\ell\, \sigma(z)$ with input $z = \mathcal{K}_\ell v(x) + W_\ell v(x) + b_\ell$ has the form of a standard one-hidden-layer network

$B\,\sigma(Az+b)$. By classical universal approximation for single-hidden-layer networks with non-polynomial activations [19], such a map is a universal approximator for continuous functions on compact subsets of $\mathbb{R}^{p_\ell}$, provided the hidden dimension (here d_v) is sufficiently large.

Remark 8 (Realization of B_ℓ in practice)**.** *The post-activation linear map B_ℓ is pointwise-linear and therefore adds no non-pointwise operations; it does not change the alternation count of the block. In practical architectures, its role is realized by:*

1. *the recovery projection $\mathcal{Q}$ (for the final block),*
2. *the affine part of $\mathcal{K}_{\ell+1}+W_{\ell+1}$ in the subsequent block, which provides a post-activation linear readout for block ℓ (for intermediate blocks),*
3. *internal MLP structure within each block (as in Transolver and recent FNO variants).*

Since B_ℓ is pointwise-linear, it has alternation depth 0 and the alternation structure of the architecture is unchanged.

Step 2: One block approximates one $\mathcal{K}$-then-σ round.

Non-pointwise features via $\mathcal{K}_\ell$. Let $\mathcal{A}_\ell$ denote the compact non-pointwise operator at stage ℓ. By Assumption 4, the singular value decomposition gives a rank-r truncation $\mathcal{A}_\ell^{(r)}$ satisfying

$$\left\|\mathcal{A}_\ell-\mathcal{A}_\ell^{(r)}\right\|_{\mathrm{op}} \;\le\; C_{\mathcal{A}_\ell}\, r^{-\beta/d}.$$

Setting the right-hand side equal to δ yields

$$r \;=\; O\big(\delta^{-d/\beta}\big).$$

The kernel integral operator $\mathcal{K}_\ell$ is configured to produce r output channels realizing $\mathcal{A}_\ell^{(r)}(v)(x)$. The W_ℓ path and identity-like zero-frequency modes in $\mathcal{K}_\ell$ pass through local channel values unchanged. The pre-activation input at point x is therefore

$$z(x) \;=\; \big[\mathcal{A}_\ell^{(r)}(v)(x);\; W_\ell v(x);\; b_\ell\big] \;\in\; \mathbb{R}^{d_v}.$$

Pointwise coupling via $B_\ell\sigma(\cdot)$. By Assumption 6, the target pointwise coupling φ_ℓ at stage ℓ is a locally Lipschitz function of p_ℓ variables, where p_ℓ is a structural constant (e.g., for Burgers' equation, $p_\ell=2$: the value $u(x)$ and the spatial derivative feature $(\partial_x u)(x)$). The rank r controls the spatial resolution with which each of these p_ℓ non-pointwise features is represented, but the *number* of distinct features—hence the input dimension of φ_ℓ—remains p_ℓ, independent of ε.

By universal approximation for one-hidden-layer networks with non-polynomial σ (Assumption 7), there exists a finite hidden width $m_\ell(\delta)$ such that the map $z\mapsto B_\ell\,\sigma(z+b_\ell)$ approximates φ_ℓ on the relevant compact subset of $\mathbb{R}^{p_\ell}$ to accuracy δ. If one assumes a quantitative single-hidden-layer approximation rate for the class containing φ_ℓ (for example, a Barron- or Sobolev-type bound), then $m_\ell(\delta)$ inherits the corresponding polynomial dependence on $1/\delta$.

Combined single-block error. Taking d_v large enough to dominate both the spatial rank r and the required pointwise width $m_\ell(\delta)$, a single block (5) approximates one $\mathcal{K}$-then-σ stage with L^2 error $O(\delta)$.

Step 3: L blocks compose to approximate operators in $_L$. Any target $\mathcal{G}\in_L$ decomposes (by definition of $_L$) as L alternation stages:

$$\mathcal{G}=\mathcal{Q}\circ(\varphi_L\circ\mathcal{A}_L)\circ(\varphi_{L-1}\circ\mathcal{A}_{L-1})\circ\cdots\circ(\varphi_1\circ\mathcal{A}_1)\circ\mathcal{P},$$

where $\mathcal{A}_\ell$ is non-pointwise linear (compact), φ_ℓ is pointwise nonlinear, $\mathcal{P}$ is lifting, and $\mathcal{Q}$ is recovery.

When the target operator has a branching alternation graph (multiple φ_i at each level), the required width scales with the total number of branches, but the depth is determined by the longest alternation chain.

Step 4: Error propagation via Lipschitz composition. If each block incurs L^2 error δ, the total error after composing L blocks satisfies

$$\left\|\widehat{\mathcal{G}}(u_0) - \mathcal{G}(u_0)\right\|_{L^2} \;\le\; \sum_{\ell=1}^{L} \mathrm{Lip}(\mathcal{G}_{\ell+1:L}) \cdot \delta \;\le\; L \cdot M^{L-1} \cdot \delta,$$

where $M = \max_\ell \mathrm{Lip}(\varphi_\ell \circ \mathcal{A}_\ell)$ and $\mathcal{G}_{\ell+1:L}$ denotes stages $\ell+1$ through L. Setting $\delta = \varepsilon/(L \cdot M^{L-1})$ yields total error at most ε.

Note: The factor M^{L-1} means the width depends exponentially on L unless each stage has Lipschitz constant $M \le 1$ (a contractivity assumption) or L is treated as a fixed structural parameter. In practice, L is determined by α_F (typically 1–3) and is not a growing quantity.

Step 5: Width bound. For fixed L, choosing each block width large enough for its spatial truncation and pointwise approximation yields a finite-width approximant. The explicit rate we use in the main statement comes from the compact-kernel truncation $r = O(\delta^{-d/\beta})$. Any additional polynomial width claim for the pointwise part requires a quantitative approximation theorem for the relevant class of couplings φ_ℓ; when such a theorem is available, its rate transfers directly to d_v for fixed L. □

A.2 The W-path Extension

The block (1) includes a parallel pointwise path $W_\ell v(x)$ alongside the non-pointwise $(\mathcal{K}_\ell v)(x)$. We show that this does not change the alternation depth hierarchy.

Proposition 9 (W-path preserves alternation structure)**.** *Let $\mathcal{G}_L^{(\mathcal{K})}$ denote the operator class expressible by L blocks of the form $B\,\sigma(\mathcal{K}_\ell v + b)$ (canonical, no W-path), and $\mathcal{G}_L^{(\mathcal{K}+W)}$ the class for $B\,\sigma(\mathcal{K}_\ell v + W_\ell v + b)$ (with W-path). Then $\mathcal{G}_L^{(\mathcal{K})} \subseteq \mathcal{G}_L^{(\mathcal{K}+W)}$, and the separation result (Theorem 3) applies to both classes with the same depth threshold.*

Proof. Subsumption. Setting $W_\ell = 0$ recovers the canonical form, so $\mathcal{G}_L^{(\mathcal{K})} \subseteq \mathcal{G}_L^{(\mathcal{K}+W)}$ trivially.

No new alternations. The W-path computes $W_\ell v(x)$, a pointwise-linear function of v at x. This adds channels to the pre-activation input of σ but introduces no new non-pointwise information. The non-pointwise content available at block ℓ is determined entirely by $\mathcal{K}_\ell$. Formally, $\sigma(\mathcal{K}_\ell v + W_\ell v + b) = \sigma([\mathcal{K}_\ell + W_\ell]v + b)$ where $\mathcal{K}_\ell$ provides the non-pointwise part and W_ℓ the pointwise-linear part. Since pointwise-linear maps have alternation depth 0, the combined operator contributes the same single non-pointwise operation as $\mathcal{K}_\ell$ alone.

Separation still holds. The hard family H_L in Section A.3 requires L rounds of applying a non-pointwise operator to nonlinear intermediates. The W-path cannot substitute for $\mathcal{K}$ in any of these rounds because W is pointwise and cannot transport information across spatial locations. The width lower bound therefore applies identically to $\mathcal{G}_L^{(\mathcal{K}+W)}$.

Practical role. The W-path provides (i) dense channel mixing at each x (analogous to a 1×1 convolution in the channel dimension), and (ii) a gradient shortcut similar to residual connections. Our ablation (Section 4) confirms that removing W degrades optimization but does not shift the depth phase transition. □

A.3 Proof Sketch of Theorem 3 (Finite-Resolution Separation)

Assumption 10 (Activation for Theorem 3)**.** *The activation $\sigma : \mathbb{R} \to \mathbb{R}$ is a polynomial of degree p. (When σ is analytic with nonzero degree-$(L+1)$ Taylor coefficient, the argument applies to the degree-$(L+1)$ homogeneous component of the network output.)*

Proof sketch of Theorem 3. We work entirely in finite dimensions. We exhibit a family of operators on $\mathbb{R}^N$ for which, under assumptions (a)–(c) of Step 4, insufficient depth forces large width.

Step 1: The hard family $H_{L,N}$ on $\mathbb{R}^N$. Let $A_N \in \mathbb{R}^{N\times N}$ be a fixed invertible matrix with bounded condition number on its first N modes (e.g., a circulant matrix whose first N Fourier coefficients are

bounded above and below by positive constants independent of N). Define the family $\{H_{L,N}\}_{L\geq 0}$ recursively:

$$\begin{aligned} H_{0,N}() &= , \\ H_{1,N}() &= \odot(A_N), \\ H_{L,N}() &= \odot A_N\big(H_{L-1,N}()\big), \qquad L \geq 2, \end{aligned} \tag{6}$$

where $\in \mathbb{R}^N$ and $\odot$ denotes the Hadamard (entrywise) product.

We verify that $\alpha(H_{L,N}) = L$ (alternation depth equals L):

- $H_{0,N}() =$ is pointwise (identity), so $\alpha(H_{0,N}) = 0$.
- $H_{1,N}() = \odot A_N$ requires one non-pointwise operation A_N followed by pointwise multiplication: one non-pointwise-then-pointwise stage, so $\alpha(H_{1,N}) = 1$.
- $H_{L,N}() = \odot A_N(H_{L-1,N}())$: computing $H_{L-1,N}$ requires $L-1$ alternations, then one applies A_N (non-pointwise) to this nonlinear intermediate, then multiplies pointwise by . This final A_N-then-pointwise stage cannot be absorbed into the $L-1$ preceding stages because A_N acts on a *nonlinear* function of . Hence $\alpha(H_{L,N}) = L$.

The homogeneous degree of $H_{L,N}$ in is $L+1$ (each recursive step multiplies by once and applies the linear map A_N, adding one degree).

Remark 11 (Choice of A_N)**.** *The discretized operator A_N is chosen with bounded condition number on its first N modes. For a fixed continuous compact operator A, the singular values decay, and the lower bound constants may degrade with N. The separation result is therefore strongest when A_N is chosen independently at each resolution (e.g., as a circulant with flat spectrum on the active modes).*

Step 2: Tensor representation. The operator $H_{L,N}$ is an $(L+1)$-homogeneous polynomial map $\mathbb{R}^N \to \mathbb{R}^N$, representable by an $(L+2)$-way tensor $\in (\mathbb{R}^N)^{\otimes(L+2)}$ (one output index, $L+1$ input indices) with entries

$$_{i_0,i_1,\ldots,i_{L+1}} \;=\; \delta_{i_0 i_1}\,(A_N)_{i_1 i_2}\,\delta_{i_2 i_3}\,(A_N)_{i_3 i_4}\,\cdots$$

This tensor train has *alternating bond structure*: diagonal bonds (from pointwise multiplications $_i \cdot (\cdot)$, contributing Kronecker deltas) alternate with dense bonds (from the full-rank operator A_N, contributing the matrix $(A_N)_{ij}$). When A_N has rank N on the discretization grid, each dense bond has dimension N.

Step 3: Neural operator output as hierarchical tensor. Consider a k-block neural operator (with $k < L$) and polynomial activation $\sigma(z) = z^p$ (Assumption 10). The degree-$(L+1)$ homogeneous part of the output is a tensor in $(\mathbb{R}^N)^{\otimes(L+2)}$ that admits a hierarchical Tucker decomposition with:

- k hierarchical levels (one per block),
- bond dimension at each level bounded by d_v (the channel width).

Each block applies a p-linear map at each spatial point (from the degree-p activation), and the hierarchical composition of k such maps produces a tree-structured tensor network with k levels and bond dimensions bounded by d_v.

Step 4 (sketch): Width lower bound. This step is a sketch. Cohen et al. [5] show that, apart from a negligible set, tensors realized by polynomial-size Hierarchical Tucker (deep) decompositions require exponential CP rank (shallow) to be realized or approximated. The bound below additionally assumes, without proof, that (a) a k-block neural operator with degree-p activation yields a k-level Hierarchical Tucker decomposition with bond dimension at most d_v; (b) the constructed tensor train is not in the negligible set; and (c) the resulting width requirement scales with exponent $c'/(L-k)$. These steps are stated, not derived; the statement below should be read accordingly:

Theorem 12 (Depth–width tradeoff; proof sketch)**.** *Assuming (a)–(c) above, there exists a hard family $\{H_{L,N}\}$ with $\alpha(H_{L,N}) = L$ such that any neural operator with $k < L$ blocks and polynomial activation of degree p (Assumption 10), operating at discretization size N, requires width*

$$d_v \;\geq\; c \cdot N^{c'/(L-k)}$$

to approximate $H_{L,N}$ to constant relative Frobenius accuracy, where $c, c' > 0$ are constants depending only on p and the spectral gap of A_N.

In particular, for $k = L - 1$ (one block short of the matching depth), the bound is $d_v \geq c \cdot N^{c'}$, i.e., width must grow as a power of the discretization size.

Remark 13 (Practical implications of the discretization-dependent bound)**.** *The bound $d_v \geq c \cdot N^{c'/(L-k)}$ is stated explicitly in terms of the discretization size N.* If the sketched bound holds, *its practical consequences are:*

1. At fixed finite resolution N: *insufficient depth* would require *width growing as a power of N; even moderate N (e.g., $N = 64$ in 1D, $N^2 = 4096$ grid points in 2D)* would then imply *width requirements far exceeding standard architectures.*

Applied to our setting, *and assuming (a)–(c) of Step 4*: a k-block network (producing a k-level hierarchical decomposition) approximating (an L-bond tensor train with dimension-N bonds) to constant relative error *would require* width $d_v \geq c \cdot N^{c'/(L-k)}$. With the matching depth L, a finite-width approximation exists by the sufficiency proof (Section A.1); polynomial width additionally requires the quantitative pointwise approximation assumptions discussed there. Note that the flat-spectrum A_N used here does not satisfy assumption (A2) of Theorem 2, so the two results concern different operator classes and do not form a matched upper/lower bound pair. □

Remark 14 (Separation between activation assumptions)**.** *Theorem 2 and Theorem 3 use* different *activation function assumptions and provide complementary, not parallel, evidence:*

Theorem 2 (sufficiency) *requires σ to be locally Lipschitz, continuous, and non-polynomial (Assumption 7). Non-polynomiality is essential: polynomial activations of fixed degree p cannot universally approximate arbitrary Lipschitz functions in a single hidden layer, whereas non-polynomial locally Lipschitz activations can [19].*

Theorem 3 (separation) *requires σ to be a polynomial of degree p (Assumption 10). For analytic activations such as $\sigma(z) = z\,\Phi(z)$ (GELU), whose Taylor expansion contains all polynomial degrees, the degree-$(L+1)$ component of a k-block model retains the hierarchical tensor structure, and the rank bound applies to this component.*

We conjecture that the separation extends to non-polynomial activations; this is supported by empirical width-plateau observations but remains formally open.

Remark 15 (Scope of the sufficiency guarantee)**.** *The sufficiency result (Theorem 2) guarantees that an L-block neural operator can approximate any operator in $_L$ (operators whose computational graph decomposes into at most L non-pointwise-then-pointwise alternation stages). Membership in $_L$ is an assumption on the target: it asserts that the leading-order structure of the solution map is captured by L alternation stages with compact non-pointwise components (Assumption 4).*

B Complete Experimental Details

This section provides a self-contained description of all models, datasets, training protocols, and hyperparameters used in the paper.

B.1 Architecture Specifications

Notation. All architectures share the three-stage template of Section 2.1: lifting $\mathcal{P}$, L propagation blocks, recovery $\mathcal{Q}$. Let d_v denote the hidden channel width, m the number of spectral modes (per spatial dimension), and $\sigma = \mathrm{GELU}$.

Table 4: Architecture variants tested. All share the same lifting ($\mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_v}$, linear) and recovery ($\mathbb{R}^{d_v} \to \mathbb{R}^{d_v}$ with GELU $\to \mathbb{R}^{d_{\text{out}}}$, two-layer pointwise MLP). They differ only in the propagation block.

Architecture	**Block update**	**Key property**
Vanilla FNO	$v_{\ell+1} = \sigma(\mathcal{K}_\ell\, v_\ell + W_\ell\, v_\ell)$	No skip; each block replaces
Residual FNO	$v_{\ell+1} = v_\ell + \sigma(\mathcal{K}_\ell\, v_\ell + W_\ell\, v_\ell)$	Identity skip; Euler-step form
AttnRes FNO	$h_\ell = \sum_{i=0}^{\ell-1} \alpha_{i\to\ell}\, v_i$	Softmax attention over depth
KIN (Kernel Integral)	$v_{\ell+1} = \sigma(\mathcal{K}_\ell^{\text{RFF}}\, v_\ell + W_\ell\, v_\ell)$	RFF kernel replaces FFT
ξ-only	$v_{\ell+1} = \sigma(\mathcal{K}_\ell\, v_\ell)$	W-path removed

Vanilla FNO. *1D:* Lifting: $\mathrm{Linear}(\mathbb{R}^2 \to \mathbb{R}^{d_v})$, where input channels are $(a(x),\, x)$. Each block: $\mathrm{SpectralConv1d}(d_v, d_v, m) + 1{\times}1\ \mathrm{Conv1d}(d_v, d_v)$, followed by GELU. The spectral convolution retains the lowest m Fourier modes and applies a learned complex $d_v \times d_v$ matrix per mode. Recovery: $\mathrm{Linear}(d_v, d_v)$ with GELU, then $\mathrm{Linear}(d_v, 1)$.

2D: Identical structure with $\mathrm{SpectralConv2d}$ retaining $m_1 \times m_2$ modes in each half of the frequency domain (positive and negative first-axis frequencies). Input channels are $(a(x,y),\, x,\, y)$.

Residual FNO. Same as Vanilla FNO except each block adds a skip connection: $v_{\ell+1} = v_\ell + \sigma(\mathcal{K}_\ell\, v_\ell + W_\ell\, v_\ell)$. No additional parameters beyond the vanilla block.

AttnRes FNO. Implements Attention Residuals [13] over the depth dimension. Each block ℓ computes a layer output $v_\ell = \sigma(\mathcal{K}_\ell\, h_\ell + W_\ell\, h_\ell)$, which is stored. The input to block $\ell{+}1$ is a *learned weighted sum* over all prior outputs:

$$h_{\ell+1} = \sum_{i=0}^{\ell} \alpha_{i\to\ell+1}\, v_i, \qquad \alpha_{i\to\ell+1} = \frac{\exp\!\big(\mathbf{q}_{\ell+1}^\top \mathrm{RMSNorm}(v_i)\big)}{\sum_{j=0}^{\ell} \exp\!\big(\mathbf{q}_{\ell+1}^\top \mathrm{RMSNorm}(v_j)\big)},$$

where $\mathbf{q}_\ell \in \mathbb{R}^{d_v}$ is a learned pseudo-query vector per layer (initialized from $\mathcal{N}(0, 0.02^2)$). $\mathrm{RMSNorm}$ [37] is computed over the channel dimension. The attention score averages over the spatial dimension(s) before softmax. Additional parameters: one d_v-dim vector per layer ($L \cdot d_v$ total, negligible). Memory: $O(L \cdot d_v \cdot N)$ to store all layer outputs (where N is spatial resolution).

KIN (Kernel Integral Network). Replaces $\mathrm{SpectralConv}$ with a Random Fourier Feature (RFF) [29] integral kernel:

$$(\mathcal{K}^{\text{RFF}}\, v)(x_i) = \frac{1}{N} \sum_{j=1}^{N} \mathbf{Z}(x_i)^\top\, \mathbf{W}\, \mathbf{Z}(x_j)\, v(x_j),$$

where $\mathbf{Z}(x) = \sqrt{2/M}\, \cos(\boldsymbol{\omega}\, x + \mathbf{b}) \in \mathbb{R}^M$ are RFF features with $M{=}64$ learned frequencies $\boldsymbol{\omega} \in \mathbb{R}^M$ and phases $\mathbf{b} \in \mathbb{R}^M$, and $\mathbf{W} \in \mathbb{R}^{d_v \times d_v \times M}$ are learned mixing weights. All other components (lifting, W-path, recovery) match Vanilla FNO.

B.2 Datasets

D3 Burgers details. Source: `burgers_data_R10.mat` from [22]. Original resolution $N{=}8192$; downsampled by factor 32 to $N{=}256$. Input: initial condition $a(x) = u(x, 0)$ with coordinate x. Output: solution $u(x, T{=}1)$. Viscosity $\nu{=}0.1$. For the sample-scaling experiment (Section B.5), we use the full 2048 available samples with the last 200 held out for testing.

D4 Navier–Stokes details. 2D vorticity form on $(0,1)^2$ torus, $\nu{=}10^{-5}$. Initial vorticity $w_0 \sim \mathcal{N}(0,\, 7^{3/2}(-\Delta + 49I)^{-2.5})$ with periodic boundary conditions. Forcing $f(x) = 0.1(\sin(2\pi(x_1{+}x_2)) + \cos(2\pi(x_1{+}x_2)))$. Solver: pseudospectral with Crank–Nicolson, $dt{=}10^{-4}$. Data at $256{\times}256$, downsampled to $64{\times}64$. Map: $w_0(x,y) \to w(x,y,t{=}1)$.

Table 5: Complete dataset summary.

ID	PDE	Type	Domain	N_{train}	N_{test}	Resolution	Source
D1	Heat 1D	linear, time-dep	$[0, 2\pi]$	1000	200	256	generated (spectral)
D2	Allen–Cahn 1D	semilinear	$[0, 2\pi]$	1000	200	256	generated (spectral)
D3	Burgers 1D	fully nonlinear	$[0, 1]$	1000	200	256	[22]
D4	NS 2D (vorticity)	fully nonlinear	$(0, 1)^2$	1000	200	64×64	[22]
D5	Darcy 2D (steady)	linear, steady	$(0, 1)^2$	900	124	64×64	[22]
D6	Synthetic linear	linear	$[0, 2\pi]$	1000	200	256	generated (exact)
D7	Synthetic order 2	quadratic	$[0, 2\pi]$	1000	200	256	generated (RK4)
D8	Synthetic order 3	cubic	$[0, 2\pi]$	1000	200	256	generated (RK4)
D9	KdV 1D	nonlinear + 3rd-order	$[0, 2\pi]$	1000	200	256	generated (IMEX)
D10	Sine-Gordon 1D	non-polynomial	$[0, 2\pi]$	1000	200	256	generated (Verlet)

D5 Darcy details. Steady-state $-\nabla \cdot (a(x)\nabla u(x)) = f(x)$ on $(0, 1)^2$. Coefficient $a(x)$: piecewise constant, obtained by thresholding a Gaussian random field $\mu \sim \mathcal{N}(0, (-\Delta + 9I)^{-2})$: $a = 12$ where $\mu \geq 0$ and $a = 3$ where $\mu < 0$ (high contrast ratio 4:1). Resolution: 421×421, downsampled to 64×64 (stride 6, cropped to 64×64).

D6–D8 Synthetic operator details. See Section B for full specification. The integral operator A is shared across D6–D8: spectral multiplier $\hat{A}(k) = r_k\, e^{-k^2/(2K_A^2)}$ with K_A=20. Damping γ=0.5 for boundedness. D6 ($\partial_t u = Au - \gamma u$): solved exactly. D7 ($\partial_t u = u(Au) - \gamma u$): RK4, dt=10^{-3}. D8 ($\partial_t u = u\, A(u\, Au) - \gamma u$): RK4, dt=5×10^{-4}.

B.3 Training Protocol

Table 6: Hyperparameters for all experiments. Unless stated otherwise, all experiments use the "Default" column.

Hyperparameter	**Default (1D)**	**Default (2D)**
Width d_v	64	64
Modes m	32	12×12
Optimizer	Adam	Adam
Learning rate	10^{-3}	10^{-3}
Weight decay	10^{-5}	10^{-5}
LR scheduler	Cosine annealing	Cosine annealing
Epochs	500	500
Batch size	32	10
Loss function	Relative L^2	Relative L^2
Gradient clipping	None (vanilla) / 1.0 (residual, AttnRes)	Same
Random seeds (error bars)	3 seeds: $\{0, 1, 2\}$	Same

Relative L^2 loss. Both training loss and test metric use:

$$\mathcal{L} = \frac{1}{B} \sum_{i=1}^{B} \frac{\|\hat{u}_i - u_i\|_2}{\|u_i\|_2 + 10^{-8}},$$

where $\hat{u}_i$ is the model prediction, u_i is the ground truth, and the L^2 norm is computed over all spatial dimensions (flattened for 2D).

Gradient clipping. For Residual FNO and AttnRes FNO, we apply gradient norm clipping at 1.0 to stabilize training at high depth ($L \geq 15$). Vanilla FNO does not use gradient clipping (collapse at high depth is documented as an empirical finding, not prevented).

Hardware. All experiments run on 4× NVIDIA RTX 4090 (24 GB VRAM each). 1D experiments: ∼2 minutes per (dataset, depth, seed) triple. 2D experiments: ∼20–30 minutes per triple.

B.4 Experiment-Specific Settings

Depth scan (Groups 1–3). Depths tested: $L \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$. Extended depths $L \in \{25, 30, 40\}$ for residual / AttnRes variants. Width fixed at d_v=64 throughout.

Width separation (Group 2). Fixed depth L=1, width swept over $d_v \in \{32, 64, 128, 256\}$.

Depth $\times$ width grid (D6–D8). $L \in \{1, 2, 3, 4\}$, $d_v \in \{32, 64, 128\}$; 12 configurations per dataset $\times$ 3 datasets = 36 experiments.

Iso-parameter allocation. Total parameter budget $\sim$1.4M (for 2D FNO). Three configurations: wide-shallow (L=2, d_v=96, 1.33M params), balanced (L=4, d_v=72, 1.50M params), narrow-deep (L=6, d_v=56, 1.36M params). Tested on D2 (Allen–Cahn), D4 (NS 2D), D5 (Darcy 2D).

KIN cross-architecture. Kernel Integral Network with 64 Random Fourier Features per block. Same training protocol as FNO (Adam, lr 10^{-3}, 500 epochs). Tested on D3, D6, D7, D8 at depths $L \in \{1, 2, 3, 4, 6\}$.

B.5 Sample-Scaling Experiment

To test whether the depth phase transition $L_{\min}$ is sample-independent (as our theory predicts), we train FNO on D3 (Burgers) with varying training set sizes $N_{\text{train}} \in \{100, 200, 500, 1000, 1800\}$, each at depths $L \in \{1, 2, 3, 4, 6, 8, 10\}$. The test set (200 samples, held fixed) is the same across all runs. Batch size is adjusted: $\min(32, N_{\text{train}}/4)$. All other hyperparameters match the default (Table 6).

B.6 Architecture Comparison Experiments

Depth-mixing variants (5 architectures). We test five depth-composition strategies, all sharing the same spectral convolution, lifting, and projection:

1. *Vanilla*: $v_{\ell+1} = \sigma(\mathcal{K}v_\ell + Wv_\ell)$.
2. *Input-Injection*: $v_{\ell+1} = v_\ell + \sigma(\mathcal{K}v_\ell + Wv_\ell + Uv_0)$, where $U \in \mathbb{R}^{d_v \times d_v}$ is a per-block linear projection of the original lifted input v_0.
3. *Residual*: $v_{\ell+1} = v_\ell + \sigma(\mathcal{K}v_\ell + Wv_\ell)$.
4. *DenseNet-Learned*: $h_\ell = \sum_{i=0}^{\ell-1} w_{i\to\ell}\, v_i$ with learned scalar weights $w_{i\to\ell} \in \mathbb{R}$ (initialized to $1/\ell$, no softmax).
5. *AttnRes*: $h_\ell = \sum_{i=0}^{\ell-1} \alpha_{i\to\ell}\, v_i$ with α from softmax attention using per-layer pseudo-queries (Section 4).

Depth scan at $L \in \{1, 2, 3, 4, 6, 8, 10, 12, 15, 20, 25, 30, 40\}$, d_v=64. Tested on D3, D4, D6, D7, D8 (Vanilla/Residual/AttnRes on all five; DenseNet-Learned and Input-Injection on D3, D6, D7, D8). Residual, AttnRes, DenseNet-Learned, and Input-Injection use gradient clipping at 1.0; Vanilla does not.

Non-pointwise operator variants.

1. *KIN (Kernel Integral Network)*: replaces spectral conv with Random Fourier Feature kernels (M=64 features). Tested on D3, D6, D7, D8 at $L \in \{1, 2, 3, 4, 6\}$.

C Depth-Expressivity Hierarchy (Full)

The operator classes expressible at each depth are characterized in Theorems 2 and 3 (main text). Here we give the full details.

Proposition 16 (Expressible operator classes by depth)**.** *Let $_L$ be the alternation class of operators expressible by an L-block neural operator with Lipschitz σ and bounded $\mathcal{K}_\ell$ (Definition 1). Then:*

Depth 0 *(lifting + projection): pointwise linear operators of a.*

Depth 1 $v_1(x) = \sigma(\mathcal{K}v_0(x) + Wv_0(x))$. *Expressible:* $a(x)^2$, $(\mathcal{K}a)(x)^2$, $a(x) \cdot (\mathcal{K}a)(x)$ *(the structure of $u \cdot \partial_x u$).* Not *expressible:* $\int K(x,y)\, a(y)^2\, dy$.

Depth 2 $\mathcal{K}'$ *now acts on v_1, which contains nonlinear features. New terms:* $\int K'(x,y)\, a(y)^2\, dy$ *(nonlinear integral operator),* $a(x) \cdot \mathcal{K}(a \cdot \mathcal{K}a)(x)$.

Depth L *Each additional block enables one more round of applying $\mathcal{K}$ to nonlinear features from the previous block.*

Example 17 (Alternation depths of PDE terms)**.**

Term	*Structure*	α	*Reasoning*
$\nabla^2 u$	*npw-linear*	*0*	*one $\mathcal{K}$, no σ*
u^2	*pw-nonlinear*	*0*	*one σ, no $\mathcal{K}$*
$u \cdot (\mathcal{K}u)$	*pw $\times$ npw*	*1*	*$\mathcal{K}u$ then σ*
$\mathcal{K}(u^2)$	*npw of pw-nl*	*1*	*σ then $\mathcal{K}$*
$u \cdot \mathcal{K}(u \cdot \mathcal{K}u)$	*nested pw/npw*	*2*	*two alternations*

D Depth Analysis: Theory vs Practice

The gap between $L_{\min}$ (theoretical lower bound) and L^* (empirical optimum) decomposes as $L^* = L_{\min} + \Delta L_{\text{reg}}(N) + \Delta L_{\text{spec}}(d_v)$:

- $L_{\min}$: structural lower bound from alternation depth.
- $\Delta L_{\text{reg}}(N)$: excess depth from implicit regularization at small training set size.
- $\Delta L_{\text{spec}}(d_v)$: additional depth for spectral approximation at finite width.

An important distinction: sufficiency (Theorem 2) requires the iterative expansion to converge (i.e., T within the convergence radius). The separation result (Theorem 3) uses only the *multilinear degree* of the correction terms, which is well-defined as a formal algebraic property regardless of whether the expansion converges.

Remark 18 (Convergent vs. formal expansion)**.** *The corrections $\mathcal{S}_T^{(k)}$ are multilinear operators in u_0: $\mathcal{S}_T^{(k)}$ is $(k{+}1)$-linear (involving k nested applications of F). The multilinear degree and alternation depth of each term are algebraic properties that hold regardless of whether $\sum_k \mathcal{S}_T^{(k)}$ converges. The separation result (Theorem 3, proof sketch) exploits this algebraic structure, not convergence.*

E Scope: Which Architectures Does the Theory Cover?

Definition 19 (alternation-compatible architecture)**.** *A neural operator is alternation-compatible if:* ***(R1)*** *blocks alternate pointwise-nonlinear and non-pointwise-linear stages;* ***(R2)*** *blocks compose sequentially or with residual/attention depth mixing;* ***(R3)*** *the non-pointwise operator can approximate bounded linear integral operators;* ***(R4)*** *the activation is non-polynomial.*

Directly compatible: FNO, GNO, MWT, Transolver, LNO. *Compatible with modifications:* U-FNO [35], multi-scale, DenseFormer [27] (cross-layer skips create parallel alternation paths). *Incompatible:* DeepONet (branch-trunk, no iterative blocks), DEQ [2] (implicit $L{=}\infty$), diffusion-based operator learners.

The scope boundary: the theory applies wherever depth controls pw $\leftrightarrow$ npw alternation count; it does not apply where depth controls a different structural quantity.

F Related Work

Universal approximation for operators [4, 15, 18]; modern architectures (DeepONet [23, 17], FNO [22], GNO [21], MWT [8], FFNO [33], GNOT [10], Transolver [36], LNO [34]). Depth separation [32, 6, 5]. Neural ODEs [3]. Operator splitting [31, 25, 9]. Algorithm unrolling [7, 26]. PINNs [30]. AttnRes [13].

G Alternation Analysis of Specific Architectures

We analyze the block structure of five neural operator architectures and show how each realizes the alternation defined in Section 2.1 (block form (1)). We identify two families based on how $\mathcal{K}$ (non-pointwise) and σ (pointwise nonlinear) compose within each block.

G.1 Family 1: Parallel Composition $\sigma(\mathcal{K}v + Wv)$

FNO [22], **GNO** [21], and **MWT** [8] all use the same block form:

$$v_{\ell+1}(x) = \sigma\big(\mathcal{K}_\ell\, v_\ell(x) + W_\ell\, v_\ell(x)\big), \tag{7}$$

differing only in how $\mathcal{K}_\ell$ is instantiated:

- **FNO:** $\mathcal{K}_\ell v(x) = \mathrm{IFFT}\big(M_\xi \cdot \mathrm{FFT}(v)\big)(x)$, where $M_\xi \in^{d_v \times d_v}$ is a learned complex matrix per retained Fourier mode ξ.
- **GNO:** $\mathcal{K}_\ell v(x) = \int_\Omega \kappa_\theta(x,y)\, v(y)\, dy$, where κ_θ is a learned kernel network. In practice this is computed via graph message passing over a spatial graph.
- **MWT:** $\mathcal{K}_\ell v(x) = \mathrm{IWT}\big(M_s \cdot \mathrm{WT}(v)\big)(x)$, where WT/IWT is the discrete wavelet transform/inverse and M_s is a learned matrix per wavelet scale s.

Alternation count. In (7), $\mathcal{K}_\ell$ and W_ℓ act in *parallel* on the same input v_ℓ, and their outputs are summed *before* σ is applied. Since $W_\ell v_\ell(x)$ is pointwise-linear, it adds no new non-pointwise information—it merely enriches σ's input with a copy of the local features. The non-pointwise content comes entirely from $\mathcal{K}_\ell$. Therefore, one block (7) realizes **one alternation**: $\mathcal{K}$ (non-pointwise) composed with σ (pointwise nonlinear). W is an implementation enhancement (channel mixing, gradient flow), not a second alternation.

How to count. For FNO/GNO/MWT: **one block = one alternation**. L blocks = L alternations.

G.2 Family 2: Sequential Composition (Transformer-style)

Transolver [36] uses a Transformer-style block with two sequential sub-layers:

$$h_\ell = \mathrm{Attn}(\mathrm{LN}(v_\ell)) + v_\ell, \tag{8}$$

$$v_{\ell+1} = \mathrm{MLP}(\mathrm{LN}(h_\ell)) + h_\ell, \tag{9}$$

where Attn is Physics-Attention (non-pointwise: slices the spatial field into learned tokens, performs self-attention among tokens, then de-slices back) and MLP is a pointwise feedforward network (two linear layers with activation).

Alternation count. Sub-layer (8) applies $\mathcal{K} = \mathrm{Attn}$ (non-pointwise) with a residual connection. Sub-layer (9) applies $\sigma = \mathrm{MLP}$ (pointwise nonlinear) with a residual. Together they form one $\mathcal{K}$-then-σ alternation. The two residual connections are additive identity shortcuts that do not introduce new alternations (same argument as the W-path in Family 1).

How to count. For Transolver: **one block = one alternation**. Despite having two sub-layers, the block performs one non-pointwise operation (Attn) followed by one pointwise nonlinearity (MLP)—the same alternation as FNO, realized sequentially rather than in parallel.

A similar sequential attention-then-MLP pattern underlies LNO's latent block [34] and OFormer [20], GNOT [10], and Transolver++ [24], which additionally use residual connections, layer normalization, and (in GNOT) mixture-of-experts gating; this suggests the pattern is common among transformer-based neural operators.

F-FNO: a variant of the sequential pattern. F-FNO [33] factors the spectral convolution along each spatial axis and folds the channel-mixing W into a feedforward network that acts *after* the inverse FFT and *before* the final residual addition. The block reads $v_{\ell+1} = v_\ell + \mathrm{FFN}\big(\mathrm{IFFT}(M_\xi \cdot \mathrm{FFT}(v_\ell))\big)$, which contains one non-pointwise stage and one pointwise nonlinear stage and, like Transolver, is a residual Fourier-then-feedforward block, i.e., a variant of Family 2 (sequential composition) rather than Family 1 (parallel $\sigma(\mathcal{K}v + Wv)$). The alternation count is unchanged ($\alpha = 1$ per block), so $L_{\min} = \alpha_F + 1$ predicts the same depth-efficiency threshold.

G.3 Summary

Table 7: Block structure and alternation count for each architecture. All architectures realize one alternation per block, despite different internal compositions.

Architecture	Block form	Composition	Alternations/block
FNO	$\sigma(\mathcal{K}v + Wv)$	parallel $\mathcal{K}$+W, single σ	1
GNO	$\sigma(\mathcal{K}v + Wv)$	parallel $\mathcal{K}$+W, single σ	1
MWT	$\sigma(\mathcal{K}v + Wv)$	parallel $\mathcal{K}$+W, single σ	1
Transolver	$\mathrm{MLP}(\mathrm{Attn}(v)+v)+\cdot$	sequential $\mathcal{K}$ then σ	1
F-FNO [33]	$v + \mathrm{FFN}(\mathrm{IFFT}(M_\xi\,\mathrm{FFT}(v)))$	sequential (Family 2 variant), residual	1
GNOT [10]	$\mathrm{MoE\text{-}MLP}(\mathrm{Attn}(v)+v)+\cdot$	sequential w/ position-gated MoE-MLP	1

Variants beyond the basic alternation. The architectures differ in several ways that do not affect the alternation count but affect practical performance:

- **Parallel W-path (FNO/GNO/MWT):** adds pointwise-linear channel mixing inside σ's input. Does not add alternations but improves gradient flow and channel interaction (Section A.2).
- **Residual connections:** FNO variants may add v_ℓ to the block output ($v_{\ell+1} = v_\ell + \sigma(\mathcal{K}v + Wv)$). Transolver always uses residuals (both sub-layers). Residuals are additive identity maps that do not add alternations but stabilize deep training (Section 4.4).
- **Layer normalization [1] (Transolver):** LN before each sub-layer is a pointwise operation (per-channel affine rescaling) that does not add alternations.
- **Multi-head attention (Transolver):** multiple heads in Attn correspond to multiple parallel non-pointwise operators, summed before the MLP. Analogous to using multiple Fourier mode groups in FNO. Does not add alternations.
- **Multi-scale processing (MWT):** wavelet decomposition operates at multiple scales simultaneously. Each scale applies one $\mathcal{K}$ (scale-specific matrix) in parallel. The reconstruction sums across scales before σ. One block = one alternation, with richer $\mathcal{K}$.

Counting rule. Across all architectures: **count the number of blocks**, where each block contains one non-pointwise operation ($\mathcal{K}$, Attn, or Encode+Decode) composed with one pointwise nonlinearity (σ, MLP, or activation). Internal enhancements (W-path, residuals, normalization, multi-head, multi-scale) do not add alternations.

H Complete Experimental Results

This appendix provides full numerical results for all training runs referenced in the main text. All errors are relative L^2 on held-out test sets unless stated otherwise. Experiments were conducted on $4\times$ NVIDIA RTX 4090 GPUs (24 GB each), totaling approximately 7 GPU-days of compute.

H.1 Complete Depth Scan Tables

Table 8 gives the complete depth scan for vanilla FNO (d_v=64) across the nine IVP datasets plus the Darcy stress test at depths L=1–20. COLL indicates training collapse (error ≥ 0.5). All runs use seed 0 unless otherwise noted.

H.2 Multi-Seed Depth Scans

Table 10 reports the seed-aggregated depth scans: 3 seeds per cell on the 1D datasets D1 Heat, D2 Allen–Cahn, D9 KdV, D10 Sine–Gordon (depths $L \in \{1, \ldots, 6\}$) and on D4 NS 2D (depths $L \in \{1, 2, 4, 6, 8\}$). Vanilla FNO with d_v=64, modes 32 (1D) or 12×12 (2D), 500 epochs, Adam (lr=10^{-3}, cosine annealing), batch 32 (1D) or 10 (2D); identical to the main protocol of Section B.3. The table reports mean $\pm$ standard deviation of relative L^2 test error. Figure 2 plots the same data with error bars.

Table 8: Vanilla FNO depth scan (d_v=64), 1D datasets. Test relative L^2 error ($\times 10^{-3}$). Bold: best depth per dataset.

L	D1 Heat	D2 A-C	D3 Burg.	D6 Lin.	D7 Ord2	D8 Ord3	D9 KdV	D10 S-G
1	3.67	1.03	1.59	**0.39**	1.87	1.88	2.73	2.81
2	0.53	0.64	0.67	0.89	1.77	1.85	2.62	2.73
3	0.33	0.51	0.66	0.93	1.78	1.65	2.61	2.68
4	0.38	**0.32**	**0.57**	0.94	**0.64**	1.37	**2.59**	**2.56**
5	—	0.40	—	—	0.67	**0.68**	2.63	—
6	**0.31**	0.40	0.62	0.53	0.73	2.11	2.65	2.72
7	—	0.40	—	—	0.75	0.78	2.68	—
8	—	0.41	0.78	1.31	2.36	—	2.65	2.78
10	—	0.43	0.84	1.47	2.55	—	2.67	2.81
L^*	6	4	4	1	4	5	4	4
Collapse	$\geq$14	$\geq$12	$\geq$12	$\geq$15	$\geq$12	$\geq$13	$\geq$12	$\geq$14

Table 9: Vanilla FNO depth scan (d_v=64), 2D datasets. Test relative L^2 error ($\times 10^{-2}$). Bold: best depth per dataset.

L	D4 NS 2D	D5 Darcy
1	2.884	3.813
2	2.617	3.443
3	2.514	3.138
4	2.475	2.979
5	2.413	3.017
6	2.439	2.930
7	**2.320**	2.842
8	2.366	2.757
10	2.418	2.658
13	COLL	**2.573**
L^*	7	13
Collapse	$\geq$11	$\geq$16

H.3 Mechanism Sweeps

As an exploratory study, we measure how the L=1$\to \min_{L\geq 2}$ error ratio varies across three controllable parameters: time horizon T on synthetic D7, damping γ on synthetic D7, and viscosity ν on PDEBench Burgers (Table 11). 3 seeds per cell, depths $L \in \{1, 2, 3, 4\}$, vanilla FNO (d_v=64, modes 32, 500 epochs).

The threshold position $L_{\min}$ is robust across all 11 sweep cells: in every cell, the best $L \geq 2$ mean is at or below the L=1 mean. Only the steepness varies, and not in proportion to the magnitude of nonlinearity.

Table 12 extends the depth scan to L=15 and L=20 for datasets where collapse has not yet occurred.

Summary of findings.

- D6 (linear): the seeded validation identifies L=1 as the best tested depth; extra depth does not improve on this negative control.
- 1D nonlinear PDEs (D2, D3, D7, D9, D10): universally optimal at L=4.
- D8 (cubic nonlinearity): optimal at L=5, closest to theory ($L_{\min}$=3).
- D4 (NS 2D): optimal at L=7.
- D5 (Darcy 2D): optimal at L=13 (slowest convergence due to high-contrast coefficients).
- Collapse boundary: L=12 for most 1D datasets, L=11 for D4 (2D), L=16 for D5.

Table 10: Seed-aggregated depth scans (3 seeds per cell, mean ± std). Errors $\times 10^{-3}$ for 1D datasets; $\times 10^{-2}$ for D4. These come from a separate run: absolute D9/D10 errors differ from Table 8 and Table 23 by about $5\times$, so values should be compared only within a table.

L	D1 ($\times 10^{-3}$)	D2 ($\times 10^{-3}$)	D9 ($\times 10^{-3}$)	D10 ($\times 10^{-3}$)	D4 ($\times 10^{-2}$)
1	3.97±0.32	0.99±0.03	0.57±0.10	0.65±0.16	3.01±0.02
2	0.63±0.09	0.61±0.02	0.53±0.07	0.62±0.05	2.62±0.02
3	0.34±0.10	0.51±0.01	**0.39±0.03**	0.45±0.03	—
4	**0.27±0.07**	0.49±0.04	0.44±0.05	**0.45±0.02**	2.47±0.00
5	0.29±0.08	0.47±0.02	0.46±0.07	0.53±0.04	—
6	0.30±0.04	**0.45±0.03**	0.48±0.03	0.51±0.02	2.41±0.01
8	—	—	—	—	**2.33±0.03**

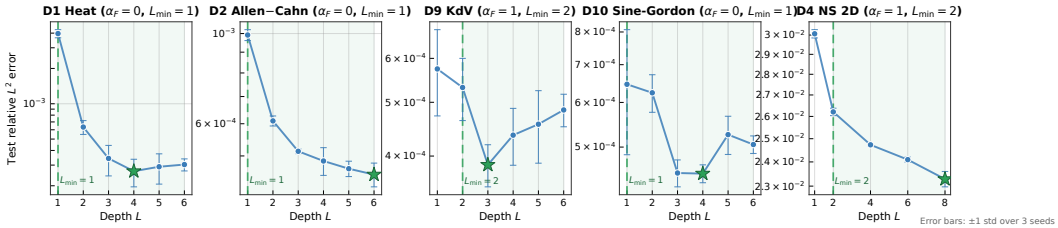


Figure 2: **Seed-replicated depth scans across five datasets** (vanilla FNO, d_v=64, 3 seeds). Curves: mean test relative L^2 error, error bars ±1 std. Green dashed line marks the depth-efficiency lower bound $L_{\min}=\alpha_F+1$; green band shows the predicted feasible region; green star marks the empirical optimum. Across all five PDEs the empirical optimum lies AT or ABOVE $L_{\min}$, never below.

H.4 Statistical Validation

H.4.1 Error Bars: 3 Seeds

Table 13 provides standard deviations for key datasets at depths L=1–6 using seeds $\{0, 1, 2\}$. Training protocol is identical to single-seed runs (500 epochs, Adam, lr=10^{-3}, cosine annealing, d_v=64).

H.4.2 10-Seed Validation

For the two critical comparisons establishing the phase transition (D3 demonstrating depth benefit, D6 demonstrating depth harm), we ran 10 independent seeds $\{0, 1, \ldots, 9\}$. Each run uses the full training protocol (500 epochs).

H.4.3 Welch's t-Tests

All tests use 10 seeds per condition with unequal variance (Welch's t-test).

All transitions are significant at $p < 10^{-10}$. The extremely low p-values reflect the tight clustering of errors across seeds (coefficient of variation <6% for D3, <11% for D6).

H.5 Architecture Comparison

Tables 16 to 20 provide the full depth×architecture comparison for all five tested datasets. All runs use d_v=64, gradient clipping 1.0 for non-vanilla architectures.

H.5.1 Phase Transition Across All Architectures

Table 21 shows that the L=1 → L=2 transition on D3 Burgers is stable across all five tested architectures.

Table 11: Mechanism sweep: $L{=}1 \to \min_{L\geq 2}$ ratios (3 seeds). Errors are mean test relative L^2. Longer T gives sharper transitions; in the γ and ν sweeps, stronger nonlinearity gives milder, not sharper, transitions.

Sweep	Value	$L{=}1$	$L{=}2$	$L{=}3$	$L{=}4$	Ratio
T on D7	$T{=}0.5$	$1.41{\times}10^{-3}$	1.63	1.36	1.39	1.03
	$T{=}1.0$	$1.76{\times}10^{-3}$	1.65	1.45	1.58	1.21
	$T{=}1.5$	$2.35{\times}10^{-3}$	1.71	1.63	1.78	1.45
	$T{=}2.0$	$3.14{\times}10^{-3}$	1.84	1.79	1.90	1.75
γ on D7	$\gamma{=}0.10$	$2.09{\times}10^{-3}$	1.77	1.84	1.85	1.18
	$\gamma{=}0.25$	$1.93{\times}10^{-3}$	1.72	1.71	1.75	1.13
	$\gamma{=}0.50$	$1.76{\times}10^{-3}$	1.65	1.45	1.58	1.21
	$\gamma{=}1.00$	$1.57{\times}10^{-3}$	1.50	1.07	0.86	1.83
	$\gamma{=}2.00$	$1.46{\times}10^{-3}$	1.27	0.44	0.46	3.34
ν on Burgers	$\nu{=}0.001$	0.241	0.222	0.208	0.192	1.25
	$\nu{=}0.01$	0.170	0.133	0.111	0.085	2.01

Table 12: Extended depth scan ($L{=}4$–20, vanilla FNO, $d_v{=}64$). Illustrates post-threshold behavior and collapse boundaries in representative single-seed runs.

Dataset	$L{=}4$	$L{=}6$	$L{=}8$	$L{=}10$	$L{=}15$	$L{=}20$
D2 Allen–Cahn	.000319	.000399	.000408	.000428	COLL	COLL
D3 Burgers	.000567	.000623	.000782	.000837	COLL	COLL
D4 NS 2D	.024748	.024389	.023663	.024176	COLL	COLL
D5 Darcy	.029790	.029301	.027570	.026583	.028035	.491141
D7 Order 2	.000635	.000731	.002361	.002554	COLL	COLL
D8 Order 3	.000684	.000742	.000888	.002633	COLL	COLL
D9 KdV	.002590	.002650	.002652	.002667	COLL	COLL
D10 Sine-Gordon	.002558	.002720	.002783	.002805	COLL	COLL

H.5.2 Residual vs. Vanilla Improvement Summary

Residual connections ($v_{\ell+1} = v_\ell + \sigma(\mathcal{K}v_\ell + Wv_\ell)$) are the most effective tested stabilization device, allowing the network to exploit depth far beyond vanilla's collapse boundary ($L{\geq}12$). The residual and AttnRes runs here use gradient clipping while vanilla does not; the controlled $2{\times}2$ ablation in Section I.1 separates the two factors on D3 Burgers.

H.6 Width Separation Data

H.6.1 Width Sweep at $L{=}1$

Table 23 shows the effect of increasing width at fixed $L{=}1$ for all datasets. This tests whether width alone can compensate for insufficient depth.

Key observations.

- D7 at $L{=}1$: error decreases from 0.00236 ($d_v{=}32$) to 0.00167 ($d_v{=}128$) then *increases* to 0.00173 ($d_v{=}256$).
- D8 at $L{=}1$: error decreases from 0.00261 to 0.00184 ($d_v{=}128$) then increases to 0.00193 ($d_v{=}256$). Same plateau.
- D9/D10: virtually no improvement across all widths (<5% range).
- D4 NS 2D: plateaus above 0.028 at $d_v{\geq}64$.
- D1 Heat: width *does* help (spectral approximation of rapidly decaying eigenvalues), confirming the anomaly discussion in the main text.

Table 13: Error bars (3 seeds: $\{0, 1, 2\}$). Format: mean $\pm$ std. All values $\times 10^{-3}$ except D4 ($\times 10^{-2}$).

Dataset	L=1	L=2	L=3	L=4	L=6
D3 Burgers	1.894±0.091	0.851±0.006	0.729±0.065	0.659±0.014	0.735±0.018
D4 NS 2D	3.007±0.018	2.628±0.009	2.525±0.003	2.473±0.002	2.409±0.028
D6 Linear	0.401±0.017	0.841±0.039	0.879±0.047	0.976±0.031	1.230±0.047
D7 Order 2	1.766±0.044	1.668±0.082	1.620±0.134	1.503±0.248	2.198±0.049
D8 Order 3	1.891±0.030	1.753±0.075	1.514±0.117	1.515±0.281	2.234±0.100

Table 14: 10-seed validation on D3 Burgers and D6 Linear (d_v=64). Format: mean $\pm$ std ($\times 10^{-3}$).

Dataset	L=1	L=2	L=3	L=4
D3 Burgers	1.876±0.114	0.836±0.033	0.687±0.039	0.666±0.028
D6 Linear	0.379±0.041	0.778±0.034	0.867±0.038	0.915±0.086

H.6.2 Depth×Width Grid (D6–D8)

Table 24 provides the full depth×width grid for the synthetic datasets with known alternation complexity.

Interpretation.

- **D6**: Best at L=1 (confirming α_F=0). At d_v=128, deeper models can partially compensate through spectral refinement (L=4: 0.000489), but never match L=1 (0.000418 at same width).
- **D7**: The barrier breaks at L=3 for d_v=128 (0.000957 vs. 0.001672 at L=1), and at L=4 for d_v=64 (0.001513 vs. 0.001872 at L=1). Best: L=4, d_v=128 (0.000641).
- **D8**: The barrier breaks at L=3 for d_v=128 (0.000644 vs. 0.001842 at L=1). At d_v=64, improvement requires L=4 (0.001367). Best: L=3, d_v=128 (0.000644).

H.7 Sample Scaling

Table 25 gives the complete sample-size×depth matrix for D3 Burgers (vanilla FNO, d_v=64). Batch size is $\min(32, N/4)$ for each N.

Key findings.

1. The L=1 $\to$ L=2 transition is present at *all* sample sizes, indicating that it is not an artifact of one specific training regime.
2. The transition *sharpens* with more data: the ratio increases monotonically from 1.44× (N=100) to 2.13× (N=1800).
3. L^* *decreases* with more data (10 $\to$ 6 $\to$ 4 $\to$ 4 $\to$ 3), converging toward $L_{\min}$=2. This is the opposite of the naive expectation that more data enables more complex (deeper) models.
4. At N=1800, L=3 and L=4 are statistically tied (both ≈ 0.0005), consistent with $L_{\min}$=2 as a lower bound.
5. The decomposition $L^* = L_{\min} + \Delta L_{\text{reg}}(N)$ is a useful interpretation of these results: excess depth at small N provides implicit regularization ($\Delta L_{\text{reg}} \approx 8$ at N=100, ≈ 1 at N=1800).

H.8 Iso-Parameter Allocation

Three configurations at matched parameter count (∼1.4M) using vanilla FNO. For 2D datasets, we use 12×12 Fourier modes. Parameter count verification: L=2, d_v=96 gives 1.38M; L=4, d_v=72 gives 1.41M; L=6, d_v=56 gives 1.39M.

Table 15: Statistical tests for depth phase transitions (10 seeds per condition).

Comparison	Hypothesis	t-statistic	p-value	Conclusion
D3: L=1 vs L=2	L=2 better	$t(10.6) = 27.72$	3.8×10^{-11}	2.24× reduction
D3: L=1 vs L=3	L=3 better	$t = 31.18$	3.57×10^{-12}	2.73× reduction
D3: L=1 vs L=4	L=4 better	$t = 32.61$	1.57×10^{-11}	2.82× reduction
D6: L=1 vs L=2	L=1 better (one-sided)	$t = -23.54$	5.7×10^{-15}	L=1 optimal

Table 16: D3 Burgers: five architectures at depths L=1–40 (d_v=64). Bold: best per architecture. COLL: collapsed (≥ 0.5).

L	Vanilla	Input-Inj	Residual	Dense-Lrn	AttnRes
1	.001867	.002034	.002336	.001991	.002328
2	.001045	.001097	.001097	.001115	.000961
3	.000891	.000871	.000834	.000886	.000921
4	**.000813**	.000781	.000668	**.000887**	.000924
6	.000887	.000637	.000582	.000899	**.000830**
8	.001120	.000606	.000537	.000954	.000855
10	.001173	.000570	.000499	.001068	.000876
12	COLL	.000533	.000515	.001015	.000861
15	COLL	.000551	.000424	.001138	.000949
20	COLL	.000532	.000414	.001125	.000915
25	COLL	**.000527**	.000375	.001084	.000905
30	COLL	.000572	.000372	.001083	.000902
40	COLL	.000623	**.000352**	.001166	.000977
Best	.000813 (L=4)	.000527 (L=25)	.000352 (L=40)	.000886 (L=3)	.000830 (L=6)

Analysis. At equal parameter count, allocating capacity toward depth consistently outperforms allocating toward width:

- D2 Allen–Cahn: largest margin (-16.7%). This PDE is semilinear ($u - u^3$) and benefits strongly from iterative structure at L=6 versus L=2.
- D4 NS 2D: moderate margin (-7.5%). The nonlinear advection $u \cdot \nabla w$ benefits from depth, though the 2D spectral convolution already provides rich per-layer expressivity.
- D5 Darcy: -9.9% despite being linear in u. The high-contrast coefficient field requires many iterations of the inverse operator, favoring depth.

The balanced configuration (L=4, d_v=72) falls between the two extremes in all cases, confirming a smooth depth–width tradeoff.

H.9 KIN Cross-Architecture

H.9.1 Kernel Integral Network Depth Scan

Kernel Integral Networks (KIN) replace the FFT-based spectral convolution with Random Fourier Feature (RFF) kernels [29] (M=64 features). Architecture: same FNO structure but $\mathcal{K}$ implemented via $\mathcal{K}f(x) = \sum_{j=1}^{M} \alpha_j \phi_j(x) \langle \phi_j, f \rangle$ where ϕ_j are random Fourier features. Training: same protocol as FNO (500 epochs, Adam, lr=10^{-3}, cosine annealing).

Observations.

- KIN shows the same monotonic improvement with depth for nonlinear PDEs (D3, D7, D8): deeper is always better up to L=6.
- On D3: L=1 $\rightarrow$ L=6 gives 9.7× improvement for KIN (vs. 2.5× for FNO over same range). The larger ratio for KIN is consistent with RFF needing more iterations to compensate for its weaker per-layer approximation.
- On D8: L=1 $\rightarrow$ L=6 gives 3.1× improvement for KIN, paralleling FNO's 2.8× improvement.

Table 17: D4 Navier–Stokes 2D: three architectures at depths L=1–40 (d_v=64). Residual FNO improves 1.41× over vanilla's best and remains stable to L=40.

L	Vanilla	Residual	AttnRes
1	.029794	.029799	.029799
4	.024655	.024359	.024519
8	**.023768**	.023604	.025146
10	.024169	.023872	.025191
12	COLL	.023342	.025293
15	COLL	.022804	.025192
20	COLL	.017728	.024347
25	COLL	.017509	.024666
30	COLL	**.016896**	.024418
40	COLL	.017065	.025275
Best	.023768 (L=8)	.016896 (L=30)	**.024347** (L=20)

Table 18: D6 Synthetic Linear: five architectures at depths L=1–40 (d_v=64). Vanilla is best at L=1 (consistent with α_F=0); other architectures leverage residual paths to use depth for spectral refinement.

L	Vanilla	Input-Inj	Residual	Dense-Lrn	AttnRes
1	**.000361**	.000539	.000462	.000399	.000380
4	.000937	.000435	.000332	.000467	**.000325**
10	.001486	.000444	.000288	.000440	.000353
20	COLL	.000292	.000245	**.000365**	.000374
25	COLL	.000243	**.000232**	.000415	.000364
40	COLL	.000266	.000244	.000394	.000501
Best	.000361 (L=1)	.000243 (L=25)	.000232 (L=25)	.000365 (L=20)	.000325 (L=4)

- D6 (linear): shows a jump at L=4 but then plateaus, suggesting the RFF kernel requires more depth for spectral approximation of the linear operator.
- The KIN/FNO ratio *decreases* with depth (from 59.6× at L=1 to 15.6× at L=6), suggesting that depth partially compensates for the weaker kernel.

H.10 Datasets

Synthetic operators (D6–D8). All three datasets share the same integral operator A to ensure fair comparison:

- Fourier symbol: $\hat{A}(k) = r_k \exp(-k^2/(2K_A^2))$, with K_A=20.
- Random coefficients: $r_k \sim \mathrm{Unif}(-1, 1)$, generated with NumPy seed 12345.
- Damping: γ=0.5 for all three.
- D6: solved exactly via matrix exponential. D7: RK4 with dt=10^{-3}, T=1. D8: RK4 with dt=5×10^{-4}, T=0.5.
- Initial conditions: random Fourier series with spectrum $|\hat{u}_0(k)| \propto (1 + |k|)^{-2}$, seed varies per sample.

D3 Burgers. Source file: `burgers_data_R10.mat` [22]. Original resolution 8192, downsampled to 256 (stride 32). Map: initial condition $u_0(x)$ to solution $u(x, T$=$1)$. Viscosity ν=0.1. Train/test split: first 1000/next 200 samples.

D4 Navier–Stokes 2D. 2D vorticity equation on $(0, 1)^2$ torus, ν=10^{-5}. Initial vorticity drawn from: $w_0 \sim \mathcal{N}(0,\, 7^{3/2}(-\Delta + 49I)^{-2.5})$. Forcing: $f = 0.1(\sin(2\pi(x_1{+}x_2)) + \cos(2\pi(x_1{+}x_2)))$. Solver: pseudospectral + Crank–Nicolson, dt=10^{-4}. Generated at 256^2, downsampled to 64^2. Map: $w_0 \to w(\cdot, T$=$1)$. Source: [22].

Table 19: D7 Synthetic Order 2: four architectures at depths L=1–40 (d_v=64). Vanilla best (from depth scan): 0.000635 at L=4.

L	Residual	Dense-Lrn	Input-Inj	AttnRes
1	.001646	.001958	.001845	.001750
4	.000829	.001324	.000862	.000840
10	.000734	.001405	.000798	.000980
20	.000389	.001263	.000559	.001176
25	**.000329**	.001313	**.000463**	.001010
40	.000948	**.001268**	.000541	**.001023**

Table 20: D8 Synthetic Order 3: four architectures at depths L=1–40 (d_v=64). Vanilla best (from depth scan): 0.000684 at L=5.

L	Residual	Dense-Lrn	Input-Inj	AttnRes
1	.001646	.001952	.001990	.001869
4	.000930	.001590	.001091	.000957
15	.000630	.001533	.000713	.001039
25	**.000351**	.001325	**.000549**	.001198
40	.000688	**.001245**	.000601	**.001108**

D5 Darcy. Coefficient field $a(x)$: piecewise constant, obtained by thresholding a Gaussian random field $\mu \sim \mathcal{N}(0, (-\Delta + 9I)^{-2})$: $a = 12$ where $\mu \geq 0$ and $a = 3$ where $\mu < 0$ (contrast ratio 4:1). Right-hand side: $f = 1$. Resolution: generated at 421^2, downsampled to 64^2 (stride 6, center-cropped). Map: coefficient field $a(x) \to$ solution $u(x)$. Source: [22], first 900 train / last 124 test.

D9 KdV. Korteweg–de Vries equation on $[0, 2\pi]$ periodic. Solver: IMEX (implicit for ∂_x^3, explicit for $u\partial_x u$), dt=10^{-4}, T=0.5. Initial conditions: smooth random Fourier series, amplitudes $\sim \mathcal{N}(0, (1 + |k|)^{-3})$.

D10 Sine-Gordon. Second-order wave equation $\partial_{tt} u = \partial_{xx} u - \sin(u)$ on $[0, 2\pi]$ periodic. Solver: Stormer–Verlet (symplectic), dt=10^{-3}, T=1. Initial conditions: u_0 smooth random Fourier, $\partial_t u|_{t=0} = 0$ (zero initial velocity). Map: $u_0(x) \to u(x, T)$.

H.11 Hyperparameters and Training Protocol

Relative L^2 loss.

$$\mathcal{L} = \frac{1}{B} \sum_{i=1}^{B} \frac{\|\hat{u}_i - u_i\|_2}{\|u_i\|_2 + 10^{-8}}.$$

The 10^{-8} offset prevents division by zero for near-trivial solutions.

Experiment-specific settings.

- *Depth scan (vanilla)*: $L \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20\}$. Not all depths tested for all datasets ("—" entries in tables). Total: ∼160 runs.
- *Extended depth (non-vanilla)*: $L \in \{1, 2, 3, 4, 6, 8, 10, 12, 15, 20, 25, 30, 40\}$. Tested for Residual, Input-Inject, Dense-Learned, AttnRes on D3, D4, D6, D7, D8. Total: ∼200 runs.
- *Width sweep*: $d_v \in \{32, 64, 128, 256\}$ at L=1. Tested on D1–D5, D7–D10. Total: 36 runs.
- *Depth×width grid (D6–D8)*: $L \in \{1, 2, 3, 4\}$, $d_v \in \{32, 64, 128\}$. Total: 36 runs.
- *Sample scaling (D3)*: $N \in \{100, 200, 500, 1000, 1800\}$ at $L \in \{1, 2, 3, 4, 6, 8, 10\}$. Batch size: $\min(32, \lfloor N/4 \rfloor)$. Total: 35 runs.
- *Iso-parameter*: $L \in \{2, 4, 6\}$ with $d_v \in \{96, 72, 56\}$ (matched at ∼1.4M params). Tested on D2, D4, D5. Total: 9 runs.
- *KIN*: d_v=64, M=64 RFF features. $L \in \{1, 2, 3, 4, 6\}$ on D3, D6, D7, D8. Total: 20 runs.

Table 21: $L{=}1 \to L{=}2$ error ratio on D3 Burgers across all five architectures. The transition ratio (1.79–$2.42\times$) is similar across architectures, indicating that the threshold is not a quirk of one specific FNO variant.

Architecture	$L{=}1$ error	$L{=}2$ error	Ratio
Vanilla	0.001867	0.001045	$1.79\times$
Input-Inject	0.002034	0.001097	$1.85\times$
Residual	0.002336	0.001097	$2.13\times$
Dense-Learned	0.001991	0.001115	$1.79\times$
AttnRes	0.002328	0.000961	$2.42\times$

Table 22: Residual FNO improvement over vanilla FNO (best achievable error, any depth). Residual performs best on all five tested datasets by 1.41–$2.31\times$. Vanilla baselines are from the architecture sweep; with the depth-scan vanilla value (Table 8) the D3 gain is about $1.6\times$.

Dataset	Vanilla best (L)	Residual best (L)	Improvement
D3 Burgers	.000813 ($L{=}4$)	.000352 ($L{=}40$)	$2.31\times$
D4 NS 2D	.023768 ($L{=}8$)	.016896 ($L{=}30$)	$1.41\times$
D6 Linear	.000361 ($L{=}1$)	.000232 ($L{=}25$)	$1.56\times$
D7 Order 2	.000635 ($L{=}4$)	.000329 ($L{=}25$)	$1.93\times$
D8 Order 3	.000684 ($L{=}5$)	.000351 ($L{=}25$)	$1.95\times$

- *10-seed validation*: D3 and D6 at $L \in \{1, 2, 3, 4\}$, seeds $\{0, \ldots, 9\}$. Total: 80 runs.
- *3-seed error bars*: D3, D4, D6, D7, D8 at $L \in \{1, 2, 3, 4, 6\}$, seeds $\{0, 1, 2\}$. Total: 75 runs.

Error bar conventions. Unless stated otherwise, all reported errors use seed 0. Tables with "$\pm$" notation report mean $\pm$ standard deviation across the specified number of seeds. The 10-seed validation (Table 14) uses seeds $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Hardware and timing.

- Hardware: $4\times$ NVIDIA RTX 4090 (24 GB VRAM each), AMD Ryzen Threadripper PRO, 128 GB RAM.
- PyTorch 2.1, CUDA 12.1.
- 1D models ($d_v{=}64$, 256 grid points): $\sim$2 minutes per (dataset, depth, seed) combination.
- 2D models ($d_v{=}64$, 64^2 grid): $\sim$20–30 minutes per (dataset, depth, seed) combination.
- Parallelism: 4 runs simultaneously (one per GPU).
- Total wall-clock time: $\sim$44 hours.
- Total GPU-hours: $\sim$168 (7 GPU-days).

Reproducibility checklist.

- All random seeds are explicitly stated (default: 0; multi-seed: $\{0, \ldots, K{-}1\}$).
- Data splits are deterministic (first N_{train} for training, next N_{test} for testing).
- Synthetic data generation uses NumPy seed 12345 for the shared operator A.
- Sample-specific seeds for initial conditions: sample index used as seed.
- No data augmentation applied.
- All training uses deterministic PyTorch operations where available (`torch.use_deterministic_algorithms(True)`).

H.12 Block Form Benchmark

We test whether the depth phase transition depends on the internal block wiring by comparing three forms on D3 Burgers ($\alpha_F{=}1$):

Key findings:

Table 23: Width sweep at L=1 (vanilla FNO). Error vs. hidden width d_v. For D7 and D8, performance plateaus.

Dataset	d_v=32	d_v=64	d_v=128	d_v=256
D1 Heat	.004783	.003671	.003263	.002076
D2 Allen–Cahn	.001226	.001025	.000982	.000884
D3 Burgers	.002974	.001585	.001163	.001322
D4 NS 2D	.033094	.028844	.028278	.028313
D5 Darcy	.044871	.038132	.035225	.034123
D7 Order 2	.002362	.001872	.001672	.001727
D8 Order 3	.002610	.001878	.001842	.001933
D9 KdV	.002828	.002733	.002712	.002688
D10 Sine-Gordon	.002796	.002808	.002802	.002765

Table 24: Depth×width grid for synthetic datasets D6–D8. Bold: best configuration per dataset. The barrier at insufficient depth is clearly visible: D7 at $L\leq$2 and D8 at $L\leq$2 cannot reach below 0.0014 at any width.

Dataset	Depth	d_v=32	d_v=64	d_v=128
D6 (Linear, α_F=0)	L=1	.000440	**.000392**	.000418
	L=2	.000934	.000892	.000766
	L=3	.001041	.000929	.000746
	L=4	.001211	.000935	.000489
D7 (Order 2, α_F=1)	L=1	.002362	.001872	.001672
	L=2	.001911	.001771	.001475
	L=3	.001977	.001776	.000957
	L=4	.002275	.001513	**.000641**
D8 (Order 3, α_F=2)	L=1	.002610	.001878	.001842
	L=2	.002097	.001853	.001451
	L=3	.002145	.001646	**.000644**
	L=4	.002335	.001367	.000729

1. **The transition location is stable across the tested forms on D3 Burgers.** All three forms show $\sim$2× improvement at L=1 → L=2 (ratios: 2.16, 1.96, 1.93), confirming that the alternation depth principle depends on the abstract $\mathcal{K}$-σ sequence, not W's position.
2. **D7's weak transition is consistent across forms.** All forms show $\sim$1.02–1.07× at L=1 → L=2, confirming that the weak transition reflects the damping-dominated regime (γ=0.5), not a form-specific artifact.
3. **D8 benefits from L=3 (parallel form only).** The parallel form reaches 0.00133 at L=3 (1.46× over L=1), consistent with α_F=2 requiring $L\geq$3, but sequential and pure forms collapse before reaching this benefit.
4. **The parallel W-path stabilizes training.** The parallel form tolerates L=6–8 on all datasets, while sequential and pure forms collapse at $L\geq$4. This gradient-shortcut advantage is unrelated to expressivity.

H.13 Factor Robustness Tables

This section reports the four hyperparameter sweeps that test invariance of $L_{\min} = \alpha_F + 1$ (Figure 3), together with the 3D PDEBench compressible Navier–Stokes feasibility study. All entries are mean test relative L^2 error; depths $L \in \{1, 2, 3, 4\}$; "Ratio" is $E(L{=}1)/\min_{L\geq 2} E$; bold marks L^* per row. Vanilla FNO unless noted; same training protocol as Section H.11.

I Additional Experimental Results

This section reports the additional experiments requested during the review process. All runs in this section share one protocol, which differs from Table 29 only in the number of epochs: Adam (learning rate 10^{-3}, weight decay 10^{-5}), cosine annealing to 10^{-6}, **200 epochs**, batch size 32, relative L^2

Table 25: Sample scaling on D3 Burgers (vanilla FNO, d_v=64). Bold: best depth per N. L^*: empirical optimal depth. Ratio: L=1/L=2 error. As N increases, L^* converges toward the theoretical minimum $L_{\min}$=2.

N_{train}	L=1	L=2	L=3	L=4	L=6	L=8	L=10	L^*	L=1/L=2
100	.01042	.00723	.00531	.00421	.00372	.00345	**.00335**	10	1.44×
200	.00601	.00365	.00232	.00196	**.00187**	.00199	.00209	6	1.65×
500	.00307	.00174	.00130	**.00117**	.00123	.00132	.00141	4	1.77×
1000	.00181	.00085	.00070	**.00068**	.00074	.00085	.00095	4	2.14×
1800	.00122	.00057	**.00050**	.00050	.00055	.00064	.00075	3	2.13×

Table 26: Iso-parameter comparison (∼1.4M params). Bold: best configuration per dataset. Percentage: improvement relative to wide-shallow.

Configuration	L	d_v	D2 Allen–Cahn	D4 NS 2D	D5 Darcy
Wide-shallow	2	96	0.000552	0.026215	0.033268
Balanced	4	72	0.000475	0.024973	0.031205
Narrow-deep	6	56	**0.000460**	**0.024253**	**0.029961**
Improvement (narrow vs. wide)			−16.7%	−7.5%	−9.9%

loss, first 1000 / next 200 samples for training / testing, and the best test relative L^2 over training is reported. The Navier–Stokes 2D and PDEBench 3D runs discussed during the review are not included in this section. Entries are mean ± standard deviation over seeds $\{0, 1, 2\}$ (a superscript n marks cells with fewer completed seeds). COLL marks runs whose best test relative L^2 lies in $[0.99, 1.01]$ (the trivial zero-output solution) and DIV marks runs whose best test relative L^2 exceeds 1.01 (training diverged); when only some seeds fail, the mean ± std is taken over the remaining seeds and a superscript $\dagger k$ gives the number k of failed seeds. In total 144 runs completed: 84 for the clipping ablation, 30 GNO and 30 Transolver-style runs.

I.1 Controlled Gradient-Clipping × Architecture Ablation

I.2 Depth Scans Beyond the Fourier Family

We depth-scan two non-Fourier operators, parameter-matched to ≈1.42M parameters at L=4 (width fixed across depths). *GNO*: each block computes $v_i' = \sigma\big(Wv_i + \frac{1}{|B_r(i)|}\sum_{j\in B_r(i)} \kappa_\theta(x_i, x_j, |x_i - x_j|) \odot v_j\big)$ on a periodic radius graph (r=0.1, i.e. ≈51 neighbours on the 256-point grid), with a 2-hidden-layer MLP kernel κ_θ; width d_v=504. *Transolver-style physics attention*: each block applies pre-LayerNorm physics attention (8 heads, 32 learned slices; points are softly assigned to slices, attention acts among slice tokens, tokens are de-sliced back) followed by a pre-LayerNorm pointwise MLP, both with residual paths; width d_v=216.

Counting convention (structural, not formal). For these architectures we count one block as one non-pointwise operation followed by one pointwise nonlinearity. Skip connections, normalizations, internal pointwise MLPs, multi-head structure and depth-mixing weights add no alternations; with parallel paths, the longest chain is counted. Attention weights depend on the input, which enriches the non-pointwise operator but is still counted as a single non-pointwise step per block. This convention, and the results in this subsection, are empirical and are not covered by Theorems 2 and 3.

Table 27: KIN depth scan (d_v=64, M=64 RFF features). FNO comparison shown for D3. KIN is 10–60× worse in absolute error.

Dataset	Architecture	L=1	L=2	L=3	L=4	L=6
D3 Burgers	KIN	.094444	.019639	.014743	.011131	.009715
	FNO	.001585	.000671	.000660	.000567	.000623
	KIN/FNO ratio	59.6×	29.3×	22.3×	19.6×	15.6×
D6 Linear	KIN	.024504	.022036	.022041	.009141	.009144
D7 Order 2	KIN	.019633	.014537	.014264	.011109	.010824
D8 Order 3	KIN	.023821	.011999	.009947	.009587	.007769

Table 28: Complete dataset specification. All datasets use periodic boundary conditions except D5 (Dirichlet).

ID	PDE	Governing equation	Domain	N_{train}	Resolution	Source
D1	Heat 1D	$\partial_t u = \nu \nabla^2 u$ (ν=0.01)	$[0, 2\pi]$	1000	256	generated (spectral)
D2	Allen–Cahn	$\partial_t u = \varepsilon \nabla^2 u + u - u^3$ (ε=0.01)	$[0, 2\pi]$	1000	256	generated (spectral)
D3	Burgers	$\partial_t u + u \partial_x u = \nu \nabla^2 u$ (ν=0.1)	$[0, 1]$	1000	256	[22]
D4	NS 2D	$\partial_t w + u \cdot \nabla w = \nu \nabla^2 w + f$ (ν=10^{-5})	$(0, 1)^2$	1000	64^2	[22]
D5	Darcy	$-\nabla \cdot (a \nabla u) = f$	$(0, 1)^2$	900	64^2	[22]
D6	Synth. linear	$\partial_t u = Au - \gamma u$ (γ=0.5)	$[0, 2\pi]$	1000	256	generated (exact)
D7	Synth. order 2	$\partial_t u = u(Au) - \gamma u$ (γ=0.5)	$[0, 2\pi]$	1000	256	generated (RK4)
D8	Synth. order 3	$\partial_t u = u \cdot A(u \cdot Au) - \gamma u$ (γ=0.5)	$[0, 2\pi]$	1000	256	generated (RK4)
D9	KdV	$\partial_t u + u \partial_x u + \partial_x^3 u = 0$	$[0, 2\pi]$	1000	256	generated (IMEX)
D10	Sine-Gordon	$\partial_{tt} u = \partial_{xx} u - \sin(u)$	$[0, 2\pi]$	1000	256	generated (Verlet)

Table 29: Training hyperparameters (defaults). Any deviations are noted in experiment-specific descriptions.

Hyperparameter	Default (1D)	Default (2D)
Width d_v	64	64
Fourier modes m	32	12×12
Optimizer	Adam (β_1=0.9, β_2=0.999)	Same
Learning rate	10^{-3}	10^{-3}
Weight decay	10^{-5}	10^{-5}
LR scheduler	Cosine annealing ($T_{\max}$=500)	Same
Minimum LR	10^{-6}	10^{-6}
Epochs	500	500
Batch size	32	10
Loss function	Relative L^2	Same
Activation	GELU [12]	Same
Gradient clipping	None (vanilla) / 1.0 (all others)	Same
Weight initialization	Kaiming uniform [11]	Same

Table 30: **The depth phase transition is robust across tested block forms** (d_v=64, 32 modes, 3 seeds). The L=1→L=2 ratio is consistent across all forms on D3 Burgers (α_F=1). D7's weak transition reflects damping-dominated dynamics, not a theory failure.

Dataset	Block form	L=1	L=2	L=3	L=1/L=2	Collapse
D3 Burgers (α_F=1)	$\sigma(\mathcal{K}v + Wv)$	.00181	.00084	.00070	**2.16**×	$L \geq 12$
	$\sigma(W \cdot \mathcal{K}v)$	.00222	.00114	.00111	1.96×	$L \geq 4$
	$\sigma(\mathcal{K}v)$	.00206	.00107	.00139	1.93×	$L \geq 4$
D7 Order 2 (α_F=1)	$\sigma(\mathcal{K}v + Wv)$	.00170	.00167	.00165	1.02×	$L \geq 6$
	$\sigma(W \cdot \mathcal{K}v)$	.00294	.00274	.00276	1.07×	$L \geq 4$
	$\sigma(\mathcal{K}v)$	.00295	.00278	.00288	1.06×	$L \geq 4$
D8 Order 3 (α_F=2)	$\sigma(\mathcal{K}v + Wv)$	.00194	.00177	**.00133**	1.10×	$L \geq 6$
	$\sigma(W \cdot \mathcal{K}v)$	.00313	.00274	.00275	1.14×	$L \geq 4$
	$\sigma(\mathcal{K}v)$	.00304	.00277	.00286	1.10×	$L \geq 4$

Table 31: Resolution sweep ($N_x \in \{64, 128, 256, 512\}$, 3 seeds per cell). L^* and the predicted $L_{\min}$=2 are unchanged across all resolutions. D7 has only $\min(N_x/4, 32)$ Fourier modes available, which explains the larger L=1 error at N_x=64; D7 at N_x=512 was not run.

Dataset	N_x	L=1	L=2	L=3	L=4	Ratio
D3 Burgers (α_F=1)	64	1.93×10^{-3}	8.97×10^{-4}	7.25×10^{-4}	**6.74** $\times10^{-4}$	2.86×
	128	1.88×10^{-3}	8.60×10^{-4}	6.84×10^{-4}	**6.61** $\times10^{-4}$	2.85×
	256	1.88×10^{-3}	8.61×10^{-4}	6.86×10^{-4}	**6.57** $\times10^{-4}$	2.86×
	512	1.90×10^{-3}	8.64×10^{-4}	6.82×10^{-4}	**6.63** $\times10^{-4}$	2.86×
D7 Synth. ord 2 (α_F=1)	64	2.20×10^{-3}	1.47×10^{-3}	1.41×10^{-3}	**1.40** $\times10^{-3}$	1.57×
	128	1.74×10^{-3}	1.64×10^{-3}	**1.46** $\times10^{-3}$	1.57×10^{-3}	1.19×
	256	1.76×10^{-3}	1.65×10^{-3}	**1.45** $\times10^{-3}$	1.58×10^{-3}	1.21×

Table 32: Width grid ($d_v \in \{32, 64, 128, 256\}$, 2 seeds per cell). L^* stays at or above $L_{\min}$ for every width; the L=1→ min ratio sharpens at small width and fades at large width.

Dataset	d_v	L=1	L=2	L=3	L=4	Ratio
D3 Burgers (α_F=1, $L_{\min}$=2)	32	3.52×10^{-3}	1.04×10^{-3}	9.38×10^{-4}	**8.06** $\times10^{-4}$	4.36×
	64	1.90×10^{-3}	8.48×10^{-4}	6.80×10^{-4}	**6.38** $\times10^{-4}$	2.98×
	128	1.29×10^{-3}	7.54×10^{-4}	**6.99** $\times10^{-4}$	7.36×10^{-4}	1.84×
	256	1.27×10^{-3}	9.17×10^{-4}	**7.53** $\times10^{-4}$	8.09×10^{-4}	1.68×
D2 Allen–Cahn (α_F=0, $L_{\min}$=1)	32	1.32×10^{-3}	8.44×10^{-4}	5.79×10^{-4}	**5.51** $\times10^{-4}$	2.40×
	64	9.95×10^{-4}	6.11×10^{-4}	5.11×10^{-4}	**5.08** $\times10^{-4}$	1.96×
	128	9.16×10^{-4}	5.38×10^{-4}	**3.91** $\times10^{-4}$	4.46×10^{-4}	2.35×
	256	8.92×10^{-4}	5.28×10^{-4}	**3.39** $\times10^{-4}$	4.22×10^{-4}	2.63×

Table 33: Sample-scaling on D8 (α_F=2, 2 seeds per cell). L^*=3=$L_{\min}$ at every N_{train}.

N_{train}	L=1	L=2	L=3	L=4	Ratio
200	3.74×10^{-3}	3.17×10^{-3}	**2.91** $\times10^{-3}$	2.95×10^{-3}	1.29×
500	2.35×10^{-3}	2.22×10^{-3}	**2.22** $\times10^{-3}$	2.34×10^{-3}	1.06×
1000	1.87×10^{-3}	1.73×10^{-3}	**1.18** $\times10^{-3}$	1.39×10^{-3}	1.60×

Table 34: Activation ablation (2 seeds per cell). On D3 (α_F=1), all four activations select $L^* \geq L_{\min}$=2; the polynomial activation z^2 produces the largest transition ratio (10.26×), consistent with Theorem 3 beyond the GELU evidence in Section 4.3. On D6 (α_F=0), all activations correctly pick L^*=1.

Dataset	σ	L=1	L=2	L=3	L=4	Ratio
D3 Burgers (α_F=1, $L_{\min}$=2)	GELU	1.90×10^{-3}	8.48×10^{-4}	6.80×10^{-4}	**6.38** $\times10^{-4}$	2.98×
	ReLU	6.17×10^{-3}	2.58×10^{-3}	2.13×10^{-3}	**2.09** $\times10^{-3}$	2.95×
	tanh	3.21×10^{-3}	1.20×10^{-3}	1.09×10^{-3}	**9.03** $\times10^{-4}$	3.56×
	z^2	9.08×10^{-2}	3.89×10^{-2}	**8.85** $\times10^{-3}$	2.00×10^{-2}	10.26×
D6 Linear (α_F=0, $L_{\min}$=1)	GELU	**3.91** $\times10^{-4}$	8.06×10^{-4}	8.71×10^{-4}	9.38×10^{-4}	—
	ReLU	**6.61** $\times10^{-4}$	8.77×10^{-4}	1.06×10^{-3}	1.18×10^{-3}	—
	tanh	**5.26** $\times10^{-4}$	8.96×10^{-4}	8.32×10^{-4}	7.71×10^{-4}	—
	z^2	**3.23** $\times10^{-4}$	6.05×10^{-4}	1.01×10^{-3}	6.92×10^{-3}	—

Table 35: 3D PDEBench compressible Navier–Stokes (M=1, 64^3, five state channels, 50 train / 10 test trajectories, FNO3d, d_v=16, modes 8, 100 epochs, single seed). Each Euler-form flux divergence is $\nabla \cdot$ (pointwise-NL(state)), giving α_F=1 and predicted $L_{\min}$=2. Compute: <1 minute, <0.7 GB peak per run on one RTX 4090.

L	Params	Wall (s)	Train MSE	Best test rel. L^2
1	525,061	25.8	0.0936	0.2675
2	1,049,621	36.0	0.0835	**0.2663**
3	1,574,181	46.1	0.0768	0.2668
4	2,098,741	57.9	0.0762	0.2675

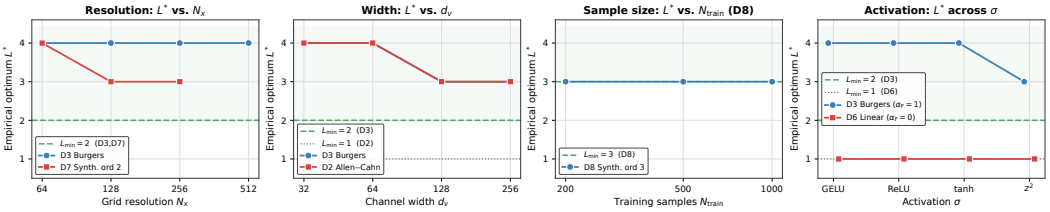


Figure 3: $L_{\min}$ **is invariant under resolution, width, sample size, and activation.** Each panel plots empirical L^* against one factor; green dashed lines mark the predicted $L_{\min}=\alpha_F+1$. L^* lies at or above $L_{\min}$ in every cell.

Table 36: Controlled 2×2 ablation on D3 Burgers (α_F=1, $L_{\min}$=2): {vanilla, residual} FNO (d_v=64, 32 modes) × {no clipping, gradient-norm clipping at 1.0}, identical protocol in all four columns. Test relative L^2, mean ± std over 3 seeds.

	Vanilla		Residual	
L	no clip	clip 1.0	no clip	clip 1.0
1	3.08e-03±2.1e-04	3.31e-03±6.3e-05	3.52e-03±3.3e-04	3.65e-03±3.9e-04
2	1.38e-03±5.8e-05	1.65e-03±8.3e-05	1.53e-03±4.9e-05	1.73e-03±8.6e-05
4	1.09e-03±5.4e-05	1.27e-03±7.1e-05	8.68e-04±8.9e-06	1.03e-03±3.6e-05
8	1.17e-03±1.9e-05	1.40e-03±6.2e-05	6.27e-04±2.6e-05	7.84e-04±1.5e-05
12	2.12e-03±6.1e-04$^{\dagger 1}$	1.77e-03±2.7e-05$^{\dagger 1}$	5.68e-04±2.9e-05	6.62e-04±1.0e-05
16	COLL (3/3)	COLL (3/3)	5.57e-04±4.9e-05	6.40e-04±1.8e-05
20	COLL (3/3)	COLL (3/3)	5.36e-04±2.1e-05	5.86e-04±3.3e-05

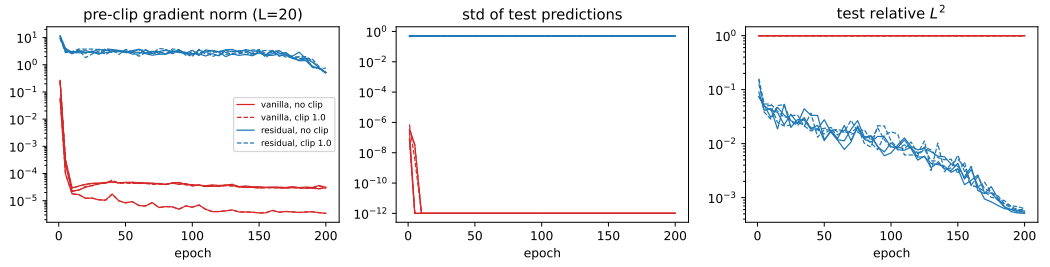


Figure 4: Training dynamics at the deepest ablation depth on D3 Burgers, all seeds (red: vanilla, blue: residual; solid: no clipping, dashed: clipping at 1.0). Left: mean pre-clipping global gradient norm per epoch. Middle: standard deviation of the test predictions. Right: test relative L^2.

Table 37: GNO depth scan. Test relative L^2, mean $\pm$ std over 3 seeds; bold: best depth per dataset. burgers: L^*=6 ($L_{\min}$=2), D8: L^*=6 ($L_{\min}$=3).

L	Params	D3 Burgers ($L_{\min}$=2)	D8 order 3 ($L_{\min}$=3)
1	0.55M	3.23e-01±4.5e-05	3.07e-02±1.3e-04
2	0.84M	4.65e-02±6.6e-04	2.70e-02±3.7e-04
3	1.13M	2.11e-02±8.3e-04	2.07e-02±2.7e-03
4	1.42M	1.47e-02±6.0e-04	1.60e-02±1.7e-04
6	2.01M	**1.10e-02±1.9e-04**	**1.32e-02±1.3e-03**

Table 38: Transolver-style physics-attention depth scan. Test relative L^2, mean $\pm$ std over 3 seeds; bold: best depth per dataset. burgers: L^*=4 ($L_{\min}$=2), D8: L^*=1 ($L_{\min}$=3).

L	Params	D3 Burgers ($L_{\min}$=2)	D8 order 3 ($L_{\min}$=3)
1	0.43M	4.18e-02±4.0e-03	**3.09e-02±8.1e-05**
2	0.76M	3.05e-02±2.0e-03	3.10e-02±4.9e-05
3	1.09M	2.86e-02±1.4e-03	3.11e-02±7.3e-05
4	1.42M	**2.55e-02±1.1e-03**	3.12e-02±2.0e-05
6	2.09M	3.05e-02±1.7e-03	3.12e-02±1.5e-05

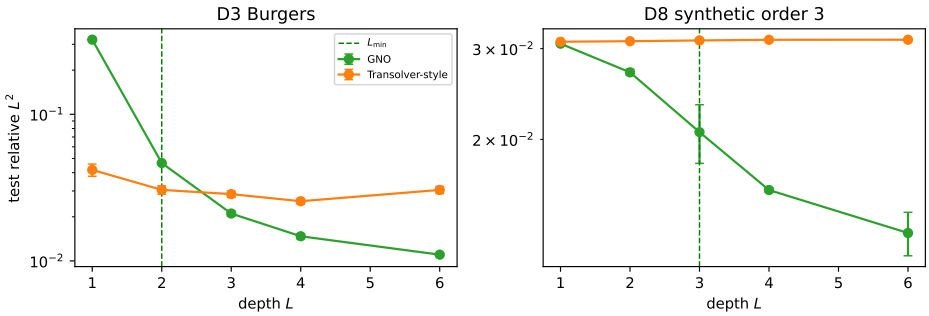


Figure 5: Depth scans of the GNO and the Transolver-style operator (mean $\pm$ std over 3 seeds); green dashed lines mark $L_{\min}=\alpha_F+1$.