CERTIFYING RESIDUAL ARCHITECTURES FROM THEIR PRIMITIVES: A SHARP STABILITY THRESHOLD

Anonymous authors
Paper under double-blind review

ABSTRACT

Whether a deep residual architecture trains stably is usually determined by training it, which is expensive and answers the question only for the architecture, depth, and floating-point format that were tried. In practice stability is secured by heuristics on where in the residual block to place a normalization and which one to use; each is supported by experiments, and no common principle explains why and when they work. We show that stability can instead be certified before training, from the architectural primitives of the residual block, as an explicit function of depth and floating-point format. The certificate rests on two elements: (a) on a power-law growth bound for the residual block, $\|v(x)\| \le c\|x\|^q + b$, whose exponent q is computed from the block's primitives by an arithmetic of exponents (forward); (b) on bounds on the gradients through a Lipschitz condition on the block over the reachable states (backward). The certificate determines the depth at which the state can reach the largest finite floating-point value M: for $q \le 1$, overflow requires at least order $\log M$ layers, whereas for $q > 1$ it can occur within order $\log_q \log M$ layers; the threshold $q = 1$ is sharp. Every normalization and every bounded activation sets $q = 0$, and the arithmetic identifies minimal modifications that bring a block from $q > 1$ to $q = 1$ without normalization. Empirically, in forecasting only $q > 1$ blocks blow up, as the forward certificate predicts; in GPT-2 on OpenWebText, $q = 1$ blocks without normalization also diverge, and there the forward certificate holds while the backward one fails, in the attention block with the largest query-key coefficient. Relaxing normalization from $q = 0$ to $q = 1$ improves out-of-distribution generalization in operator learning, and gives accuracy gains in time-series forecasting. As recent deep models, including foundation models, grow more complex and more costly to train, our results give a way to settle the trade-off between stability and representational flexibility of a residual architecture directly from the primitives of its blocks, before any training.

1 INTRODUCTION

Residual architectures, from Transformers to state-space models and neural operators, update a hidden state by adding the output of a block to it, one layer at a time. Whether such a stack trains stably at a given depth and in a given floating-point format is in practice decided by training it. Normalization is the standard safeguard, and where to place it and which one to use (Pre-LN, Peri-LN, RMSNorm, Dynamic Tanh; Xiong et al., 2020; Kim et al., 2025; Zhang & Sennrich, 2019; Zhu et al., 2025) are chosen by experiment. At the scale of current models this is expensive, and a negative result says little about why a configuration failed or which change would fix it.

A residual architecture of depth D updates its hidden state $X_k \in \mathbb{R}^N$ by

$$X_{k+1} = X_k + \Delta t\, v(X_k; \theta_k), \qquad k = 0, \dots, D-1, \tag{1}$$

where the *velocity* $v(\cdot; \theta_k)$ is the k-th residual block with parameters $\theta_k \in \Theta$ and $\Delta t > 0$ is the residual scaling ($\Delta t = 1$ by default). Training minimizes a terminal loss $\mathbb{E}\, G(X_D, y)$, and backpropagation computes the costate $p_k = \nabla_{X_k} G$ backward through the block Jacobians,

$$p_k = \big(I + \Delta t\, \nabla_X v(X_k; \theta_k)\big)^\top p_{k+1}, \qquad p_D = \nabla_X G(X_D, y). \tag{2}$$

We fix $\Delta t = 1$ and D and take $G(\cdot, y)$ to be C^1 with Lipschitz constant $\mathrm{Lip}(G)$; training stays finite only if X_k and p_k stay below the largest finite floating-point value M.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Certified depths as a function of the forward growth exponent q and Jacobian growth exponent σ. M: largest finite floating-point value; D_F: forward depth; D_∇: backward depth; $D_{\text{cert}} = \min\{D_F, D_\nabla\}$. The $\sigma = 0$ row is the global-Lipschitz baseline. See Section 2.

q	σ	D_F	$D_\nabla \geq$	D_{cert}
< 1	> 0	(i) $\frac{M^{1-q}}{(1-q)(c+b)\Delta t}$	$\frac{1-q}{\sigma}\,\frac{\log M}{\log\log M}$	$\geq \frac{1-q}{\sigma}\,\frac{\log M}{\log\log M}$
$= 1$	> 0	(ii) $\frac{\log M}{\log(1+c\Delta t)}$	$\sqrt{\frac{\log M}{\sigma \log(1+c\Delta t)}}$	$\geq \sqrt{\frac{\log M}{\sigma \log(1+c\Delta t)}}$
> 1	> 0	$\log_q \log M + O(1)$	$\log_q \log M\,(1 - o(1))$	$\sim \log_q \log M$
≤ 1	0	(i) if $q < 1$, (ii) if $q = 1$	$\frac{\log M}{\log(1+\Delta t(\ell_0+\ell_1))}$	$\geq \frac{\log M}{\log(1+\Delta t(\ell_0+\ell_1))}$

A global Lipschitz bound $\sup_{x,\theta} \|\nabla_X v(x;\theta)\|_2 \leq L_\infty$ on the block would suffice: it bounds every Jacobian in (2) and, by the mean value inequality, forces linear growth, so $\|X_D\|$ and $\|p_0\|$ grow at most like $(1 + \Delta t\, L_\infty)^D$. Enforcing it primitive by primitive is nontrivial, however. For linear layers and MLPs, spectral normalization bounds the constant (Miyato et al., 2018), but dot-product self-attention is not globally Lipschitz (Kim et al., 2021), the selective state-space block of Mamba multiplies input-dependent quantities and grows superlinearly, and for Pre-LN blocks the constant scales like $1/\sqrt{\epsilon}$. A global Lipschitz condition on v is therefore generally too restrictive: the recursion only visits the states reachable within D layers, so a global bound controls a much larger region than necessary. To address the limitation, this paper develops three components:

A sharp stability threshold at $q = 1$ beyond global Lipschitzness. The required bounds are therefore a bound on how far the state can travel within D layers and a bound on the Jacobian on that region alone. Under the growth bound $\|v(x)\| \leq c\|x\|^q + b$, every state reachable in D layers lies in a ball of radius $\mathcal{R}_D(q)$. The Jacobians in (2) are then bounded by the Lipschitz constant $L(\mathcal{R}_D(q))$ of the block on that ball, and how this constant grows with the radius is the second exponent, $L(R) = O(R^\sigma)$: $\sigma = 0$ is the global-Lipschitz case, $\sigma > 0$ lets the Jacobian grow with the reachable radius. The certified depth is therefore the minimum of the forward depth allowed by the reachable-state bound and the backward depth allowed by the worst-case sensitivity over that region:

$$\text{growth exponent } q \Longrightarrow \text{reachable radius } \mathcal{R}_D(q) \Longrightarrow \text{Lipschitz constant } L(\mathcal{R}_D(q)) = O(\mathcal{R}_D(q)^\sigma)$$
$$D_{\text{cert}} = \min\{D_F, D_\nabla\}$$

Here D_F and D_∇ are the largest depths at which the forward and backward bounds stay below M (Section 2). For $\sigma > 0$ the exponent $q = 1$ is a sharp threshold: $D_{\text{cert}} \gtrsim \log M / \log\log M$ for $q < 1$ and $D_{\text{cert}} \gtrsim \sqrt{\log M}$ for $q = 1$, whereas for $q > 1$ the reachable radius grows doubly exponentially and $D_{\text{cert}} = O(\log_q \log M)$, however small $\sigma > 0$ (Table 1). Moreover, the optimal-control characterization of training shows that the loss-minimizing block attains the worst case of the forward growth (Theorem A.1).

A primitive-level arithmetic of exponents for a priori architecture certification. In Section 3, we develop a symbolic arithmetic that propagates the state-growth exponent q through the primitives of a residual velocity block (Table 2), together with analogous rules for the sensitivity-growth exponent σ (App. B.2). This yields a priori, architecture-level stability estimates directly from a model's computation graph, allowing potentially unstable primitives and targeted modifications to be identified before training.

Empirical validation of certificate: stability and generalization across tasks. In Section 4, we test how the architectural exponents q and σ govern stability and whether the critical regime $q = 1$ can improve generalization over the normalized baseline $q = 0$. At small scale, only $q > 1$ blocks exhibit blow-up, while reducing them to $q = 1$ without normalization removes the instability; among stable architectures, $q = 1$ improves out-of-distribution generalization in operator learning and accuracy in forecasting. At GPT-2 scale on OpenWebText, however, $q = 1$ alone is not sufficient: the no-LN model diverges despite satisfying the forward certificate, with the backward certificate localizing the failure to the attention block with the largest query-key sensitivity. Finally, in the

Toto 2.0 foundation model, the certification first calls for eliminating the $q = 2$ gated-MLP growth by reducing $q > 1$ to $q \leq 1$, before addressing state-dependent sensitivity.

2 FORWARD AND BACKWARD CERTIFICATION

We study the residual recursion (1) and its adjoint (2) to certify whether the states and costates remain finite to a given depth, without requiring global Lipschitz continuity. The forward analysis bounds the reachable state radius, while the backward analysis bounds the costate through the local Jacobian sensitivity over that region. Together, these bounds give the certified depths D_F, D_∇, and $D_{\text{cert}} = \min\{D_F, D_\nabla\}$.

2.1 FORWARD PASS: REACHABLE-STATE GROWTH

The first question is how far the hidden state can grow under repeated residual updates. To isolate this mechanism, we consider residual blocks with a uniform polynomial growth envelope. This class is broad enough to include linear-growth blocks as well as blocks whose state-dependent operations produce superlinear growth, while reducing their large-state behavior to a single exponent that can be tracked across architectural primitives.

Assumption 2.1 (Uniform polynomial growth)**.** *There are $q \geq 0$, $c > 0$, and $b \geq 0$ such that*

$$\|v(x;\theta)\| \leq c\|x\|^q + b \qquad \text{for all } x \in \mathbb{R}^N,\ \theta \in \Theta. \tag{3}$$

We call q the *input-magnitude exponent*. The cases $q < 1$, $q = 1$, and $q > 1$ correspond to sublinear, linear, and superlinear growth, respectively. For a depth-D network, let $\mathcal{R}_D(q)$ denote a deterministic upper envelope on the reachable radius,

$$\|X_k\| \leq \mathcal{R}_k(q), \qquad k \leq D,$$

obtained from Assumption 2.1 and the residual recursion. The following comparison bound gives an explicit envelope in each growth regime. The proof is in App. A.1.

Theorem 2.2 (Reachable-radius bound)**.** *Under Assumption 2.1, let $\|X_0\| \leq r_0$ with $r_0 > 0$, and $\bar{r}_0 = \max(r_0, 1)$. Every trajectory satisfies $\|X_k\| \leq \mathcal{R}_k(q)$ for all $k \geq 0$, where*

(i) *$q < 1$:* $\mathcal{R}_D = \big(\bar{r}_0^{\,1-q} + (1-q)(c+b)D\Delta t\big)^{1/(1-q)}$;

(ii) *$q = 1$:* $\mathcal{R}_D = (1 + c\Delta t)^D (r_0 + b/c) - b/c$;

(iii) *$q > 1$, $b = 0$:* $\mathcal{R}_D = \rho \exp\big(C_q\, q^D\big)$ *with* $\rho = (c\Delta t)^{-1/(q-1)}$ *and* $C_q = \log\big(2^{1/(q-1)}(1 + r_0/\rho)\big)$; *for $b > 0$, the same form holds with $c + b$ and $\bar{r}_0$ in place of c and r_0.*

The envelope is finite and nondecreasing in D.

The forward certified depth is the largest depth for which the reachable radius remains representable, $D_F := \max\{D \in \mathbb{N} : \mathcal{R}_D(q) < M\}$. Since $\mathcal{R}_D$ is nondecreasing, Theorem 2.2 can be inverted to give

$$D_F = \left\lfloor \begin{cases} \dfrac{M^{1-q} - \bar{r}_0^{\,1-q}}{(1-q)(c+b)\Delta t}, & q < 1, \\[2ex] \dfrac{\log\big((M + b/c)/(r_0 + b/c)\big)}{\log(1 + c\Delta t)}, & q = 1, \\[2ex] \log_q \dfrac{\log(M/\rho)}{C_q}, & q > 1, \end{cases} \right\rfloor - 1. \tag{4}$$

Thus the forward depth scales polynomially in M for $q < 1$, grows logarithmically as $\Theta(\log M)$ at $q = 1$, and collapses to the double-logarithmic scale $\Theta(\log_q \log M)$ for $q > 1$.

2.2 BACKWARD PASS: LOCAL SENSITIVITY AND CERTIFICATION

The forward exponent q controls how far the state can reach, but it does not by itself determine how large the block Jacobian can become on that region. We therefore measure the worst-case local

sensitivity over the reachable states. For $R > 0$, define

$$L(R) := \sup_{\theta \in \Theta,\ \|x\| \le R} \|\nabla_X v(x;\theta)\|_2. \tag{5}$$

Assuming $\nabla_X v$ is continuous in (x, θ) and Θ is compact, $L(R)$ is finite for every finite R. Since every state satisfies $\|X_k\| \le \mathcal{R}_D(q)$ for $k \le D$ by Theorem 2.2, every Jacobian along a depth-D trajectory is bounded by $L(\mathcal{R}_D(q))$. The adjoint recursion therefore gives

$$\|p_i\| \le \big(1 + \Delta t\, L(\mathcal{R}_D(q))\big)^{D-i} \mathrm{Lip}(G), \qquad 0 \le i \le D. \tag{6}$$

Theorem 2.3 (Backward depth certificate)**.** *Under Assumption 2.1, with each $v(\cdot;\theta)$ continuously differentiable, if*

$$L(\mathcal{R}_D(q)) < \frac{(M/\mathrm{Lip}(G))^{1/D} - 1}{\Delta t}, \tag{7}$$

then $\|p_i\| < M$ for every $0 \le i \le D$.

The proof is in App. A.3. The right-hand side of (7) is the allowable Jacobian budget at depth D; it decreases as D increases. We define D_∇ as the largest depth satisfying (7). Thus the backward certificate reduces to a single question: how does the local sensitivity $L(R)$ grow with the reachable radius?

Assumption 2.1 controls the growth of the state, but it does not specify how that state growth is amplified by the Jacobian. A global Lipschitz bound would set $L(R) = O(1)$, but, as discussed in Section 1, that restriction excludes important state-dependent blocks. We therefore use the weakest parallel power-law description needed to track this amplification:

Assumption 2.4 (Uniform Jacobian growth)**.** *Each $v(\cdot;\theta)$ is C^1, and there are $\sigma \ge 0$ and $\ell_1, \ell_0 \ge 0$ such that*

$$L(R) \le \ell_1 R^\sigma + \ell_0 \qquad \textit{for all } R \ge 1. \tag{8}$$

We call σ the *Jacobian growth exponent*. Thus $\sigma = 0$ recovers the global-Lipschitz case, whereas $\sigma > 0$ allows the local sensitivity to increase with the state radius. Affine maps, Lipschitz activations, and bounded-weight MLPs have $\sigma = 0$; pre-normalization also gives $\sigma = 0$, while gated MLPs and unnormalized self-attention have $\sigma = 1$ and $\sigma = 2$, respectively (App. B.2, Lemma A.7). For self-attention, the $\sigma = 2$ growth arises from the bilinear query-key logits $S = X^\top W_Q^\top W_K X/\sqrt{d_k}$ and the value-dependent Jacobian $W_V X\, \partial_X \,\mathrm{softmax}(S)$ (Lemma A.7).

Under Assumption 2.4, combining (6) with Theorem 2.2 gives, with $\bar{\mathcal{R}}_D = \max\{\mathcal{R}_D, 1\}$,

$$\log \frac{\|p_0\|}{\mathrm{Lip}(G)} \le D \log\big(1 + \Delta t(\ell_0 + \ell_1)\big) + \sigma D \log \bar{\mathcal{R}}_D, \qquad \sigma D \log \bar{\mathcal{R}}_D = \begin{cases} O(\sigma D \log D), & q < 1, \\ O(\sigma D^2), & q = 1, \\ O(\sigma D\, q^D), & q > 1, \end{cases} \tag{9}$$

Requiring the right-hand side to remain below $\log(M/\mathrm{Lip}(G))$ gives

$$D_\nabla \gtrsim \begin{cases} \dfrac{\log M}{\log\big(1 + \Delta t(\ell_0 + \ell_1)\big)}, & \sigma = 0 \text{ (then } q \le 1), \\ \dfrac{1-q}{\sigma} \dfrac{\log M}{\log\log M}, & \sigma > 0,\ q < 1, \\ \sqrt{\dfrac{2 \log M}{\sigma \log(1 + c\Delta t)}}, & \sigma > 0,\ q = 1, \\ \log_q \log M - O(1), & \sigma > 0,\ q > 1. \end{cases} \tag{10}$$

Thus, $D_{\mathrm{cert}} := \min\{D_F, D_\nabla\}$, as summarized in Table 1. The exponents q and σ are distinct and may each be small or large. Yet $q = 1$ is a sharp threshold: for every fixed $\sigma > 0$, any $q > 1$ forces $D_{\mathrm{cert}} = \Theta(\log_q \log M)$, because the doubly exponential growth of $\mathcal{R}_D$ is inherited by the backward bound, however small σ is.

3 FROM STABILITY EXPONENTS TO ARCHITECTURE DESIGN

Section 2 characterizes the growth classes of the forward and backward states in terms of q and σ. For architecture design, however, it is important to determine a priori how q grows across the primitives of a residual block and which primitives increase its state-dependent sensitivity, rather than estimating these exponents empirically during or after training (App. C). We therefore develop arithmetic rules for the input-magnitude exponents of individual operations and use the exponents of standard primitives to obtain an upper bound on the exponents of the entire residual block.

Input-magnitude exponent arithmetic. To obtain an a priori bound for a composite residual block, we propagate the input-magnitude exponent through its constituent operations. Let $q(B)$ denote the infimum of the exponents in Assumption 2.1 for a primitive B (App. B).

Theorem 3.1 (Arithmetic of input-magnitude exponents)**.** *Let B_1, B_2 be primitives satisfying Assumption 2.1, with $q_i = q(B_i)$ and compatible dimensions. Then*

(i) composition: $q(B_1 \circ B_2) \le q_1 q_2$*;*

(ii) sum: $q(B_1 + B_2) \le \max(q_1, q_2)$*;*

(iii) product: $q(B_1 \odot B_2) \le q_1 + q_2$*;*

(iv) residual wrapping: $q(I + \Delta t B_1) \le \max(1, q_1)$*;*

(v) normalization: $q(N \circ B) = 0$ *for* $N \in \{\mathrm{LN}, \mathrm{RMSNorm}, \mathrm{ScaleNorm}\}$.

The proof is in App. B.1. The Jacobian growth exponent σ obeys analogous arithmetic (Thm. B.2, App. B.2). The rules of both arithmetics induce *associative* operations on the exponents: a residual velocity can therefore be decomposed into primitives, each carrying the exponents in Table 2, and the block exponents obtained by applying Theorems 3.1 and B.2 are independent of the parenthesization of its computation graph (App. B.3).

Exponents of standard primitives. The arithmetic reduces the exponent bounds of a residual block to those of its constituent primitives. Table 2 lists the (q, σ) exponents of common building blocks, from which block-level bounds follow by the two arithmetics. The corresponding coefficients are given in App. B.4, with architecture-specific derivations in App. D.2.

Table 2: Input-magnitude exponents q and Jacobian growth exponents σ of common architectural primitives. Normalizations use $\epsilon > 0$; ReLU is interpreted in the Clarke sense; – indicates that the exponent is not derived. Attention is multi-token, with $q = 1$ understood in the worst-token sense (App. D.2).

Primitive	q	σ
Saturating activation (tanh, sigmoid, ReLU6 (Howard et al., 2017))	0	0
DyT $\gamma \tanh(\alpha x) + \beta$ (Zhu et al., 2025)	0	0
BHyT (Byun et al., 2026)	0	–
Softmax, row-stochastic (Bridle, 1990)	0	0
LayerNorm (Ba et al., 2016) ($\epsilon > 0$), RMSNorm (Zhang & Sennrich, 2019), ScaleNorm (Nguyen & Salazar, 2019)	0	0
Affine map $x \mapsto Wx + b$	1	0
Lipschitz activation (ReLU, GELU (Hendrycks & Gimpel, 2016), SiLU (Elfwing et al., 2018))	1	0
MLP $W_L \phi_L(\cdots W_1 x)$ with bounded weights and Lipschitz ϕ_ℓ	1	0
Query-key logits $X^\top W_Q^\top W_K X / \sqrt{d_k}$	2	1
Self-attention $\mathrm{softmax}(QK^\top/\sqrt{d_k})V$ (Vaswani et al., 2017)	1	2
Multi-head attention $\sum_h W_O^{(h)} \mathrm{Attn}^{(h)}(X)$	1	2
Entrywise power $x \mapsto x^p$, $p \ge 2$	p	$p-1$
Selective SSM, Mamba without normalization (Gu & Dao, 2024)	5	≥ 4

From symbolic exponents to architecture-level certification. Propagating (q, σ) through a computation graph yields a per-block stability certificate. The theory requires controlling both exponents: Table 1 shows that $q > 1$ sharply reduces the certified depth, so a first design principle is to

enforce $q \leq 1$ for each residual block, as illustrated in Sections 4.1 and 4.4. Once $q \leq 1$, a second principle is to regularize blocks with large σ, as illustrated in Section 4.3, since state-dependent sensitivity can still limit certified depth. This gives a per-block certificate based on the forward and backward bounds (App. B.5), which extends to heterogeneous residual architectures through the between-block estimate in App. B.6.

4 RESULTS

The experiments examine how the architectural exponents q and σ govern stability across forecasting, operator learning, and foundation models, and whether allowing the critical regime $q = 1$ instead of the normalized baseline $q = 0$ can improve generalization without compromising stability (Table 6, App. D.1). We first test the sharp $q = 1$ threshold by varying the residual growth exponent and depth in time-series forecasting. We then examine whether $q = 1$ can improve extrapolation in operator learning and forecasting without compromising stability. We then use GPT-2 to test whether controlling $q \leq 1$ alone is sufficient or whether state-dependent sensitivity must also be regularized, and finally Toto 2.0 to test the importance of eliminating $q > 1$ growth in a pre-trained forecasting architecture.

4.1 TIME-SERIES FORECASTING

In long-horizon time-series forecasting, a model observes a multivariate series over a fixed input window, here 96 time steps, and predicts its values over the next H steps. We evaluate on two standard benchmarks, ETTm1 (Zhou et al., 2021) (7 channels, 15-minute sampling) and Weather (Wu et al., 2021) (21 channels, 10-minute sampling), with $H \in \{96, 192, 336, 720\}$. To study the effect of the residual exponent, we use two backbones, Mamba (Gu & Dao, 2024) in the CMamba forecaster (Zeng et al., 2024) and Transformer-based PatchTST (Nie et al., 2023), with $D \in \{3, 8\}$ residual updates. Each backbone is evaluated with RMSNorm, Pre-LN, or Peri-LN (Zhang & Sennrich, 2019; Xiong et al., 2020; Kim et al., 2025), or without normalization (no-LN), and Mamba is additionally evaluated with the linear-growth modification described below. We use 6 seeds per cell on ETTm1 and 3 on Weather (App. D.3).

Only the $q > 1$ block blows up, and more often at larger depth. Without normalization, Transformer attention and MLP blocks have $q = 1$, whereas the selective state-space block of Mamba has $q = 5$ because input-dependent quantities are multiplied together. Normalizing only the input to the selective-parameter branch and gating its output with a sigmoid reduces the Mamba block to $q = 1$ without introducing a normalization layer; we call this the linear-growth block. With normalization, all blocks have $q = 0$ (App. D.2). Table 3 reports the fraction of runs that blow up, defined as either a nonfinite loss or a final test MSE above 10 on standardized targets. For $q > 1$, Table 1 predicts a certified depth on the double-logarithmic scale $D_{\mathrm{cert}} = O(\log_q \log M)$, so stability degrades rapidly with depth. Consistent with this threshold, only no-LN Mamba ($q = 5$) blows up, in 67% of runs, with the rate increasing from 39% at $D = 3$ to 94% at $D = 8$.

Table 3: Blow-up rate (%) of forecasting runs per cell, per dataset ("all"), and over both datasets ("total"). Blow-ups occur only for $q > 1$, and their rate increases with depth. Counts are given in App. D.3.

				ETTm1 (6 seeds / cell)					Weather (3 seeds / cell)					
Backbone	Variant	q	D	96	192	336	720	**all**	96	192	336	720	**all**	total
Mamba	no-LN	5	3	33	50	67	50	**50**	0	33	0	33	**17**	39
			8	100	100	100	100	**100**	33	100	100	100	**83**	94
Mamba	linear growth	1	3, 8	0	0	0	0	**0**	0	0	0	0	**0**	0
PatchTST	no-LN	1	3, 8	0	0	0	0	**0**	0	0	0	0	**0**	0
Mamba & PatchTST	RMSNorm, Pre-LN, Peri-LN	0	3, 8	0	0	0	0	**0**	0	0	0	0	**0**	0

Among stable blocks, $q = 1$ tends to be more accurate. Figure 1 compares the test MSE of the stable variants. For each backbone and dataset, we take the median MSE of the three normalized variants as the reference and report relative deviations from it. For PatchTST, the no-LN variant ($q = 1$) is below this reference in all cells, while for Mamba it is consistently lower on Weather and

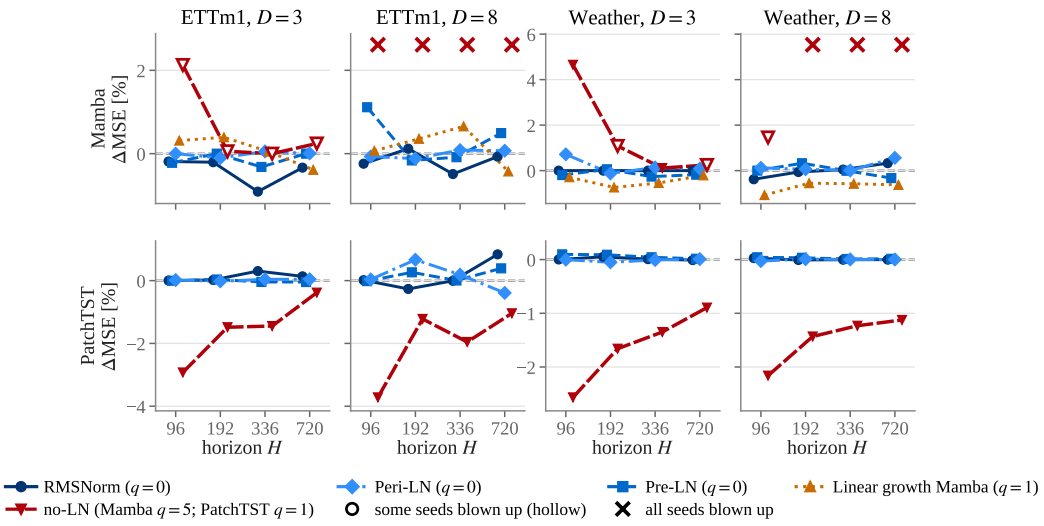


Figure 1: Forecasting accuracy, shown as the signed relative deviation of each variant's mean test MSE from the per-cell median across variants, $\Delta\text{MSE} = 100(\text{MSE} - \text{med}(\text{MSE}))/\,\text{med}(\text{MSE})$. The dashed line indicates the per-cell median; negative values indicate lower MSE. For PatchTST, $q = 1$ is below the median in all cells, while for Mamba it is consistently lower on Weather and comparable to the normalized variants on ETTm1.

comparable to the normalized variants on ETTm1. A possible explanation is that a $q = 1$ update can scale with the state, whereas a $q = 0$ update is uniformly bounded in it. This interpretation is not isolated by the experiment, however, since removing normalization also changes the optimization and removes its affine parameters.

4.2 Operator Learning

Problem. We learn the solution operator for transient heat conduction in a 3-D battery module (The MathWorks, Inc., n.d.). Given the ambient temperature, convection coefficient on the cooled faces, and heat generated by each cell, $\theta = (T_{\text{amb}}, h, Q)$, the model predicts the temperature field $T(\cdot, 600\,\text{s})$. Specifically, the governing equation is

$$\rho c_p\, \partial_t T = \nabla\cdot\,(K\nabla T) + Q\,\mathbf{1}_{\Omega_{\text{cell}}}, \qquad -n\cdot K\nabla T = h\,(T - T_{\text{amb}}) \text{ on } \Gamma_{\text{conv}}.$$

The three inputs affect the target to different degrees: the spatial mean temperature changes by 1.00 K per K of T_{amb}, 4.03 K per W of Q, and -0.16 K per W/(m^{2}K) of h (App. D.4). The FNO (Li et al., 2021) takes these parameters as fields on a 32^3 grid and predicts the final temperature field. We use two parameter configurations for training. The **original box** is $[280, 300]$ K $\times$ $[10, 20]$ W/(m^{2}K) $\times$ $[10, 20]$ W, with 216 training solves. The **larger box** is $[270, 310] \times [10, 40] \times [10, 40]$, sampled at the same grid spacing and giving 2,816 solves. In the four residual Fourier layers, $X_{k+1} = X_k + \text{GELU}(\text{LN}(\mathcal{K}_k X_k + W_k X_k))$, LN gives $q = 0$ and removing it gives $q = 1$; the LNs in the lifting and projection networks are kept in both variants (Figure 6).

The exponent affects extrapolation and out-of-distribution (OOD) performance. Figure 2 shows how test error changes as the evaluation parameters move away from the training domain. The two variants have comparable errors within the training box, with small differences depending on the evaluation set. Outside the training box, however, no-LN consistently achieves lower RMSE, and the gap increases with extrapolation distance. This effect becomes more pronounced when training on the larger parameter box. The single-parameter shifts in Figure 2b,d show the same overall pattern, with the largest difference occurring for Q, which has the strongest effect on the target, while the ranking can reverse for the weaker h perturbation.

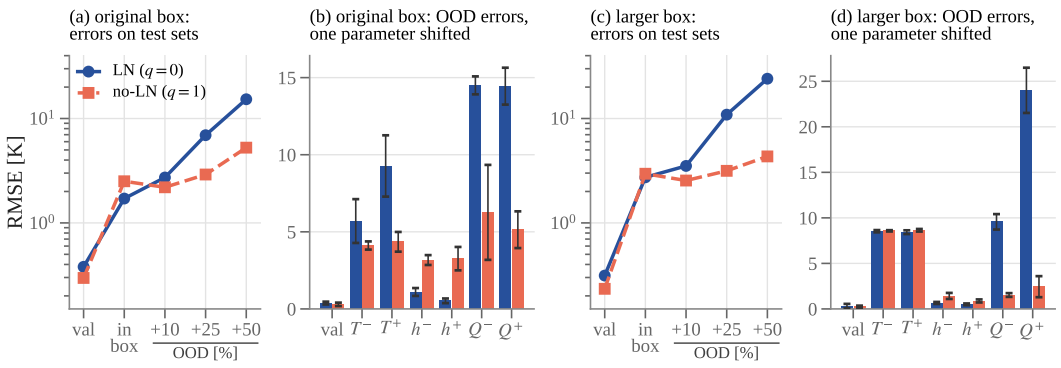


Figure 2: FNO test RMSE for LN ($q = 0$) and no-LN ($q = 1$). The original parameter box is used in (a,b), and the larger box at the original sampling density in (c,d). (a,c) Test sets ordered by distance from the training data: validation, off-grid points within the training box, and OOD points at 10, 25, and 50% beyond the box. (b,d) One parameter shifted 50% beyond its range, with the others fixed at mid-range; whiskers show ± 1 sd. The two variants are comparable within the training box, while no-LN achieves lower RMSE outside the box, with the gap increasing with extrapolation distance and training-box size.

4.3 Language Modeling

To test whether $q \leq 1$ is sufficient at scale, we train nanoGPT GPT-2 124M (Radford et al., 2019; Karpathy, 2022) on OpenWebText (Gokaslan & Cohen, 2019) in two variants. *Pre-LN* normalizes the input of each branch ($q = 0$), while *no-LN* removes the LayerNorm from each residual update ($q = 1$). The model has 12 Transformer layers, each with two heterogeneous residual updates: a self-attention branch $(q, \sigma) = (1, 2)$ followed by an MLP branch $(q, \sigma) = (1, 0)$, with $\sigma = 0$ from bounded weights and Lipschitz activations. Thus, there are $D = 24$ residual updates, and we use 4 seeds per variant (App. D.5). All four Pre-LN runs train to completion at 20,000 iterations, whereas all four no-LN runs diverge at iterations 1,250, 3,000, 4,750, and 7,500. We analyze the Pre-LN and no-LN variants at the last finite checkpoint (iteration 6,750) and track the subsequent blow-up behavior through a training replay from this checkpoint.

$q \leq 1$ **is not enough: sensitivity must be controlled.** The Pre-LN variant remains stable: normalizing the input to each branch fixes $q_k = \sigma_k = 0$, keeping the branch input bounded. In contrast, the no-LN variant has $q = 1$, so the forward certificate still holds at the last finite checkpoint (iteration 6,750), but its attention sensitivity is already large. Figure 3a shows that the query-key coefficient κ_h reaches 175 in block 10 versus 4.4 for Pre-LN. Because the attention logits scale as $\kappa R^2/\sqrt{d_k}$, the large coefficient is further amplified by the token norm: in block 10, $R \approx 50$ gives logits up to 4.0×10^4 (Figure 3b). For $q = 1, \sigma > 0$, Table 1 gives a $\sqrt{\log M/\sigma}$-scale backward-depth budget, so the backward certificate can fail while the forward certificate still holds. Here, the failure stems from the large attention sensitivity at $D = 24$, with the largest factor arising from block 10.

4.4 Foundation Models

Toto 2.0 (Khwaja et al., 2026) is a time-series foundation model with released checkpoints up to 2.5B parameters. We use a 10.6M-parameter instantiation that preserves its architectural components, trained from scratch using the released recipe. It has 12 Transformer layers, each with two residual updates in sequence: a self-attention branch with $(q, \sigma) = (1, 2)$ and a gated-MLP branch $(W_g X) \odot \mathrm{SiLU}(W_u X)$ with $(q, \sigma) = (2, 1)$ without normalization. Thus, counting the two sub-blocks separately gives $D = 24$. As released, parameter-free RMSNorm is applied to the input of each branch, giving $q = 0$, which trains stably. We therefore remove RMSNorm and focus on the trigger of blow-up in the resulting architecture, where the $(q, \sigma) = (1, 2)$ attention and $(q, \sigma) = (2, 1)$ gated-MLP updates coexist. Unlike Section 4.1, we pretrain once and evaluate across tasks: training uses GIFT-Eval's leakage-free pretraining corpus mixed with synthetic series from TempoPFN, and evaluation uses 97 dataset-frequency-horizon configurations spanning 23 datasets, seven domains, ten frequencies, and horizons from 6 to 900 steps, scored by MASE (Aksu et al., 2024) (App. D.6).

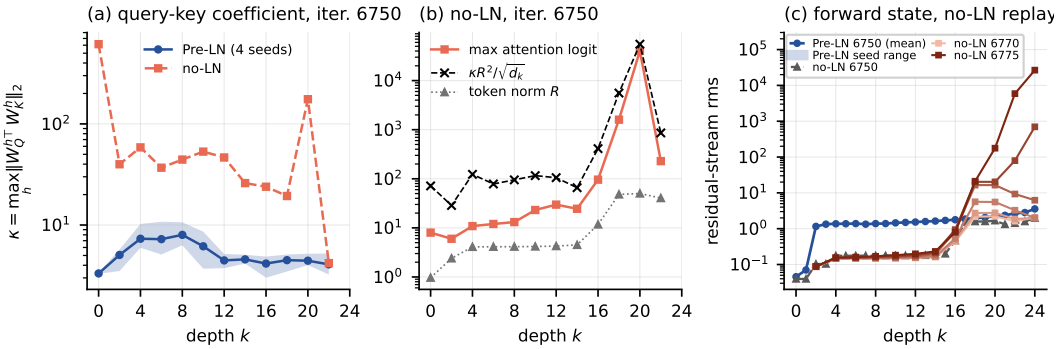


Figure 3: GPT-2 124M on OpenWebText at iteration 6,750. All panels share the depth axis, with $D = 24$ residual updates. (a) The query-key coefficient $\kappa = \max_h \|W_Q^{h\top} W_K^h\|_2$ is substantially larger in no-LN than in Pre-LN. (b) In no-LN, the large coefficient is further amplified by the token norm R, yielding attention logits consistent with the bilinear scale $\kappa R^2/\sqrt{d_k}$. (c) The forward residual-stream norm remains well within the floating-point range at the same checkpoint and during the replay, even as the backward pass becomes unstable.

$q > 1$ provides an early signal of instability. Table 1 shows that crossing the $q = 1$ threshold is more critical for certified depth than reducing σ: for any fixed $\sigma > 0$, $q > 1$ reduces the certified depth to $O(\log_q \log M)$, whereas $q = 1$ retains a $\sqrt{\log M/\sigma}$-scale backward-depth budget. Consistent with this threshold, the no-LN Toto 2.0 variant becomes unstable, with the first signal appearing in the gated-MLP sub-block, whose $q = 2$ growth exceeds the $q = 1$ growth of the attention sub-block (Figure 4). The two sub-blocks amplify comparably through layer 10, but the gated MLP overtakes attention in the final two layers as the residual-stream RMS grows large enough for the quadratic $\|x\|^2$ term to dominate the linear $\|x\|$ term.

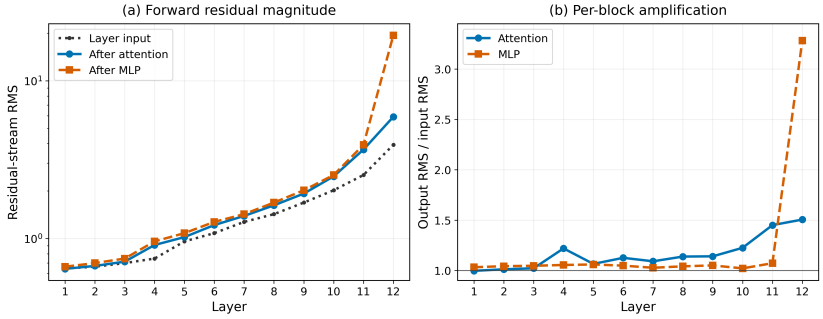


Figure 4: Forward state of the no-LN Toto 2.0 variant at the last step before blow-up across 12 Transformer layers ($D = 24$). (a) Residual-stream RMS at the input of each Transformer layer and after its attention (blue) and gated-MLP (orange) residual sub-blocks, with the two sub-blocks together representing the $D = 24$ residual updates. (b) Per-sub-block amplification, measured as the output-to-input RMS ratio. In the final two layers, amplification reaches 3.3 for the $q = 2$ gated MLP, compared with 1.5 for the $q = 1$ attention. The concentration of amplification in the final gated-MLP blocks in this pre-blow-up snapshot suggests that the $q = 2$ gated MLP may be the trigger of the observed blow-up.

5 Conclusion and Discussion

Our analysis reveals a principle for residual stability under conditions weaker than global Lipschitz constraints. By separating state growth from state-dependent sensitivity through the exponents q and σ, the resulting certificate turns architecture search from trial-and-error into a principled and targeted process. A natural direction for future work is to develop automated tools that propagate primitive exponents through a model definition, scan architectures for potentially problematic blocks, and suggest targeted modifications before training. Another direction is to apply the certificate to residual-scaling schemes such as ReZero, Fixup, and LayerScale (Bachlechner et al., 2021; Zhang et al., 2019; Touvron et al., 2021), whose learned factors play the role of Δt.

6 AI USE STATEMENT

We used large language models in three ways, in every case under the supervision of an author. First, for *writing assistance*: polishing prose, tightening exposition, and making notation and terminology consistent across sections. Second, for *retrieval and discovery*: surfacing candidate related work and pointers into the literature. Every reference found this way was checked against its primary source, and no citation appears in this paper that an author has not read and verified. Third, for *drafting*: producing first drafts of expository passages, figure captions, and appendix material from author-supplied outlines, equations, and experimental results.

The research content is the authors' own. The problem formulation, the definitions, the statements of Theorems 2.2, 2.3, 3.1 and B.2 and their proofs, the exponent derivations of App. D.2, the experimental design, and the interpretation of the results were developed by the authors, and every mathematical claim was verified by an author. The authors take full responsibility for the content of this paper, including any errors that remain.

7 REPRODUCIBILITY STATEMENT

Code for the three benchmarks with released source is included in the supplementary material, in three self-contained directories, `tsf_small/`, `fno_battery/` and `nanogpt_openwebtext/`. Each directory contains the training entry points, the measurement and aggregation code, and the tables that the figures of this paper are drawn from; each carries a `README` or `REPRODUCE` file listing the commands in order. Model weights, checkpoints and generated datasets are not included, since all of them are regenerated by the scripts; the files that are included are the ones a reader would otherwise have to retrain to obtain. For the time-series foundation model benchmark, we provide the architecture and model configuration rather than the training code.

Forecasting (Section 4.1). ETTm1 (Zhou et al., 2021) and Weather (Wu et al., 2021) are public. The two backbones are the CMamba forecaster (Zeng et al., 2024), which the code registers under the name `Mamba`, and PatchTST (Nie et al., 2023), in the variants of Table 6; `scripts/long_term_forecast/` holds one script per dataset and backbone. Seeds are 2021–2026 on ETTm1 and 2021–2023 on Weather, giving 4 horizons $\times$ 2 depths $\times$ 9 seed–dataset pairs per variant. Figure 1 and Tables 3 and 7 are built from the spreadsheets in `test_results/`, which are assembled from the per-run test metrics.

Operator learning (Section 4.2). The PDE model, geometry and reference network are the MathWorks battery-module cooling example (The MathWorks, Inc., n.d.), included unmodified under `example/`. The datasets are not public data but finite-element solves, one per parameter combination, regenerated by `datagen/`; the `README` states which scripts produce which set and why the example's own parameter box is regenerated on the local mesh rather than read from the shipped file. Training is five seeds $\times$ two variants per sweep, and the two sweeps reported here are the original box and the wider box at the original grid spacing. The CSVs behind Figure 2, Table 12 and Figure 7 are included, so those can be regenerated without retraining.

Language modeling (Section 4.3). OpenWebText (Gokaslan & Cohen, 2019) is public and is tokenized by `owt_dataset/prepare.py`; the resulting binary is not shipped. The model is nanoGPT GPT-2 124M (Radford et al., 2019; Karpathy, 2022) with `use_layernorm` added, so that the no-LN variant replaces `ln_1`, `ln_2` and `ln_f` by identities. `REPRODUCE.md` lists the six stages from tokenization through the four CSVs that Figure 3 reads. Divergence replays are not reproducible to the iteration: disabling compilation changes the numerics and the per-iteration probe is itself a perturbation, so a replay trips wherever it trips, and the analysis is indexed to the detected onset rather than to an absolute iteration.

Time-series foundation model (Section 4.4). The Toto 2.0 architecture is publicly available from Datadog. We use the 4m parameter model as a starting point and extend its depth from 4 layers to 12. The 4m parameter model, and our extension, use an internal dimension of 256, 4 attention heads, an MLP width of 688, and a patch size of 32. The training data is a mix of real pretraining data, publicly available from the GIFT-Eval repository, and synthetic data generated with the open-source

library TempoPFN. Code for this experiment is not included in the supplementary material; instead, we provide the architecture setup needed to reproduce the model configuration.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

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

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Henry Mao, Gary Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, 2021.

John S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Françoise Fogelman Soulié and Jeanny Hérault (eds.), *Neurocomputing*, pp. 227–236. Springer, 1990.

Hoyoon Byun, Youngjun Choi, Taero Kim, Sungrae Park, and Kyungwoo Song. Bounded hyperbolic tangent: A stable and efficient alternative to pre-layer normalization in large language models. *arXiv preprint arXiv:2601.09719*, 2026.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

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

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Andrej Karpathy. nanoGPT. `https://github.com/karpathy/nanoGPT`, 2022.

Emaad Khwaja, Chris Lettieri, Gerald Woo, Eden Belouadah, Marc Cenac, Guillaume Jarry, Enguerrand Paquin, Xunyi Zhao, Viktoriya Zhukov, Othmane Abou-Amal, Chenghao Liu, Ameet Talwalkar, and David Asker. Toto 2.0: Time series forecasting enters the scaling era. *arXiv preprint arXiv:2605.20119*, 2026.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5562–5571, 2021.

Jeonghoon Kim, Byeongchan Lee, Cheonbok Park, Yeontaek Oh, Beomjun Kim, Taehwan Yoo, Seongjin Shin, Dongyoon Han, Jinwoo Shin, and Kang Min Yoo. Peri-LN: Revisiting normalization layer in the transformer architecture. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30400–30436, 2025.

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.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Toan Q. Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation*, 2019.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

The MathWorks, Inc. 3-D battery module cooling analysis using Fourier neural operator. Partial Differential Equation Toolbox documentation, `https://www.mathworks.com/help/pde/ug/3-d-battery-module-cooling-analysis-using-fourier-neural-operator.html`, n.d. Accessed September 2026.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingzhou Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 2021.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanhong Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10524–10533, 2020.

Chaolv Zeng, Zhanyu Liu, Guanjie Zheng, and Linghe Kong. CMamba: Channel correlation enhanced state space models for multivariate time series forecasting. *arXiv preprint arXiv:2406.05316*, 2024.

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

Hongyi Zhang, Yann N. Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. In *International Conference on Learning Representations*, 2019.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.

Jiachen Zhu, Xinlei Chen, Kaiming He, Yann LeCun, and Zhuang Liu. Transformers without normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

A PROOFS FOR SECTION 2

Throughout, $a(x) := c\|x\|^q + b$ denotes the growth bound of Assumption 2.1, $r_k = \|X_k\|$, and a block *saturates* at x if $\|v(x;\theta)\| = a(x)$.

A.1 PROOF OF THEOREM 2.2

Step 1 (scalar comparison). By the triangle inequality and Assumption 2.1, $r_{k+1} \le r_k + \Delta t\,(cr_k^q + b) =: \phi(r_k)$. Since ϕ is nondecreasing on $[0,\infty)$, r_k is bounded by the sequence that starts at r_0 and satisfies the recursion with equality.

Step 2 ($q = 1$). The recursion is affine, $r_{k+1} \le (1 + c\Delta t)r_k + b\Delta t$; with $y_k = r_k + b/c$, $y_D \le (1 + c\Delta t)^D y_0$, which is (ii).

Step 3 ($q < 1$). Concavity of $t \mapsto t^{1-q}$ gives $(\alpha + h)^{1-q} \le \alpha^{1-q} + (1-q)\alpha^{-q}h$ for $\alpha > 0$, $h \ge 0$. With $\alpha = r_k$ and $h = c\Delta t\, r_k^q$ (case $b = 0$), $r_{k+1}^{1-q} \le r_k^{1-q} + c(1-q)\Delta t$, which telescopes. For $b \ge 0$, let $\bar r_k = \max(r_k, 1)$. Then $b \le b\,\bar r_k^q$, so $r_{k+1} \le \bar r_k + (c+b)\Delta t\,\bar r_k^q$, and the right side is at least 1, so $\bar r_{k+1}$ obeys the same inequality. This is the case $b = 0$ with $c + b$ in place of c, which gives (i).

Step 4 ($q > 1$). Let $b = 0$, $\rho = (c\Delta t)^{-1/(q-1)}$, and $u_k = 1 + r_k/\rho \ge 1$. Since $c\Delta t\,\rho^{q-1} = 1$, Step 1 gives $u_{k+1} \le u_k + (u_k - 1)^q \le u_k + u_k^q \le 2u_k^q$. With $s_k = \log u_k$, $s_{k+1} + \frac{\log 2}{q-1} \le q\big(s_k + \frac{\log 2}{q-1}\big)$, so $\log u_D \le q^D\big(\log u_0 + \frac{\log 2}{q-1}\big)$, and $r_D \le \rho u_D$ gives (iii). For $b > 0$, the reduction of Step 3 applies.

Attainment. The aligned block $v^\sharp(x) = a(x)\,x/\|x\|$ saturates the bound and is a positive multiple of x, so Step 1 holds with equality: $r_{k+1} = r_k + \Delta t\,(cr_k^q + b)$. At $q = 1$ this is (ii) with equality. At $q \in [0,1)$ with $b = 0$, the sequence is nondecreasing, and $r_{k+1}/r_k = 1 + c\Delta t\, r_k^{q-1} \le \kappa := 1 + c\Delta t\, r_0^{q-1}$. Concavity with the tangent at r_{k+1} gives $r_{k+1}^{1-q} - r_k^{1-q} \ge (1-q)r_{k+1}^{-q}(r_{k+1} - r_k) = (1-q)c\Delta t\,(r_k/r_{k+1})^q \ge (1-q)c\Delta t\,\kappa^{-q}$, so $r_D^{1-q} \ge r_0^{1-q} + (1-q)c\Delta t\,\kappa^{-q}D$, and r_D grows like $D^{1/(1-q)}$, the rate of (i). Adding $b > 0$ only increases the sequence. At $q > 1$, $r_{k+1} \ge c\Delta t\, r_k^q$, and Lemma A.2 below (with $\gamma = c\Delta t$) gives $\liminf_D \log\log r_D/D \ge \log q$ once $r_K \ge e\rho$, which happens at a finite K because r_k increases by at least $\Delta t(cr_0^q + b)$ per step; (iii) gives the matching lim sup. □

A.2 SUPERCRITICAL BLOCKS ARE DRIVEN TO THE WORST CASE

Theorem A.1 (Supercritical blocks are driven to the worst case)**.** *Let $q > 1$ and $R_\star = (2/(c\Delta t))^{1/(q-1)}$.*

(i) *If the blocks saturate at every step and $r_k \ge eR_\star$, then $r_j \ge R_\star \exp(q^{j-k})$ for all $j \ge k$, whatever the direction of v; hence $D^*(M) \le k + \lceil \log_q \log(M/R_\star)\rceil$.*

(ii) *Let $v_0^*, \dots, v_{D-1}^*$ minimize $\mathbb{E}\,G(X_D, y)$ over all C^1 fields satisfying the bound of Assumption 2.1, under the regularity conditions of Proposition A.4, and let $\bar p_{k+1}(x) = \mathbb{E}[p_{k+1} \mid X_k = x]$ along the optimal trajectory. Then, for almost every state x with $\bar p_{k+1}(x) \ne 0$, $v_k^*(x) = -a(x)\,\bar p_{k+1}(x)/\|\bar p_{k+1}(x)\|$; in particular $\|v_k^*(x)\| = a(x)$.*

Part (ii) is a discrete Pontryagin maximum principle: the optimal block saturates, so part (i) applies on every layer where the conditional costate is nonzero; a parametric family inherits this when it can represent the saturating field on the data. At $c = \Delta t = 1$ and $q \ge 2$, $eR_\star \le 2e$, so typical inputs meet the hypothesis of (i).

Proof of part (i). Two lemmas do the work: the first turns a superlinear recursion into doubly exponential growth, the second shows that once $r_k \ge R_\star$ a saturated step cannot reduce the radius, whatever the direction of v.

Lemma A.2 (Doubly exponential growth)**.** *Let $q > 1$, $\gamma > 0$, $\rho = \gamma^{-1/(q-1)}$, and let $r_{j+1} \ge \gamma r_j^q$ for $j \ge K$ with $r_K \ge e\rho$. Then $r_j \ge \rho \exp(q^{j-K})$ for all $j \ge K$.*

Proof. With $z_j = \log(r_j/\rho)$ and $\gamma\rho^{q-1} = 1$, the hypothesis reads $z_{j+1} \geq qz_j$ with $z_K \geq 1$; induction gives $z_j \geq q^{j-K}$. □

Lemma A.3 (One saturated step)**.** *Let $q > 1$ and $\|v(X_k)\| = a(X_k)$. Then $|\Delta t\, a(X_k) - r_k| \leq r_{k+1} \leq \Delta t\, a(X_k) + r_k$, with the lower end attained when $v(X_k)$ points to the origin and the upper end when it points away. Hence $r_{k+1} \geq r_k$ for every direction if and only if $\Delta t\, a(X_k) \geq 2r_k$, which for $b = 0$ means $r_k \geq R_\star$; and whenever $r_k \geq R_\star$, $r_{k+1} \geq \frac{1}{2}c\Delta t\, r_k^q + \Delta t\, b$.*

Proof. The two bounds are the reverse and direct triangle inequalities for $X_k + \Delta t\, v(X_k)$. The lower bound is at least r_k iff $\Delta t\, a(X_k) \geq 2r_k$, and $\Delta t\, cr_k^{q-1} \geq 2$ iff $r_k \geq R_\star$. If $r_k \geq R_\star$, then $r_k \leq \frac{1}{2}c\Delta t\, r_k^q$, so $r_{k+1} \geq c\Delta t\, r_k^q + \Delta t\, b - r_k \geq \frac{1}{2}c\Delta t\, r_k^q + \Delta t\, b$. □

Proof of Theorem A.1(i). By Lemma A.3, whenever $r_j \geq R_\star$ we have $r_{j+1} \geq \frac{1}{2}c\Delta t\, r_j^q \geq R_\star$, because $r \mapsto \frac{1}{2}c\Delta t\, r^q$ is increasing and equals $R_\star$ at $R_\star$. So the trajectory stays above $R_\star$, and Lemma A.2 with $\gamma = \frac{1}{2}c\Delta t$, for which $\gamma^{-1/(q-1)} = R_\star$, gives $r_j \geq R_\star \exp(q^{j-k})$. This exceeds M once $q^{j-k} \geq \log(M/R_\star)$. □

The threshold $R_\star$ applies even to blocks that point toward the origin. For the inward field $v(x) = -c\|x\|^{q-1}x$, one step multiplies X_k by $1 - \tau_k$ with $\tau_k = c\Delta t\, r_k^{q-1}$. The state decays to the origin from below $R_\star$ ($\tau_k < 2$), but from above $R_\star$ it overshoots the origin and escapes doubly exponentially.

Discrete maximum principle and proof of part (ii). Let $\mathcal{V}^q$ be the set of C^1 fields $u : \mathbb{R}^N \to \mathbb{R}^N$ with $\|u\| \leq a$, and consider

$$\min_{v_0,\dots,v_{D-1}\in\mathcal{V}^q} \mathbb{E}\, G(X_D, y), \qquad X_{k+1} = X_k + \Delta t\, v_k(X_k), \quad (X_0, y) \sim \mu_0. \tag{11}$$

The set $\mathcal{V}^q$ is convex and contains every C^1 field bounded by a. We assume: (P1) $b > 0$; (P2) a minimizer $v_0^*, \dots, v_{D-1}^*$ exists, with trajectory X_k^* whose law is μ_k; (P3) μ_0 has compact support; (P4) $G(\cdot, y)$ is C^1 with $\nabla_X G$ bounded on compact sets.

Proposition A.4 (Discrete maximum principle)**.** *Under (P1)–(P4), let p_k follow* (2) *along the optimal trajectory and $\bar{p}_{k+1}(x) = \mathbb{E}[p_{k+1} \mid X_k^* = x]$. Then $\int \langle \bar{p}_{k+1}(x), u(x) - v_k^*(x)\rangle\, d\mu_k(x) \geq 0$ for every $u \in \mathcal{V}^q$, and for μ_k-a.e. x with $\bar{p}_{k+1}(x) \neq 0$,*

$$v_k^*(x) = -a(x)\,\frac{\bar{p}_{k+1}(x)}{\|\bar{p}_{k+1}(x)\|}, \qquad \textit{so} \qquad \|v_k^*(x)\| = a(x).$$

Proof. Step 1 (co-state). With $\Psi_D = I$ and $\Psi_j = \Psi_{j+1}(I + \Delta t\, \nabla_X v_j^*(X_j^*))$, the recursion (2) gives $p_j = \Psi_j^\top \nabla_X G(X_D^*, y)$.

Step 2 (first variation). Fix $u \in \mathcal{V}^q$ and $\varepsilon \in [0, 1]$, and set $v_k^\varepsilon = v_k^* + \varepsilon(u - v_k^*) \in \mathcal{V}^q$, leaving the other layers unchanged. Then $X_{k+1}^\varepsilon - X_{k+1}^* = \varepsilon\Delta t\,(u - v_k^*)(X_k^*)$, and propagation through the remaining C^1 layers gives $X_D^\varepsilon - X_D^* = \varepsilon\Delta t\, \Psi_{k+1}(u - v_k^*)(X_k^*) + o(\varepsilon)$. By (P3) and $\|u\| \leq a$, all perturbed trajectories lie in one compact set, on which $\nabla_X G$ and $\nabla_X v_j^*$ are bounded; dominated convergence then gives

$$\frac{d}{d\varepsilon}\Big|_{0^+} \mathbb{E}\, G(X_D^\varepsilon, y) = \Delta t\, \mathbb{E}\langle p_{k+1}, (u - v_k^*)(X_k^*)\rangle = \Delta t \int \langle \bar{p}_{k+1}(x), u(x) - v_k^*(x)\rangle\, d\mu_k(x),$$

where the second equality conditions on X_k^*. Optimality makes the right derivative nonnegative.

Step 3 (localization). For $\varphi \in C^1$ with $0 \leq \varphi \leq 1$, the competitor $v_k^* + \varphi(u - v_k^*)$ is admissible, so the inequality holds with the integrand multiplied by φ. Letting $\varphi \uparrow \mathbf{1}_B$ for open balls B and applying the Besicovitch differentiation theorem to μ_k gives $\langle \bar{p}_{k+1}(x), u(x) - v_k^*(x)\rangle \geq 0$ for μ_k-a.e. x.

Step 4 (saturation). Let (e_m) be dense in the unit sphere, and let $a_n \in C^1$ with $a_n \leq a$ and $a_n \uparrow a$ pointwise (take $a_n = a$ when a is C^1, i.e., $q = 0$ or $q > 1$; for $0 < q \leq 1$ take $a_n(x) =$

$c(\|x\|^2 + n^{-1})^{q/2} - cn^{-q/2} + b)$. Step 3 with the countably many $u = a_n e_m \in \mathcal{V}^q$ gives, for μ_k-a.e. x, $\langle \bar{p}_{k+1}(x), v_k^*(x)\rangle \le a_n(x)\langle \bar{p}_{k+1}(x), e_m\rangle$ for all n, m; letting $n \to \infty$ and minimizing over m yields $\langle \bar{p}_{k+1}(x), v_k^*(x)\rangle \le -a(x)\|\bar{p}_{k+1}(x)\|$. Since $\|v_k^*(x)\| \le a(x)$, Cauchy–Schwarz forces $v_k^*(x) = -a(x)\bar{p}_{k+1}(x)/\|\bar{p}_{k+1}(x)\|$ when $\bar{p}_{k+1}(x) \neq 0$. □

Theorem A.1(ii) is Proposition A.4. On the event $\bar{p}_{j+1}(X_j^*) \neq 0$ for $k \le j < D$, the optimal blocks saturate on layers $k, \dots, D-1$, and part (i) applies there. A parametric family $\{v(\cdot;\theta)\}$ is generally not convex, so Step 2 does not apply to it directly; the conclusion transfers when the family can represent the saturating field on the support of the data. Both factors of the conclusion can be monitored during training, as the ratio $\|v(X_k)\|/a(X_k)$ and the cosine between $v(X_k)$ and $-p_{k+1}$.

A.3 Backward bound and proof of Theorem 2.3

The proof uses only $\|X_k\| \le \mathcal{R}_k$, which holds for every $q \ge 0$ by Theorem 2.2; for $q > 1$ the envelope is finite at every depth, so no restriction on q is needed. By (2), $p_i = J_{i:D}^\top p_D$ with $J_{i:D} = \prod_{k=D-1}^{i}(I + \Delta t\, \nabla_X v(X_k;\theta_k))$, and $\|p_D\| \le \mathrm{Lip}(G)$, the Lipschitz constant of $G(\cdot, y)$ on the ball of radius $\mathcal{R}_D$. Submultiplicativity, $\|I + \Delta t\, A\|_2 \le 1 + \Delta t\|A\|_2$, and $\|\nabla_X v(X_k;\theta_k)\|_2 \le L(\mathcal{R}_k) \le L(\mathcal{R}_D)$ give

$$\begin{aligned}\|p_i\| &\overset{\text{(T)}}{\le} \prod_{k=i}^{D-1}\big(1 + \Delta t\,\|\nabla_X v(X_k;\theta_k)\|_2\big)\mathrm{Lip}(G)\\ &\overset{\text{(A)}}{\le} \prod_{k=i}^{D-1}\big(1 + \Delta t\, L(\mathcal{R}_k)\big)\mathrm{Lip}(G) \;\le\; \big(1 + \Delta t\, L(\mathcal{R}_D)\big)^{D-i}\mathrm{Lip}(G),\end{aligned} \tag{12}$$

whose last member is (6). The trajectory bound (T) uses the Jacobians actually visited, an a posteriori quantity; step (A) replaces each by its worst case over $\theta \in \Theta$ and the ball of radius $\mathcal{R}_k$, an outer bound on the reachable region. Hence $\|\nabla_X v(X_k;\theta_k)\|_2 \le L(\mathcal{R}_k) \le L(\mathcal{R}_D) \le \sup_R L(R)$, from trajectory-specific to global, and $L(\mathcal{R}_D)$ can be loose by many orders of magnitude (Section 4.3). For $v(x) = Lx$ with $L \le c$, the block satisfies Assumption 2.1 at $q = 1$, $\nabla_X v \equiv LI$, and every inequality is an equality for $G(x, y) = \langle e, x\rangle$ with a unit vector e, so the certificate is sharp.

Proof of Theorem 2.3. $(1 + \Delta t\, L(\mathcal{R}_D))^D \mathrm{Lip}(G) < M$ if and only if (7) holds, and $(1 + \Delta t\, L(\mathcal{R}_D))^{D-i} \le (1 + \Delta t\, L(\mathcal{R}_D))^D$, so (6) gives $\|p_i\| < M$ for every i. □

Budget. With $\mathrm{Lip}(G) = 1$ in FP32 ($\log M = 88.7$) at $\Delta t = 1$, the right side of (7) is 39.3 at $D = 24$ and 1.52 at $D = 96$. Since $\mathcal{R}_D$ and $L(\mathcal{R}_D)$ are nondecreasing in D and the budget decreases, (7) holds for every $D \le D_\nabla$.

Layerwise refinement. The middle member of (12) replaces $L(\mathcal{R}_D)$ by the smaller $L(\mathcal{R}_k)$ in each layer. It certifies at least the same depth, improves the constant at $q = 1$ by a factor $\sqrt{2}$ (Proposition A.5), and is what Section 4.3 evaluates.

A.4 Radius dependence of the backward bound

Proposition A.5 (Radius dependence of the backward bound)**.** *Let Assumptions 2.1 and 2.4 hold. With $\mathcal{R}_k$ as in Theorem 2.3:*

(i) *($q < 1$)* $\log(\|p_0\|/\mathrm{Lip}(G)) \le D\log(1+\Delta t(\ell_0+\ell_1)) + \frac{\sigma}{1-q} D\log\big(\bar{r}_0^{\,1-q} + (1-q)(c+b)D\Delta t\big)$, *which stays below* $\log M$ *for* $D \le \frac{1-q}{\sigma}\frac{\log M}{\log\log M}(1 + o(1))$ *as* $M \to \infty$*;*

(ii) *($q = 1$)* $\log(\|p_0\|/\mathrm{Lip}(G)) \le D\big[\log(1 + \Delta t(\ell_0 + \ell_1)) + \sigma \log^+(r_0 + b/c)\big] + \frac{\sigma}{2}\log(1 + c\Delta t)\, D(D-1)$, *which stays below* $\log M$ *for* $D \le \sqrt{2\log M/(\sigma\log(1 + c\Delta t))}\,(1 + o(1))$*;*

(iii) *($q > 1$, $b = 0$) with* $\rho = (c\Delta t)^{-1/(q-1)}$ *and* $C_q = \log\big(2^{1/(q-1)}(1 + r_0/\rho)\big) > 0$, $\log(\|p_0\|/\mathrm{Lip}(G)) \le D\big[\log(1 + \Delta t(\ell_0 + \ell_1)) + \sigma\log^+\rho\big] + \sigma C_q \frac{q^D - 1}{q-1}$, *which stays below* $\log M$ *for* $D \le \log_q \log M - K$*, with K depending only on $q, \sigma, c\Delta t, r_0, \ell_0, \ell_1$; for $b > 0$ the same holds with $c + b$ and $\bar{r}_0$ in place of c and r_0.*

Proof. Put $\bar{\mathcal{R}}_k = \max(\mathcal{R}_k, 1)$. For $R \geq 1$, $1 + \Delta t(\ell_1 R^\sigma + \ell_0) \leq (1 + \Delta t(\ell_0 + \ell_1))R^\sigma$, so $\log(1 + \Delta t\, L(\mathcal{R}_k)) \leq \log(1 + \Delta t(\ell_0 + \ell_1)) + \sigma \log \bar{\mathcal{R}}_k$; sum over $k < D$ in the middle member of (12). At $q < 1$, $\bar{\mathcal{R}}_k \leq \mathcal{R}_D$ since $\bar{r}_0 \geq 1$. At $q = 1$, $\log \bar{\mathcal{R}}_k \leq k \log(1 + c\Delta t) + \log^+(r_0 + b/c)$ and $\sum_{k<D} k = D(D-1)/2$. At $q > 1$, $\log \bar{\mathcal{R}}_k \leq \log^+ \rho + q^k C_q$ by Theorem 2.2(iii), and $\sum_{k<D} q^k = (q^D - 1)/(q - 1)$; with $D = \log_q \log M - K$ the last term is at most $\sigma C_q \log M/((q-1)q^K)$, below $\frac{1}{2} \log M$ for K large, and the linear term is $O(\log \log M)$. The right sides of (i) and (ii) are $\frac{\sigma}{1-q} D \log D(1 + o(1))$ and $\frac{\sigma}{2} \log(1 + c\Delta t)D^2(1 + o(1))$; solving for the depth at which each equals $\log M$ gives the ranges. □

Against the linear $D \log(1 + \Delta t\, L)$ of a bounded Jacobian, the exponent of the backward bound is $D \log D$ at $q < 1$, quadratic in D at $q = 1$, and exponential in D at $q > 1$; the last requires $\sigma > 0$, not $q > 1$ alone. Both are upper envelopes; equality requires $L(R) \asymp R^\sigma$ and Jacobians aligned along the trajectory. At $\sigma = 2$, (ii) in FP32 at $c = \Delta t = 1$ gives about $\sqrt{2 \cdot 88.7/(2 \log 2)} \approx 11$ layers, against 128 for a bounded Jacobian with $L = 1$.

Proposition A.6 (An attained exponent forces radius dependence)**.** *Let Assumptions 2.1 and 2.4 hold, and suppose that for some $\theta \in \Theta$, unit vector e, and $c' > 0$, $b' \geq 0$, $\|v(se; \theta)\| \geq c' s^q - b'$ for all $s > 0$. Then $L(R) \geq c' R^{q-1} - (b + b')/R$ for all $R > 0$. In particular $q > 1$ attained along a ray implies $L(R) \to \infty$ with $\sigma \geq q - 1$ in* (8)*. Without attainment, Assumption 2.1 gives no lower bound on L.*

Proof. By the mean value inequality along the segment from 0 to Re, $\|v(Re; \theta)\| \leq \|v(0; \theta)\| + L(R)R \leq b + L(R)R$. With the lower bound at $s = R$, $c' R^q - b' \leq b + L(R)R$. For the last claim, $v \equiv 0$ satisfies Assumption 2.1 for every q and has $L \equiv 0$. □

A.5 Self-attention has $\sigma = 2$; normalization removes it

Lemma A.7 (Self-attention: $q = 1$ and $\sigma = 2$)**.** *Let $\mathrm{Attn}(X) = W_V X\, A(X)$ on $X \in \mathbb{R}^{d \times n}$, with $A(X) = \mathrm{softmax}(X^\top W_Q^\top W_K X/\sqrt{d_k})$ taken over keys for each query, so that A is column-stochastic, and let $\bar{w}_V = \sup_\Theta \|W_V\|_2$ and $\kappa = \sup_\Theta \|W_Q^\top W_K\|_2$. Then $\|\mathrm{Attn}(X)\|_F \leq \sqrt{n}\, \bar{w}_V \|X\|_F$, and on $\|X\|_F \leq R$,*

$$L_{\mathrm{attn}}(R) \leq \sqrt{n}\, \bar{w}_V + \frac{\bar{w}_V\, \kappa}{\sqrt{d_k}}\, R^2, \tag{13}$$

where $\ell_0 = \sqrt{n}\, \bar{w}_V$ is the value-path constant and $\ell_1 = \bar{w}_V \kappa/\sqrt{d_k}$, with κ the norm of the bilinear form that produces the logits. Hence $\sigma = 2$.

Proof. Two terms. A column-stochastic A has $\|A\|_2 \leq \|A\|_F \leq \sqrt{n}$, which gives the growth bound with $q = 1$. Writing $S = X^\top W_Q^\top W_K X/\sqrt{d_k}$ for the logits and differentiating,

$$\nabla_X \mathrm{Attn}(X)[\delta] = \underbrace{W_V\, \delta\, A(X)}_{\text{value path}} + \underbrace{W_V X\, \partial A\big[\partial S[\delta]\big]}_{\text{attention path}}. \tag{14}$$

The value path is bounded by $\sqrt{n}\, \bar{w}_V \|\delta\|_F$ uniformly in X and contributes no growth.

Why the exponent is 2*.* The attention path is a product of three factors, and its degree in X is the sum of their degrees:

(a) the prefactor $W_V X$ is left undifferentiated when the derivative falls on A, and is linear in X, so $\deg_X = 1$;

(b) S is *bilinear* in X, so $\partial S[\delta] = (\delta^\top W_Q^\top W_K X + X^\top W_Q^\top W_K \delta)/\sqrt{d_k}$ is linear in X, so $\deg_X = 1$;

(c) $\partial A/\partial S$ is bounded uniformly in its own input, so $\deg_X = 0$.

Hence $\sigma \leq 1 + 1 + 0 = 2$. The second power of R therefore comes from the value stream and from differentiating a quadratic form, one factor each; it does not come from the logits being quadratic,

since differentiation lowers that degree to one. Only (c) is a property of the softmax, and what it contributes is boundedness, not the value of the bound.

The constants. $\|\operatorname{diag}(\alpha) - \alpha\alpha^\top\|_2 \le \frac{1}{2}$ for α in the simplex and the map is block diagonal over queries, so $\|\partial A/\partial S\|_2 \le \frac{1}{2}$; $\|\partial S[\delta]\|_F \le 2\kappa\|X\|_F\|\delta\|_F/\sqrt{d_k}$; and $\|W_V X\|_2 \le \bar{w}_V\|X\|_F$. The product of the three gives $\ell_1 = \bar{w}_V\kappa/\sqrt{d_k}$. For h heads with output projections W_O^h, the same argument gives $L(R) \le \sum_h \big(\sqrt{n}\,\|W_O^h W_V^h\|_2 + \|W_O^h W_V^h\|_2 \kappa_h R^2/\sqrt{d_k}\big)$. □

The exponent is attained; the coefficient is not. Scaling $X \mapsto sX$ multiplies every logit by s^2, so a tie between the two largest logits of a query is preserved along the entire ray: the set on which the softmax does not saturate is a scale-invariant cone, present in every ball $\|X\|_F \le R$. On it the factor $\|\partial A/\partial S\|_2$ stays at its maximum $\frac{1}{2}$ and both degree-one factors grow linearly, so $L_{\mathrm{attn}}(R) = \Theta(R^2)$ and $\sigma = 2$ is attained rather than an artifact of the bound; in a numerical check with $d = d_k = 6$, $n = 2$ and Gaussian weights, $d\log L/d\log R = 1.9994$ over four decades along such a ray. Along a generic ray the logit gap grows like R^2, the softmax saturates, and L settles at exactly $\ell_0 = \sqrt{n}\,\bar{w}_V$ with slope 0; the two regimes are separated by an angular width that shrinks like R^{-2}, since an angular logit gap g produces an actual gap $\approx gR^2$. This is why the worst case $L(R)$ can greatly exceed the Jacobians visited in (T), and why a trajectory-based estimate of σ (App. C) reads low.

The coefficient is a different matter: $\frac{1}{2}$ is sharp, attained at a two-way tie, but the factor 2 in $\|\partial S\|$ requires both Leibniz terms to align with the same δ and is not, and on the tied ray above $L_{\mathrm{attn}}(R)/R^2$ settles at 0.100 against $\ell_1 = 1.47$. Coefficients such as ℓ_1 should therefore be read as upper bounds rather than estimates, and margins computed from them, including those in App. D.5, are conservative.

Normalization removes the radius dependence. For a block $f \circ N$ preceded by RMSNorm or ScaleNorm $N(x) = g\,x/\|x\|$ (with $\epsilon = 0$), $\nabla N(x) = g(I - \hat{x}\hat{x}^\top)/\|x\|$ with $\hat{x} = x/\|x\|$, so $\|\nabla(f \circ N)(x)\|_2 \le \mathrm{Lip}(f|_{\|z\|\le|g|})\,|g|/\|x\|$, where f is evaluated only on the bounded range of N. On $\{\|x\| \ge r_0\}$ the Jacobian is therefore bounded by its value at the inner radius, uniformly in R ($\sigma = 0$); measured at single points of $\|X\| = 4, 16, 64, 256, 1024$ for one draw of a pre-normalized attention block, it is $1.6, 0.39, 0.099, 0.025, 0.0062$, consistent with $\|X\|^{-1}$. LayerNorm divides by the norm of the centered input, which can be small at large $\|x\|$, so the same statement holds for it only on the set where the centered norm is bounded below.

A.6 Complementarity of the two limits

Proposition A.8 (The growth bound does not constrain the Jacobian)**.** *For every $q \ge 0$, $c > 0$, $b \ge 0$, and $R > 0$, $\sup\{\|\nabla_X v(x)\|_2 :\ v \in C^1,\ \|v\| \le a,\ \|x\| = R\} = \infty$.*

Proof. Fix a unit vector e and let $v^\lambda(x) = a(x)\sin(\lambda\|x\|^2)\,e$, so $\|v^\lambda\| \le a$. Writing $v^\lambda(x) = f(\|x\|)e$ with $f(r) = (cr^q + b)\sin(\lambda r^2)$, $\nabla_X v^\lambda(x) = f'(\|x\|)\,e\,(x/\|x\|)^\top$ for $x \ne 0$, of spectral norm $|f'(\|x\|)|$. As $r \downarrow 0$, $f(r) = O(r^2)$ and $f'(r) = cq\,r^{q-1}\sin(\lambda r^2) + 2\lambda r(cr^q + b)\cos(\lambda r^2) = O(r^{q+1}) + O(r) \to 0$, so $v^\lambda \in C^1$ with $\nabla_X v^\lambda(0) = 0$. For $\lambda \in 2\pi\mathbb{N}/R^2$, $\sin(\lambda R^2) = 0$ and $\cos(\lambda R^2) = 1$, so $\|\nabla_X v^\lambda(x)\|_2 = 2\lambda R(cR^q + b)$ on $\|x\| = R$, which is unbounded in λ. □

Proposition A.9 (The two limits are complementary)**.** (a) *No backward condition compensates a small D_F at $q > 1$. Let $q > 1$ and $M > R_\star$. On the event of Theorem A.1(ii), if the optimal trajectory has $r_k \ge eR_\star$ at a layer k with $k + \lceil \log_q \log(M/R_\star)\rceil \le D$, then $\|X_j\| \ge M$ for some $j \le D$, and X_D and every p_i are* `NaN`*, whatever the Jacobians.*

(b) *D_F does not bound D_∇. Let $q \ge 0$, $0 < r_* \le r_0$, $\|x_*\| = r_*$, $\hat{x}_* = x_*/r_*$, $\lambda \in 2\pi\mathbb{N}/r_*^2$, and $G(x, y) = \langle \hat{x}_*, x\rangle$. The block $v(x) = a(x)\sin(\lambda\|x\|^2)\hat{x}_*$ satisfies Assumption 2.1; from $X_0 = x_*$ we get $X_k \equiv x_*$, so the state never grows, and $\|p_i\| = (1 + 2\lambda\Delta t\,a(x_*)r_*)^{D-i}$. Hence $\|p_{D-1}\| \ge M$ for λ large, while D_F does not depend on λ.*

(c) *D_∇ does not bound D_F. If $L(\infty) < \infty$, then Assumption 2.1 holds with $q = 1$, and $D_\nabla \ge \lceil \log(M/\mathrm{Lip}G)/\log(1 + \Delta t\,L(\infty))\rceil - 1$ for every q with which the growth bound is stated, whereas D_F decreases as q or c increases.*

(d) Both limits are defined for $q > 1$. *For every $q \geq 0$ and D, $\mathcal{R}_D(q) < \infty$ and $L(\mathcal{R}_D(q)) < \infty$; for $q > 1$, $D_F \leq \log_q \log(M/\rho) + O(1)$ for the envelope of Theorem 2.2(iii), and D_∇ may be smaller.*

Proof. (a) follows from Theorem A.1: $r_j \geq R_\star \exp(q^{j-k}) \geq M$ once $q^{j-k} \geq \log(M/R_\star)$, and a `NaN` state makes $p_D = \nabla_X G(X_D, y)$ and every p_i `NaN`. (b) The block is admissible and C^1 by the proof of Proposition A.8. Since $\sin(\lambda r_*^2) = 0$, $v(x_*) = 0$ and $X_k \equiv x_*$; since $\cos(\lambda r_*^2) = 1$, $\nabla_X v(x_*) = 2\lambda a(x_*) r_* \, \hat{x}_* \hat{x}_*^\top$, so each step of (2) multiplies $p_D = \hat{x}_*$ by $1 + 2\lambda \Delta t \, a(x_*) r_*$. (c) By the mean value inequality, $\|v(x)\| \leq \|v(0)\| + L(\infty)\|x\|$; the bound on D_∇ is Theorem 2.3 with $L(\mathcal{R}_D) \leq L(\infty)$. (d) Finiteness is Theorem 2.2 and Assumption 2.4; the bound on D_F is obtained by solving $\mathcal{R}_D = M$ in (iii) for D. □

A.7 Overflow depths

Table 4: Overflow depth in layers at $\Delta t = c = r_0 = \mathrm{Lip}(G) = 1$ and $b = 0$: forward for the aligned block, backward the depth $D_\nabla^*(M) = \lceil \log M / \log(1 + \Delta t \, L) \rceil = D_\nabla + 1$ at which the bound of Theorem 2.3 with a bounded Jacobian first reaches M. The $q = \frac{1}{2}$ entries beyond FP16 are the bound of Theorem 2.2(i), which is within 1% of the exact depth at FP16.

	FP16 $M = 65{,}504$	FP32 / BF16 $M = 3.40 \times 10^{38}$	FP64 $M = 1.80 \times 10^{308}$
Forward $D^*(M)$			
$q = 0$	65,503	3.4×10^{38}	1.8×10^{308}
$q = 1/2$	513	3.7×10^{19}	2.7×10^{154}
$q = 1$	16	128	1,024
$q = 2$	5	8	11
$q = 5$	3	5	6
Backward $D_\nabla^*(M)$			
$L = 1/2$	28	219	1,751
$L = 1$	16	128	1,024
$L = 2$	11	81	647

B Exponent arithmetic: proofs, primitives, and certification

Definition B.1 (Architectural primitives)**.** *The set of* primitives *is $\mathcal{P} := \{B : \mathbb{R}^{N_{\mathrm{in}}} \to \mathbb{R}^{N_{\mathrm{out}}} \mid B$ continuous, and $\exists\, q, c, b \geq 0 : \ \|B(x)\| \leq c\|x\|^q + b \ \forall x\}$, where $N_{\mathrm{in}}, N_{\mathrm{out}}$ are fixed by the architectural context and $\|\cdot\|$ is the Euclidean (Frobenius) norm. For $B \in \mathcal{P}$, the* input-magnitude exponent *$q(B)$ is the infimum of the $q \geq 0$ for which such c, b exist.*

B.1 Proof of Theorem 3.1

Continuity is preserved by each operation, so it suffices to verify the growth bounds. Because $q(B)$ is an infimum, we prove each rule for arbitrary exponents $q_i' > q_i$ (or $q_i' = q_i$ when attained) for which the bound $\|B_i(x)\| \leq c_i \|x\|^{q_i'} + b_i$ holds; letting $q_i' \downarrow q_i$ gives the stated inequalities. Write $\rho = \|x\|$.

(i) $\|B_1(B_2(x))\| \leq c_1(c_2\rho^{q_2'} + b_2)^{q_1'} + b_1 \leq c_1 \max(1, 2^{q_1'-1})\big(c_2^{q_1'}\rho^{q_1' q_2'} + b_2^{q_1'}\big) + b_1$, using $(u+w)^p \leq \max(1, 2^{p-1})(u^p + w^p)$ for $u, w \geq 0$, $p \geq 0$. The leading coefficient for large ρ is proportional to $c_1 c_2^{q_1'}$.

(ii) $\|B_1(x) + B_2(x)\| \leq c_1\rho^{q_1'} + c_2\rho^{q_2'} + b_1 + b_2 \leq (c_1 + c_2)\rho^{\max(q_1', q_2')} + c_1 + c_2 + b_1 + b_2$, since $\rho^p \leq \rho^{p'} + 1$ for $0 \leq p \leq p'$.

(iii) Entrywise, $\|u \odot w\|_2^2 = \sum_i u_i^2 w_i^2 \leq \|u\|_2^2 \|w\|_2^2$, so $\|B_1(x) \odot B_2(x)\| \leq (c_1\rho^{q_1'} + b_1)(c_2\rho^{q_2'} + b_2)$, a sum of powers of ρ with largest exponent $q_1' + q_2'$; the bound in (ii) then gives exponent $q_1' + q_2'$.

(iv) $\|x + \Delta t \, B_1(x)\| \leq \rho + \Delta t(c_1\rho^{q_1'} + b_1)$, which has exponent $\max(1, q_1')$ by (ii).

(v) $\mathrm{LN}(z) = \gamma \odot (z - \mu\mathbf{1})/\sqrt{\sigma^2 + \epsilon} + \beta$ with $\sigma^2 = \|z - \mu\mathbf{1}\|^2/d$, so $\|(z - \mu\mathbf{1})/\sqrt{\sigma^2 + \epsilon}\| \le \sqrt{d}$ and $\|\mathrm{LN}(z)\| \le \|\gamma\|_\infty \sqrt{d} + \|\beta\|$ for every $z \in \mathbb{R}^d$. Applied tokenwise to a $d \times n$ input, the Frobenius norm is at most $\sqrt{n}$ times this constant. Hence $\mathrm{LN} \circ B$ is bounded, continuous for continuous B, and has exponent 0. RMSNorm and ScaleNorm are bounded by $\|\gamma\|_\infty \sqrt{d}$ and $|g|$ in the same way. □

B.2 Arithmetic of Jacobian growth exponents

Restricting $q \le 1$ controls the forward growth class but does not prevent the Jacobian from increasing with the state radius. We therefore track the Jacobian growth exponent at the same primitive level. For a C^1 primitive B, let

$$J_B(R) := \sup_{\|x\| \le R} \|\nabla B(x)\|_2,$$

and let $\sigma(B)$ be the infimum of the $\sigma \ge 0$ for which

$$J_B(R) \le \ell_1 R^\sigma + \ell_0, \qquad R \ge 1,$$

as in (8).

Theorem B.2 (Arithmetic of Jacobian growth exponents)**.** *Let B_1, B_2 be C^1 primitives with $q_i = q(B_i)$ and $\sigma_i = \sigma(B_i)$. Then*

(i) composition: $\sigma(B_1 \circ B_2) \le \sigma_1 q_2 + \sigma_2$*;*

(ii) sum: $\sigma(B_1 + B_2) \le \max(\sigma_1, \sigma_2)$*;*

(iii) product: $\sigma(P(B_1, B_2)) \le \max(\sigma_1 + q_2, q_1 + \sigma_2)$*;*

(iv) residual wrapping: $\sigma(I + \Delta t B_1) \le \sigma_1$*;*

(v) pre-normalization: $\sigma(B \circ N) = 0$ *for* $N \in \{\mathrm{LN}, \mathrm{RMSNorm}, \mathrm{ScaleNorm}\}$.

Proof. As in the proof of Theorem 3.1, we prove each rule for exponents $q_i' > q_i$ and $\sigma_i' > \sigma_i$ for which $\|B_i(x)\| \le c_i \|x\|^{q_i'} + b_i$ and $J_{B_i}(R) \le \ell_{1,i} R^{\sigma_i'} + \ell_{0,i}$, and let them decrease to q_i, σ_i. Fix $R \ge 1$ and $\|x\| \le R$; every polynomial in R below is bounded by a constant times its leading power.

(i) $\nabla(B_1 \circ B_2)(x) = \nabla B_1(B_2(x)) \nabla B_2(x)$ and $\|B_2(x)\| \le \rho := c_2 R^{q_2'} + b_2$, so $J_{B_1 \circ B_2}(R) \le J_{B_1}(\rho) J_{B_2}(R) \le (\ell_{1,1} \rho^{\sigma_1'} + \ell_{0,1})(\ell_{1,2} R^{\sigma_2'} + \ell_{0,2})$, of order $R^{\sigma_1' q_2' + \sigma_2'}$.

(ii) $J_{B_1 + B_2} \le J_{B_1} + J_{B_2}$.

(iii) $\nabla P(B_1, B_2)(x)[\delta] = P(\nabla B_1(x)\delta, B_2(x)) + P(B_1(x), \nabla B_2(x)\delta)$, so $J \le J_{B_1}(R)\, \|B_2(x)\| + \|B_1(x)\|\, J_{B_2}(R)$, of order $R^{\max(\sigma_1' + q_2',\, q_1' + \sigma_2')}$. The growth bound $\|P(B_1, B_2)(x)\| \le \|B_1(x)\| \|B_2(x)\|$ gives $q \le q_1 + q_2$ as in Theorem 3.1(iii).

(iv) $\|I + \Delta t\, \nabla B_1\|_2 \le 1 + \Delta t\, J_{B_1}$.

(v) Each N maps into a bounded set, $\|N(x)\| \le \rho_N$ (Theorem 3.1(v)), and with $\epsilon > 0$ its Jacobian is bounded on $\mathbb{R}^d$ by a constant C_N of order $\|\gamma\|_\infty / \sqrt{\epsilon}$ (for ScaleNorm with ϵ added to $\|x\|$, $|g|/\epsilon$). Hence $J_{B \circ N}(R) \le J_B(\rho_N)\, C_N$ for every R, so $\sigma(B \circ N) = 0$. □

The resulting primitive-level values are listed in Table 2; architecture-specific derivations, including self-attention and Mamba, are in App. D.2.

B.3 Well-posedness of the arithmetic

The rules of Theorems 3.1 and B.2 are applied by traversing a computation graph, so two questions must be settled before the exponents of a block are well defined: whether the value depends on the order in which a chain, a sum, or a product of several primitives is combined, and whether it depends on how finely the block is broken into primitives. The first has a negative answer, the second a positive one with a definite direction.

Write $x_i = (q_i, \sigma_i) \in [0,\infty)^2$ for the exponent pair of a primitive B_i, and let

$$x_1 \circ x_2 := (q_1 q_2,\ \sigma_1 q_2 + \sigma_2), \qquad x_1 \oplus x_2 := \big(\max(q_1, q_2),\ \max(\sigma_1, \sigma_2)\big),$$
$$x_1 \otimes x_2 := \big(q_1 + q_2,\ \max(\sigma_1 + q_2,\ q_1 + \sigma_2)\big)$$

be the operations that the two theorems attach to composition, sum, and bilinear product.

Proposition B.3 (Associativity)**.** *The operations $\circ$, $\oplus$, and $\otimes$ are associative on $[0,\infty)^2$; $\oplus$ and $\otimes$ are commutative, and $\circ$ and $\oplus$ have units $(1,0)$ and $(0,0)$. The value of an n-fold chain, sum, or product is therefore independent of the parenthesization, and equals*

$$\bigcirc_{i=1}^{n} x_i = \Big(\prod_{i=1}^{n} q_i,\ \sum_{i=1}^{n} \sigma_i \prod_{j>i} q_j\Big), \qquad \bigoplus_{i=1}^{n} x_i = \big(\max_i q_i,\ \max_i \sigma_i\big),$$
$$\bigotimes_{i=1}^{n} x_i = \Big(\sum_i q_i,\ \max_i\big(\sigma_i + \textstyle\sum_{j\neq i} q_j\big)\Big).$$

Proof. Put $M(x) = \left(\begin{smallmatrix} q & 0 \\ \sigma & 1 \end{smallmatrix}\right)$. Then $M(x_1 \circ x_2) = M(x_1)M(x_2)$, so associativity of $\circ$ is associativity of matrix multiplication and the n-fold formula is the product $M(x_1)\cdots M(x_n)$; the unit is $M^{-1}(I) = (1,0)$, the pair of the identity map. For $\oplus$ both components are $\max$ on a totally ordered set, with unit $(0,0)$, the pair of the zero map. For $\otimes$ the first component is addition; for the second, both bracketings of three factors expand to $\max(\sigma_1 + q_2 + q_3,\ q_1 + \sigma_2 + q_3,\ q_1 + q_2 + \sigma_3)$, and the n-fold formula follows by induction and is symmetric in the factors. The operation $\circ$ is not commutative in its second component, which is immaterial because the graph fixes the order of composition. □

The unary rules, residual wrapping $x \mapsto (\max(1, q), \sigma)$ and normalization, take a single argument and raise no such question. For an n-fold product the underlying map must itself admit rebracketing for the statement to have content; the Hadamard and matrix products used here are associative, and for a general bilinear P the graph fixes the bracketing.

Proposition B.4 (Monotone refinement)**.** *Let T be an expression tree whose leaves are primitives and whose nodes are the operations of Theorems 3.1 and B.2, let F_T be the map it computes, and let $\hat{x}(T) = (\hat{q}(T), \hat{\sigma}(T))$ be the value obtained by evaluating the rules from the leaves to the root. Then*

(i) soundness: *$q(F_T) \le \hat{q}(T)$ and $\sigma(F_T) \le \hat{\sigma}(T)$;*

(ii) monotonicity: *each rule is nondecreasing in each of its arguments;*

(iii) refinement: *if T' is obtained from T by replacing a subtree S with a single leaf carrying $\big(q(F_S), \sigma(F_S)\big)$, then $\hat{x}(T') \le \hat{x}(T)$ componentwise.*

Consequently $q(F_T) \le \hat{q}(T') \le \hat{q}(T)$, and likewise for σ: a coarser decomposition whose atoms carry exactly evaluated exponents never gives a weaker bound than a finer one, and the sharpest exponents the arithmetic delivers are the infima of $\hat{q}$ and $\hat{\sigma}$ over admissible decompositions of the block.

Proof. (i) is Theorems 3.1 and B.2 applied at each node, by induction on the height of T. (ii) is inspection: $q_1 q_2$, $\max(q_1, q_2)$, $q_1 + q_2$, $\sigma_1 q_2 + \sigma_2$, $\max(\sigma_1 + q_2, q_1 + \sigma_2)$, $\max(\sigma_1, \sigma_2)$, $\max(1, q_1)$ and σ_1 are all nondecreasing in each argument on $[0,\infty)$. For (iii), (i) applied to S gives $\big(q(F_S), \sigma(F_S)\big) \le \hat{x}(S)$ componentwise, and (ii) propagates this inequality along the path from the replaced leaf to the root. As in the proof of Theorem 3.1, exponents that are infima are handled by working with $q' > q$ and $\sigma' > \sigma$ and passing to the limit, which is legitimate because every rule is continuous in its arguments. □

Remark B.5 (The bound depends on the decomposition)**.** *The inequality in Proposition B.4(iii) can be strict, for three distinct reasons.*

Cancellation under fan-out. *For complementary coordinate projections P_1, P_2 on $\mathbb{R}^d$, the block $F(x) = (P_1 x) \odot (P_2 x)$ vanishes identically, so $q(F) = 0$, while Theorem 3.1(iii) returns $1 + 1 = 2$.*

The rule bounds $\|B_1(x) \odot B_2(x)\|$ by $\|B_1(x)\|\,\|B_2(x)\|$ and so treats the factors as independent, although both are functions of the same x.

Loss of coefficient information. *The operation $\otimes$ has no unit: for an affine B with $(q, \sigma) = (1, 0)$ and the constant $C \equiv \mathbf{1}$ with $(q, \sigma) = (0, 0)$ we have $B \odot C = B$, yet Theorem B.2(iii) returns $\sigma \le \max(0 + 0, 1 + 0) = 1$. The underlying bound $J \le J_B\|C\| + \|B\|\,J_C$ is exact, with $J_C \equiv 0$; what the exponent does not record is that $\sigma(C) = 0$ holds with $\ell_1 = \ell_0 = 0$. Inserting a factor that is numerically close to a constant, such as a saturated gate, therefore inflates the certified σ by q. The attention values in Table 2 come from Lemma A.7 rather than from rule (iii) and are not affected.*

Atoms outside $\mathcal{P}$. *Self-attention reaches $q = 1$ because softmax is taken as a primitive with $q = 0$, which annihilates the exponent 2 of the logits under rule (i). Written out as $\exp(z)/\sum_j \exp(z_j)$, the numerator lies outside $\mathcal{P}$ and the quotient is not among the operations of Theorem 3.1, so the finer decomposition returns no bound at all.*

Certification therefore proceeds by coarsening: one collapses subexpressions whose exponents are known exactly until the rules suffice, which by Proposition B.4 never weakens the result. The composite entries of Table 2, self-attention, multi-head attention, and the selective state-space block, are atoms of this kind; their tabulated exponents are themselves upper bounds obtained by direct calculation, so collapsing to them is safe but not necessarily sharp.

B.4 Primitive catalogue

Table 2 in Section 3 lists the exponents of common primitives, obtained from Theorems 3.1 and B.2 or by direct calculation; Table 5 gives the coefficients that accompany them.

Table 5: Coefficients of the primitives of Table 2: c is the growth coefficient of Assumption 2.1 (M: saturation level; biases omitted) and ℓ_1 the coefficient of R^σ in Assumption 2.4, with $\kappa := \|W_Q^\top W_K\|_{\mathrm{op}}$ and $\bar{w}_\bullet^{(h)} := \|W_\bullet^{(h)}\|_{\mathrm{op}}$. At $\sigma = 0$ we have $\ell_1 = 0$ and ℓ_0 is the Lipschitz constant of the primitive ($\|W\|_{\mathrm{op}}$ for an affine map, of order $\|\gamma\|_\infty/\sqrt{\epsilon}$ for LayerNorm).

Primitive	Coefficient c	ℓ_1
Saturating activation	$M\sqrt{d}$	0
DyT $\gamma\tanh(\alpha x) + \beta$	$\|\gamma\|\sqrt{d} + \|\beta\|$	0
BHyT	data-dependent bound	–
Softmax, row-stochastic	1	0
LayerNorm ($\epsilon > 0$)	$\|\gamma\|_\infty\sqrt{d} + \|\beta\|$	0
RMSNorm $\gamma \odot x/\mathrm{RMS}(x)$	$\|\gamma\|_\infty\sqrt{d}$	0
ScaleNorm $g\,x/\|x\|$	$\|g\|$	0
Affine map $x \mapsto Wx + b$	$\|W\|_{\mathrm{op}}$	0
Lipschitz activation	$\mathrm{Lip}(\phi)$	0
MLP with Lipschitz ϕ_ℓ	$\prod_\ell \mathrm{Lip}(\phi_\ell)\|W_\ell\|_{\mathrm{op}}$	0
Query-key logits	$\kappa/\sqrt{d_k}$	$2\kappa/\sqrt{d_k}$
Self-attention	$\|W_V\|_{\mathrm{op}}$	$\|W_V\|_{\mathrm{op}}\kappa/\sqrt{d_k}$
Multi-head attention	$\sum_h \bar{w}_O^{(h)}\bar{w}_V^{(h)}$	$\sum_h \bar{w}_O^{(h)}\bar{w}_V^{(h)}\kappa_h/\sqrt{d_k}$
Entrywise power $x \mapsto x^p$, $p \ge 2$	1	p
Selective SSM, Mamba without normalization	parameter-dependent	

The $q = 0$ entries share an exponent but not a geometry: the normalization family (LayerNorm, RMSNorm, ScaleNorm) divides the input by a statistic of itself and maps it onto a fixed sphere or ellipsoid, whereas saturating activations map into a bounded box. Theorem 3.1 treats them as interchangeable; distinguishing them requires structure beyond the exponent. Within the normalization family, any member placed in a block sets its exponent to 0 by rule (v), regardless of where it sits in the block. Super-polynomial maps lie outside $\mathcal{P}$, but composing them with a normalization returns the composite to $\mathcal{P}$ by (v). Discontinuous primitives such as vector quantization are excluded by Definition B.1; during training they are typically replaced by a continuous surrogate, which is the map the velocity actually uses.

B.5 PER-BLOCK CERTIFICATION

A candidate block is certified in six steps, the first three for the forward exponent and the next two for the Jacobian growth exponent, in the order used in Section 3: screen for $q > 1$ first, then for $\sigma > 0$ among the blocks that survive.

1. Decompose the block into primitives and read the pair (q, σ) of each from Table 2.
2. Propagate the input-magnitude exponents with Theorem 3.1 to obtain the block exponent q.
3. If $q > 1$, take the primitive with the largest exponent and either wrap it with a normalization (rule (v), $q \to 0$) or replace it with a Lipschitz substitute ($q \to 1$); repeat step 2. This is where the linear-growth Mamba block comes from (Section 4.1).
4. Propagate the Jacobian growth exponents with Theorem B.2 to obtain the block exponent σ. A block at $q \leq 1$ can still have $\sigma > 0$: un-normalized self-attention is the case that matters, at $(q, \sigma) = (1, 2)$.
5. If $\sigma > 0$ and the backward budget (7) is not met at the target depth, bound the primitive responsible. Normalizing the input of the branch sets $\sigma = 0$ by Theorem B.2(v); for attention the same effect is obtained without touching the residual stream by normalizing the queries and keys, which bounds the logits and removes the R^2 term of Lemma A.7.
6. Once $q \leq 1$ and σ is acceptable, the block satisfies $\|v(x)\| \leq c\|x\|^q + b$ and $L(R) \leq \ell_1 R^\sigma + \ell_0$ with constants built from the primitive coefficients (App. B.4), and Theorems 2.2 and 2.3 apply.

Steps 2 and 4 are independent of each other, but the order matters in practice: at $q > 1$ the reachable radius is doubly exponential, so no achievable σ rescues the backward bound, and repairing q first is what makes step 5 worth doing. The procedure is conservative: rules (i)–(iii) are upper bounds, so a block whose computed exponent exceeds 1 may still admit a tighter direct bound. Certification therefore amounts to finding a decomposition for which the rules suffice, and by Proposition B.4 coarsening a decomposition never weakens the bound (App. B.3). App. D.2 applies the procedure to the blocks used in our experiments.

B.6 BETWEEN-BLOCK CERTIFICATION

In a heterogeneous stack (attention, MLP, gated state-space, and normalization blocks) the exponent q_k can vary with depth. The stack is admissible when $\max_k q_k \leq 1$, so that the per-block procedure applies at every depth. Within this constraint, two quantities remain free: the fraction $|S_1|/D$ of blocks in $S_1 = \{k : q_k = 1\}$, which controls forward amplification, and the depth-averaged Lipschitz constant $\bar{L} = D^{-1} \sum_k L_k$, with $\|\nabla_X v_k\|_2 \leq L_k$, which controls backward amplification.

Remark B.6 (Mixed-exponent stacks)**.** *Let $q_k \in \{0, 1\}$ with per-block constants (c_k, b_k). Applying the one-step inequalities of Step 1 in the proof of Theorem 2.2 block by block gives*

$$\|X_D\| \leq \prod_{k \in S_1} (1 + c_k \Delta t)\, \|X_0\| + \sum_{k=0}^{D-1} b_k \Delta t \prod_{j > k,\, j \in S_1} (1 + c_j \Delta t),$$

and the Jacobian $J_{i:D} = \prod_{k=i}^{D-1} (I + \Delta t\, \nabla_X v_k)$ of the map $X_i \mapsto X_D$ satisfies $\|J_{i:D}\|_2 \leq \prod_{k=i}^{D-1} (1 + L_k \Delta t) \leq \exp\big(\Delta t \sum_{k=i}^{D-1} L_k\big)$. With $\Delta t = T/D$, the forward factor is at most $\exp(\bar{c}\, T\, |S_1|/D)$ with $\bar{c} = |S_1|^{-1} \sum_{k \in S_1} c_k$, and the backward factor is at most $\exp(\bar{L}\, T)$; both are independent of D once $|S_1|/D$ and $\bar{L}$ are fixed. These bounds assume the additive update (1)*; Post-LN blocks, $X_{k+1} = \mathrm{LN}(X_k + \Delta t\, f(X_k))$, fall outside this form.*

No primitive in Table 2 has an exponent strictly between 0 and 1. At such an exponent, Theorem 2.2(i) gives a forward bound polynomial in D, sharper than the geometric bound at $q = 1$; whether a useful primitive exists in this range is open.

C EMPIRICAL EXPONENT

For a residual update $X_{k+1} = X_k + \Delta t\, v_k(X_k)$, let $\{X_k^{(i)}\}_{i=1}^B$ be the states at depth k produced by a fixed evaluation batch. For $s \in \mathcal{S} = \{s_1, \ldots, s_9\}$, geometrically spaced in $[1/4, 4]$, we evaluate

the velocity branch alone at the rescaled state and fit

$$\log \left\| v_k\big(sX_k^{(i)}\big) \right\|_F = \hat{q}_k \, \log \left\| sX_k^{(i)} \right\|_F + \beta_k + \varepsilon_{i,s}, \qquad i = 1, \dots, B, \;\; s \in \mathcal{S},$$

by ordinary least squares, per seed, and average over seeds. Evaluating v_k rather than X_{k+1} removes the identity path, whose contribution would otherwise dominate the slope; the controlled rescaling replaces the narrow spread of $\|X_k\|_F$ within a batch. Since s is bounded, $\hat{q}_k$ measures the local growth rate on $[\frac{1}{4}\|X_k\|_F, 4\|X_k\|_F]$ rather than the infimum in Definition B.1. For a block with $\|v_k(x)\| \le c\|x\|^q + b$ and $b > 0$, the local slope can lie below q at small scales, and multiplicative effects inside the block can push it above q before the asymptotic regime is reached. In GPT-2 without LN, for example, $\hat{q}_k$ lies in $[1.07, 1.59]$ on attention sub-blocks and in $[0.75, 1.07]$ on MLP sub-blocks ($D = 12$, end of training, mean over 5 seeds); rescaling X_k also sharpens the attention weights, a plausible pre-asymptotic cause.

A backward counterpart, and why it reads low. The same protocol estimates the Jacobian growth exponent. On the same rescaling grid $\mathcal{S}$, fit

$$\log \left\| \nabla_X v_k\big(sX_k^{(i)}\big) \right\|_2 = \hat{\sigma}_k \, \log \left\| sX_k^{(i)} \right\|_F + \beta'_k + \varepsilon'_{i,s}$$

by ordinary least squares. The spectral norm needs no explicit Jacobian: a few steps of power iteration on $\delta \mapsto \nabla_X v_k[\delta]$ using forward-mode products, with the transpose supplied by reverse mode, give it to the accuracy the fit requires, at the cost of a small multiple of one backward pass per scale.

The asymmetry with $\hat{q}_k$ is worth stating, because it limits what the estimate can be used for. Both $\hat{q}_k$ and $\hat{\sigma}_k$ probe the states a trained model actually visits, rescaled, whereas q and σ are defined by a supremum over Θ and over the ball of radius $\mathcal{R}_D$. For q the gap is usually mild, since the growth bound is attained along generic rays for the blocks of Table 2. For σ it need not be: as computed in App. A.5, the quadratic term of un-normalized attention is attained only on the scale-invariant cone where a query has two tied maximal logits, and the angular width of that set shrinks like R^{-2}. A trajectory-based probe will essentially never land on it, so $\hat{\sigma}_k$ measured on visited states is a lower bound on σ and can read close to 0 for a block whose certified σ is 2. We therefore use $\hat{\sigma}_k$ as a diagnostic of what a run is doing, not as a substitute for the certificate, which is the reason the analysis of Section 4.3 reports measured coefficients and gradients rather than a fitted $\hat{\sigma}_k$.

D Details of experiments

D.1 Task configurations

D.2 Architectural exponent derivations

Figure 5 summarizes the exponent accounting for the forecasting blocks. All statements use the rules of Theorem 3.1:

- (i) $q(B_1 \circ B_2) \le q_1 q_2$;
- (ii) $q(B_1 + B_2) \le \max(q_1, q_2)$;
- (iii) $q(B_1 \odot B_2) \le q_1 + q_2$;
- (iv) $q(I + \Delta t\, B) \le \max(1, q(B))$;
- (v) $q(\mathrm{LN} \circ B) = 0$.

The primitive exponents are those of Table 2: affine maps and Lipschitz activations (ReLU, GELU, SiLU) have $q = 1$; softmax, sigmoid, and the normalization family have $q = 0$. All bounds are upper bounds on the exponent.

Transformer block (PatchTST). The attention branch is

$$v^{\mathrm{attn}}(X) = W_O \, \mathrm{softmax}\big(QK^\top/\sqrt{d_k}\big)\, V,$$

with Q, K, V linear in X ($q = 1$). The logits $QK^\top$ are bilinear ($q = 2$ by (iii)), but softmax is row-stochastic, so $\mathrm{softmax}(QK^\top)$ has $q = 0 \cdot 2 = 0$ by (i). The product with V gives $0 + 1 = 1$ by (iii),

Table 6: Task configurations. D: number of residual updates; transformer-type backbones count the attention and MLP sub-blocks of a layer separately, so GPT-2 and Toto 2.0, each instantiated with 12 layers, have $D = 24$. The derivation of the exponent q for each variant is given in App. D.2. Parameter counts are real-valued degrees of freedom; the FNO's complex spectral weights count as two each.

Task (data; where it is discussed)	Backbone (params)	Depth	Variants (exponent q)
Forecasting (ETTm1, Weather) §4.1, App. D.3	Mamba (0.58–2.27M)	$D \in \{3, 8\}$	RMSNorm, Pre-LN, Peri-LN (0); linear growth (1); no-LN (5)
	PatchTST (0.62–2.24M)	$D \in \{3, 8\}$	RMSNorm, Pre-LN, Peri-LN (0); no-LN (1)
Forecasting (GIFT-Eval) §4.4, App. D.6	Toto 2.0 (10.6M)	$D = 24$	pre-RMSNorm (0); no-LN (2)
Operator learning (3-D battery cooling) §4.2, App. D.4	FNO (6.46M)	$D = 4$	in-branch LN (0); no-LN (1)
Language modeling (OpenWebText) §4.3, App. D.5	GPT-2 (124M)	$D = 24$	Pre-LN (0); no-LN (1)

and W_O keeps $q = 1$. The MLP branch $W_2\,\mathrm{GELU}(W_1 X)$ is a composition of $q = 1$ primitives, hence $q = 1$. Without normalization both velocities are at $q = 1$; with a normalization in front of each branch (Pre-LN) or around it, rule (i) or (v) gives $q = 0$. Removing normalization from a transformer with ungated MLP therefore leaves it at the threshold, not above it. Attention is a multi-token primitive, so the norm in which its exponent is stated matters. With Q, K, V linear in X and a row-stochastic softmax, each output column satisfies $\|v^{\mathrm{attn}}(X)_{:,j}\|_2 \le \|W_O\|_{\mathrm{op}}\|W_V\|_{\mathrm{op}}\|X\|_{\infty,2}$, where $\|X\|_{\infty,2} := \max_k \|X_{:,k}\|_2 \le \|X\|_F$. Its $q = 1$ therefore holds in this worst-token sense, and, with a possibly larger coefficient, in the Frobenius norm used throughout.

Mamba block. With input u (the residual state, or its normalization), the block computes $x, z = W_{\mathrm{in}}u$ ($q = 1$), $x \leftarrow \mathrm{SiLU}(\mathrm{conv}(x))$ ($q = 1$), and the selective parameters $\Delta = \mathrm{softplus}(W_\Delta x)$, $B = W_B x$, $C = W_C x$ (each $q = 1$). For a Hurwitz state matrix A and $\Delta \ge 0$, the transition $\bar{A} = \exp(\Delta A)$ satisfies $\|\bar{A}\| \le 1$ ($q = 0$); without this assumption $\bar{A}$ is not polynomially bounded. The input matrix $\bar{B} = \Delta \odot B$ has $q = 1 + 1 = 2$ by (iii). The recurrence $h_t = \bar{A}_t h_{t-1} + \bar{B}_t \odot x_t$ adds $q(\bar{B}) + q(x) = 3$ at every step and $\bar{A}_t$ does not increase the exponent, so by induction over the finite sequence h has $q = 3$. The readout $y = C \odot h$ has $q = 1 + 3 = 4$, the gated output $y \odot \mathrm{SiLU}(z)$ has $q = 4 + 1 = 5$, and the output projection and the channel-mixing MLP keep it at 5 by (i). The no-LN block is therefore at $q = 5$; a normalization in front of the block gives $q = 0$.

Linear-growth Mamba block. Two primitives are changed (Figure 5c). The parameter branch receives $\hat{x} = x/\|x\|$ ($q = 0$), so Δ, B, C are affine functions of a bounded input and have $q = 1 \cdot 0 = 0$ by (i); consequently $\bar{B}$ has $q = 0$, the recurrence h has $q = 0 + 1 = 1$, and the readout $y = C \odot h$ has $q = 0 + 1 = 1$. The output gate uses the sigmoid ($q = 0$), so $y \odot \sigma(z)$ has $q = 1 + 0 = 1$. The residual stream is unchanged, and the block is at $q = 1$ without a normalization layer.

Toto 2.0. *Layer structure.* Every layer contains two residual sub-blocks *in sequence*, attention and then the gated MLP, each with its own parameter-free RMSNorm in front and its own skip connection; attention and the gated MLP are not separate layers and do not alternate. The depth in the sense of (1) is therefore twice the number of layers, and the 12-layer instantiation we train has $D = 2 \times 12 = 24$ (the released 2.5B checkpoint has 48 layers and $D = 96$). One layer per group attends along variates and the rest along time, which changes the mask and not any exponent. These statements are read from the released configuration and reference implementation.

Gated MLP branch. The feed-forward network is SwiGLU, $W_2\big((W_g X) \odot \mathrm{SiLU}(W_u X)\big)$: one linear gate multiplied entrywise by a Lipschitz activation of a second linear map, with the two halves produced by a single fused projection. Both factors have $q = 1$, so the Hadamard product has $q = 2$

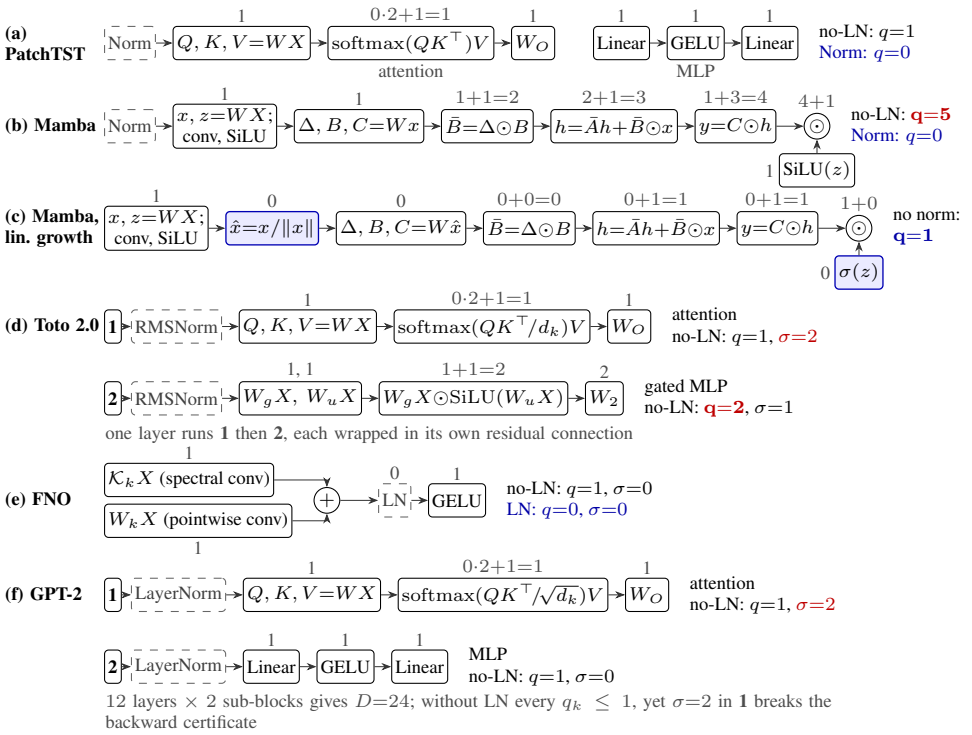


Figure 5: Velocity fields of the blocks used in the experiments, with the exponent of each intermediate quantity (numbers above the boxes), computed with Theorem 3.1: composition multiplies exponents, parallel sums take the maximum, Hadamard products ($\odot$) add them, and a normalization layer (dashed) sets the block to $q = 0$. $\bar{A} = \exp(\Delta A)$ is a contraction ($q = 0$) for Hurwitz A; σ is the sigmoid ($q = 0$) and ϕ a Lipschitz activation ($q = 1$). (a) PatchTST. (b) Mamba. (c) The linear-growth Mamba block; the two modified primitives are shaded. (d) One Toto 2.0 layer, which is two residual sub-blocks in sequence rather than one block: attention, then the gated MLP. (e) One residual Fourier layer of the FNO (Section 4.2); both convolutions are linear, so the layer is the one case here with $q = 1$ and $\sigma = 0$ without normalization. (f) One GPT-2 layer as implemented in nanoGPT (Karpathy, 2022) (Section 4.3), again two residual sub-blocks: attention, then a plain MLP. Unlike the gated MLP of (d) the plain MLP stays at $q = 1$, so the no-LN model satisfies $q \leq 1$ at every sub-block and fails only through σ. Pairs (q, σ) are given where the backward exponent is the operative one.

by (iii) and the output projection keeps it there by (i). For the Jacobian, rule (iii) of Theorem B.2 gives $\sigma \leq \max(0 + 1, 1 + 0) = 1$, and (i) leaves it at 1. The branch is therefore at $(q, \sigma) = (2, 1)$.

Attention branch. As in PatchTST, $(q, \sigma) = (1, 2)$. The released configuration disables query-key normalization, so the logits are the unmodified bilinear form and Lemma A.7 applies verbatim, with $\sqrt{d_k}$ replaced by d_k because the implementation uses the μP logit scaling $1/d_k$. The per-dimension query scale and the rotary or xPos projection are linear maps and fold into κ without changing an exponent.

Without normalization. Removing the two RMSNorms leaves

$$(q, \sigma)_{\text{attn}} = (1, 2), \qquad (q, \sigma)_{\text{MLP}} = (2, 1),$$

so over the 24 updates $\max_k q_k = 2$ and $\max_k \sigma_k = 2$. The forward certificate fails through the gate and the backward certificate fails through the attention, the only architecture we examine in which both fail at once. Normalizing the input of the gate returns the MLP branch to $q = 1$ without a normalization layer, as in the linear-growth Mamba block, and enabling query-key normalization bounds the logits and returns the attention branch to $\sigma = 0$. Any SwiGLU or GeGLU transformer has the same structure.

The residual form is not additive. The released model combines each sub-block with its skip by a τ-rule rather than by addition. In the forward pass the split is the identity and the recombination is a weighted sum, so

$$X_{k+1} = \beta_k X_k + \alpha_k F_k(N(X_k)), \qquad \beta_k = (1 + \tau_k^2)^{-1/2}, \quad \alpha_k = \tau_k \beta_k, \quad \alpha_k^2 + \beta_k^2 = 1,$$

with τ_k decreasing with depth and separate values for the two sub-blocks ($\tau^{\mathrm{attn}} = 0.150$ and $\tau^{\mathrm{mlp}} = 0.0289$ at the first layer, in the ratio 5.14 fixed by the configuration). This is (1) with $\Delta t = 1$ and $v_k(x) = (\beta_k - 1)x + \alpha_k F_k(N(x))$, so the damping term puts the normalized block at $q = 1$ with $c_k = 1 - \beta_k$ (1.1×10^{-2} and 4.2×10^{-4} at the first layer), not at $q = 0$. Since $\beta_k < 1$, the exact recursion $\|X_{k+1}\| \le \beta_k \|X_k\| + \alpha_k b_k$ is a contraction and $\mathcal{R}_D$ is bounded uniformly in D, whereas Assumption 2.1 absorbs the sign of the damping and reports growth $(1 + c_k)$; the certificate is sound but conservative for this block.

FNO for the battery-cooling operator. The FNO consists of a lifting network (convolution, LN, GELU, convolution), four residual Fourier layers, and a projection network (convolution, LN, GELU, convolution, inverse target normalization), with hidden width 64 and 4 Fourier modes per dimension (Figure 6). Each residual Fourier layer has the velocity $v_k = \mathrm{GELU} \circ \mathrm{LN} \circ (\mathcal{K}_k + W_k)$, so the normalization is LayerNorm placed *inside* the branch, between the linear part and the activation, rather than on the branch input (Pre-LN) or after the residual sum (Post-LN); we keep the example's scheme unchanged. The spectral and pointwise convolutions are linear ($q = 1$), so their sum is at $q = 1$ by rule (ii); composition with LN ($q = 0$) and GELU ($q = 1$) gives $q(v_k) = 1 \cdot 0 \cdot 1 = 0$ by rule (i). The no-LN variant removes only the LN inside each residual Fourier layer, which leaves $q(v_k) = 1 \cdot 1 = 1$. The Jacobian exponent is $\sigma(v_k) = 0$ in both variants: the two convolutions are linear and GELU has bounded derivative, so all three primitives have $\sigma = 0$, and rules (i) and (ii) of Theorem B.2 give $\sigma \le 0 \cdot 1 + 0 = 0$. The FNO is therefore the one architecture here whose un-normalized variant is subcritical on both sides, $(q, \sigma) = (1, 0)$, which is consistent with its residual stream never overflowing; what the exponent changes is extrapolation, not finiteness (Section 4.2). The LNs in the lifting and projection networks are kept in both variants: these networks are applied once, outside the residual stream, so they do not enter the exponent of any v_k.

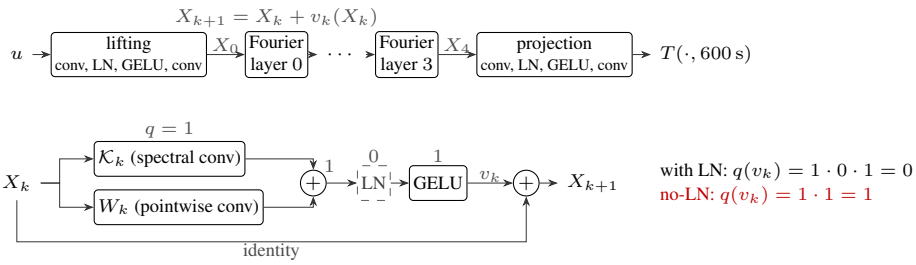


Figure 6: FNO for the battery-cooling operator (top) and one residual Fourier layer (bottom), with the exponent of each primitive. The dashed LN is present in the LN variant ($q = 0$) and removed in the no-LN variant ($q = 1$); the LNs in the lifting and projection networks are kept in both variants. Figure 5e gives the same layer in the format used for the other architectures.

These derivations use the upper bounds of Theorem 3.1, so they certify $q = 5$ for the un-normalized Mamba block and $q = 2$ for the gated MLP without asserting that either block grows that fast.

D.3 Time-series forecasting

Data. ETTm1 (Zhou et al., 2021): 7 channels at 15-minute sampling, fixed $12/4/4$-month borders ($34{,}560/11{,}520/11{,}520$ time steps). Weather (Wu et al., 2021): 21 channels at 10-minute sampling, $0.7/0.1/0.2$ split of $52{,}696$ rows ($36{,}887/5{,}270/10{,}539$). Input length 96 (12 patches of length 16, stride 8); horizons $H \in \{96, 192, 336, 720\}$; sliding windows with stride 1 give about 34.4K (ETTm1) and 36.7K (Weather) training windows per horizon. Both backbones are trained channel-independently: each channel is embedded and forecast on its own, with no cross-channel mixing and no channel mixup.

Mamba. CMamba (Zeng et al., 2024) built on the selective SSM of Gu & Dao (2024): width 128, $d_{\mathrm{ff}} = 128$, state size 16, convolution width 4, expansion 2, Δ-rank 32, parallel scan, dropout 0. We replace CMamba's gated global data-dependent MLP with the standard MLP of PatchTST (width 256), so that the two backbones differ only in their token-mixing block, and we drop its channel mixing in favor of channel independence; the selective state-space block, which carries the $q = 5$ exponent, is unchanged. Parameters: 0.583M ($D = 3$, $H = 96$) to 2.265M ($D = 8$, $H = 720$),

about 0.145M per residual update; the flatten head has 0.148M ($H = 96$) to 1.106M ($H = 720$) parameters. Adam, learning rate 10^{-3}, batch 64, MSE loss, learning-rate schedule `type3`, at most 100 epochs with early stopping (patience 3). Variants: RMSNorm baseline (Zhang & Sennrich, 2019), Pre-LN (Xiong et al., 2020), and Peri-LN (Kim et al., 2025) ($q = 0$); linear growth ($q = 1$); no-LN, i.e., normalization removed ($q = 5$). The no-LN variant is labeled *free velocity* in our code and result files.

PatchTST. PatchTST (Nie et al., 2023) with 16 heads, width 128, $d_{\text{ff}} = 256$, end padding, dropout 0.2, RevIN, and MLP width 256. Parameters: 0.615M ($D = 3$, $H = 96$) to 2.236M ($D = 8$, $H = 720$), about 0.132M per residual update; channel independence makes the counts identical on both datasets. Adam, learning rate 10^{-4}, batch 128, MSE loss; one-cycle schedule (warm-up fraction 0.4) on ETTm1 and step decay on Weather; at most 100 epochs with early stopping (patience 3). Variants: RMSNorm baseline, Pre-LN, Peri-LN ($q = 0$), and no-LN ($q = 1$).

Protocol. Seeds 2021–2026 on ETTm1 and 2021–2023 on Weather, giving 4 horizons $\times$ 2 depths $\times$ 9 seed–dataset pairs $=$ 72 runs per variant. Blow-up detection checks the training loss for NaN at every step and the final test MSE against the threshold in Section 4. Accuracy is averaged over surviving seeds. Significance in Section 4.1 uses Welch's two-sample t-test on the per-cell seed means and standard deviations, comparing the $q = 1$ block with the most accurate $q = 0$ variant, without multiple-comparison correction.

Table 7: Blow-up counts. ETTm1: 6 seeds per cell; Weather: 3 seeds per cell.

				ETTm1					Weather				
Backbone	Variant	q	D	96	192	336	720	$\Sigma/24$	96	192	336	720	$\Sigma/12$
Mamba	no-LN	5	3	2	3	4	3	12	0	1	0	1	2
			8	6	6	6	6	24	1	3	3	3	10
	RMSNorm	0	3, 8	0	0	0	0	0	0	0	0	0	0
	Pre-LN	0	3, 8	0	0	0	0	0	0	0	0	0	0
	Peri-LN	0	3, 8	0	0	0	0	0	0	0	0	0	0
	linear growth	1	3, 8	0	0	0	0	0	0	0	0	0	0
PatchTST	no-LN	1	3, 8	0	0	0	0	0	0	0	0	0	0
	RMSNorm	0	3, 8	0	0	0	0	0	0	0	0	0	0
	Pre-LN	0	3, 8	0	0	0	0	0	0	0	0	0	0
	Peri-LN	0	3, 8	0	0	0	0	0	0	0	0	0	0

D.4 Operator learning

Problem. We follow the MathWorks battery-module example (The MathWorks, Inc., n.d.). The heat equation of Section 4.2 is solved by finite elements on $\Omega = \Omega_{\text{cell}} \cup \Omega_{\text{tab}} \cup \Omega_{\text{conn}}$ (Table 8) up to $t = 600$ s, with convective cooling on the front and back faces Γ_{conv}. The operator maps the six-channel input $u(x) = (x_1, x_2, x_3,\ T_{\text{amb}}\mathbf{1}_\Omega(x),\ h\,\mathbf{1}_{\Gamma_{\text{conv}}}(x),\ Q\,\mathbf{1}_{\Omega_{\text{cell}}}(x))$ to the terminal field $v(x) = T(x, 600)$, both sampled on a 32^3 grid; the coordinate channels let the network localize the geometry, and each parameter is extended by zero outside the region where it acts.

Table 8: Material properties of the three regions (The MathWorks, Inc., n.d.).

Region	K [W/(m K)]	ρ [kg/m^3]	c_p [J/(kg K)]
Ω_{cell}	$\text{diag}(2, 80, 80)$	780	785
Ω_{tab}	$386\,I$	2700	890
Ω_{conn}	$400\,I$	540	840

Parameter sensitivity. Table 9 lists the target field when one parameter of the original box is moved 50% of its range width outside the box and the others are at mid-range. The sensitivities quoted in Section 4.2 are finite differences of the spatial mean between the two shifts of each parameter. T_{amb} translates the field without changing its spread, Q rescales its excess over ambient and moves it outside the range of target values seen in training, and h mainly changes its spread.

Table 9: Target temperature field $T(\cdot, 600\,\mathrm{s})$ under one-parameter shifts of the original box.

Shifted parameter	spatial mean [K]	1st–99th pct. [K]	min–max [K]
mid-range	350.5	[341.4, 353.3]	[341.4, 353.3]
$T_{\mathrm{amb}} = 270\,\mathrm{K}$	330.5	[321.4, 333.3]	[321.4, 333.3]
$T_{\mathrm{amb}} = 310\,\mathrm{K}$	370.5	[361.4, 373.3]	[361.4, 373.3]
$h = 5$	352.3	[348.9, 353.4]	[348.9, 353.4]
$h = 25$	349.1	[335.5, 353.3]	[335.4, 353.3]
$Q = 5\,\mathrm{W}$	310.2	[307.1, 311.1]	[307.1, 311.1]
$Q = 25\,\mathrm{W}$	390.9	[375.6, 395.5]	[375.6, 395.6]

Training. Adam with a constant learning rate of 10^{-3}, minibatch size 8 with the training set reshuffled every epoch, an $80/20$ train/validation split, and 300 epochs (6,600 iterations in the original box, 84,600 in the larger box); 5 seeds per variant.

Evaluation sets. (i) The validation split: parameter sets held out from training, on the training grid. (ii) In-box: the $3^3 = 27$ combinations of three interior levels per parameter, $\{282, 290, 298\}$ K and $\{11, 15, 19\}$ for h and Q in the original box, and $\{274, 290, 306\}$ K and $\{13, 25, 37\}$ in the larger box. The h and Q levels lie between training grid values in both boxes, so no in-box point is a training point. (iii) OOD shells: for each parameter with training range $[\ell, u]$ and width $w = u - \ell$, a test value at distance δ lies δw below the lower bound or above the upper bound, and each parameter takes one of the three levels

$$\ell - \delta w, \qquad (\ell + u)/2, \qquad u + \delta w, \qquad \delta \in \{10\%, 25\%, 50\%\}.$$

For each δ, all $3^3 = 27$ combinations are simulated except the all-centre point, which lies inside the box; each of the remaining 26 points is outside the training range in at least one parameter (Table 10). In the original box the centre values lie between training grid values, so no test point reuses a training value on any axis. In the larger box the T_{amb} centre (290 K) is a training grid value, but the h and Q centres are not, so no test point coincides with a training point. Because h and Q are floored at 1 in the larger box, their lowest level is 1, i.e., 30% rather than 50% of the width below the range. (iv) One parameter shifted: the six points of the $\delta = 50\%$ shell in which exactly one parameter is off-centre.

Table 10: OOD test levels per parameter: $\ell - \delta w$, the centre, and $u + \delta w$ for $\delta \in \{10, 25, 50\}\%$. *: floored at 1.

			below				above		
Box	Parameter	$[\ell, u]$	-50%	-25%	-10%	centre	$+10\%$	$+25\%$	$+50\%$
original	T_{amb} [K]	$[280, 300]$	270	275	278	290	302	305	310
	h [W/(m^{2}K)]	$[10, 20]$	5	7.5	9	15	21	22.5	25
	Q [W]	$[10, 20]$	5	7.5	9	15	21	22.5	25
larger	T_{amb} [K]	$[270, 310]$	250	260	266	290	314	320	330
	h [W/(m^{2}K)]	$[10, 40]$	1^*	2.5	7	25	43	47.5	55
	Q [W]	$[10, 40]$	1^*	2.5	7	25	43	47.5	55

Table 11: Training sets for the FNO. Iterations correspond to 300 epochs.

	original box	larger box
T_{amb} [K]	280–300	270–310
h [W/(m^{2}K)]	10–20	10–40
Q [W per cell]	10–20	10–40
values per axis	6 / 6 / 6	11 / 16 / 16
spacing	4 / 2 / 2	4 / 2 / 2
N	216	2816
training iterations	6,600	84,600

Results. Table 12 lists the values plotted in Figure 2. All values are computed from the logged evaluations of the final checkpoints (6,600 and 84,600 iterations). For the validation, in-box, and OOD-shell sets, each seed's RMSE is pooled over all parameter points of the set; each one-parameter entry is a single parameter point.

Table 12: FNO test RMSE [K], mean (± sd) over 5 seeds.

		distance from the training data					one parameter shifted 50% beyond its range					
Box	Variant (q)	val	in box	+10%	+25%	+50%	T^-	T^+	h^-	h^+	Q^-	Q^+
original	LN (0)	$0.38_{\pm 0.09}$	$1.71_{\pm 0.10}$	$2.72_{\pm 0.19}$	$6.92_{\pm 0.37}$	$15.29_{\pm 0.57}$	$5.70_{\pm 1.42}$	$9.27_{\pm 1.99}$	$1.10_{\pm 0.25}$	$0.53_{\pm 0.15}$	$14.50_{\pm 0.58}$	$14.45_{\pm 1.20}$
	no-LN (1)	$0.30_{\pm 0.11}$	$2.51_{\pm 0.17}$	$2.19_{\pm 0.14}$	$2.91_{\pm 0.32}$	$5.26_{\pm 0.95}$	$4.11_{\pm 0.27}$	$4.35_{\pm 0.64}$	$3.17_{\pm 0.32}$	$3.26_{\pm 0.76}$	$6.26_{\pm 3.08}$	$5.14_{\pm 1.19}$
larger	LN (0)	$0.31_{\pm 0.24}$	$2.73_{\pm 0.15}$	$3.51_{\pm 0.37}$	$10.89_{\pm 0.96}$	$24.06_{\pm 1.05}$	$8.53_{\pm 0.12}$	$8.43_{\pm 0.21}$	$0.65_{\pm 0.11}$	$0.50_{\pm 0.10}$	$9.56_{\pm 0.85}$	$24.00_{\pm 2.48}$
	no-LN (1)	$0.23_{\pm 0.13}$	$2.96_{\pm 0.15}$	$2.55_{\pm 0.16}$	$3.15_{\pm 0.18}$	$4.34_{\pm 0.19}$	$8.58_{\pm 0.07}$	$8.64_{\pm 0.14}$	$1.43_{\pm 0.33}$	$0.88_{\pm 0.17}$	$1.53_{\pm 0.21}$	$2.44_{\pm 1.16}$

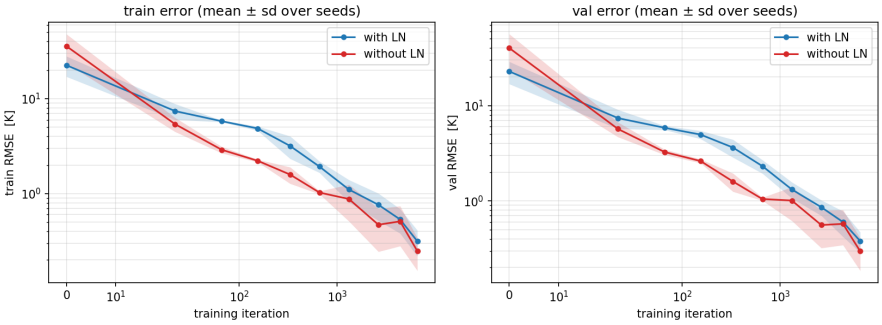


Figure 7: FNO training and validation RMSE over training (mean ± sd over 5 seeds), original box.

D.5 Language modeling: GPT-2 on OpenWebText

Setup. nanoGPT GPT-2 124M (12 layers, $D = 24$, 12 heads, width 768, $d_k = 64$, context 1,024, biases, dropout 0) on OpenWebText (Gokaslan & Cohen, 2019), with Pre-LN ($q = 0$) or with `ln_1`, `ln_2`, and `ln_f` replaced by identities (no-LN, $q = 1$). Batch $12 \times 1{,}024$ tokens with 40 gradient-accumulation steps (491,520 tokens per iteration); AdamW with learning rate 6×10^{-4}, 750 warm-up iterations, cosine decay to 6×10^{-5} at iteration 20,000, $\beta = (0.9, 0.95)$, weight decay 0.1, gradient clipping at 1.0; bf16 autocast, whose largest finite value 3.39×10^{38} equals that of FP32. Seeds 1337–1340 per variant. All four Pre-LN runs are finite through iteration 20,000. The no-LN runs first give a non-finite validation loss at iterations 1,250 (seed 1337), 3,000 (1340), 4,750 (1338), and 7,500 (1339); validation is evaluated every 250 iterations, and only seed 1339 has a finite checkpoint at 6,750.

Measurements at iteration 6,750. Table 13 lists, per block, the query-key coefficient, the token norm at the attention input (per-element RMS times $\sqrt{768}$, median over 16 fixed validation windows), the largest attention logit, the forward coefficients c_k, and the layerwise factor $L_k(R_k) = c_k^{\mathrm{attn}} + c_k^{\mathrm{attn}} \kappa_k R_k^2 / \sqrt{d_k}$ used in Section 4.3. The factor is a surrogate: it uses the largest head and the full-layer $\|W_O\|_2 \|W_V\|_2$ in place of the per-head sum of Lemma A.7, and the typical rather than the largest token norm. The MLP constant uses $\mathrm{Lip}(\mathrm{GELU}) \leq 1.13$.

Replay. From the seed-1339 checkpoint at iteration 6,750, training is resumed with per-iteration logging of the pre-clip gradient norm, the per-block attention logits, and the query-key coefficients, and with the gradient norm of block 10's attention recorded per micro-batch. GPU kernels are not deterministic, so the onset differs between replays from the same checkpoint: in the replay of Figure 3 the pre-clip gradient norm exceeds 1 at iteration 6,764 and the first non-finite value appears at 6,781; in a second replay used for the depth profiles, the norm stays near 0.5 until 6,769 and the first non-finite step is 6,776. In both, the weights change by a few percent at most before divergence, and in the replay of Figure 3 the maximum attention logit is attained in block 10 at every iteration but one.

Table 13: GPT-2 124M on OpenWebText at iteration 6,750. κ: $\max_h \|W_Q^{h\top} W_K^h\|_2$ (Pre-LN: mean over 4 seeds). All other columns: no-LN (seed 1339). R: typical token norm at the attention input; logit: largest attention logit; $c^{\mathrm{attn}} = \|W_O\|_2 \|W_V\|_2$; $c^{\mathrm{MLP}} = 1.13\|W_2\|_2\|W_1\|_2$; $L^{\mathrm{attn}}(R)$: layerwise attention factor.

block	κ Pre-LN	κ no-LN	R	logit	c^{attn}	c^{MLP}	$L^{\mathrm{attn}}(R)$
0	3.3	614.5	1.0	8.0	3.9	13.0	285
1	5.1	39.9	2.4	6.0	1.6	6.1	47.7
2	7.3	58.6	4.1	10.9	1.4	3.2	174
3	7.2	36.9	4.1	12.0	2.0	3.0	161
4	8.0	44.4	4.1	13.1	2.7	3.3	256
5	6.2	53.2	4.2	23.1	4.7	3.9	549
6	4.5	46.6	4.3	29.8	9.7	5.3	1.0×10^3
7	4.6	26.0	4.5	24.5	4.3	21.2	287
8	4.2	23.9	11.7	96.0	29.6	73.9	1.2×10^4
9	4.5	19.4	48.0	1.6×10^3	5.6	100.6	3.1×10^4
10	4.4	175.5	50.2	4.0×10^4	7.1	102.0	3.9×10^5
11	4.1	4.2	40.4	229	12.7	93.4	1.1×10^4

D.6 Time-series foundation model: Toto 2.0

Toto 2.0 is included as a configuration in Table 6; its exponents are derived in App. D.2 and its layer structure, gate, and τ-rule residual form are described there. The pre-RMSNorm and no-LN versions were run for a short pilot of 3,300 steps (until blowup of no-LN variant). Through step 2,500, the no-LN variant was more performant, scoring 1.0028 on GIFT-Eval's (Aksu et al., 2024) MASE metric as compared to 1.0391 for the standard pre-RMSNorm variant. Training was performed in accordance with Toto 2.0's official documentation: a hybrid AdamW/NorMuon optimizer, contiguous patch masking (CPM), which the release uses for single-pass parallel decoding, to generate training examples, and a mixture of synthetically generated data from TempoPFN and real GIFT-Eval pretraining data. More details and specific optimization parameters are available in the official Toto 2.0 release.

Model size. The released Toto 2.0 checkpoints are 4M, 22M, 313M, 1B, and 2.5B parameters. The 12-layer, 10.6M instantiation used here is not one of them; it is trained from scratch with the released recipe, whose u-μP parameterization is designed to transfer a single set of hyperparameters across sizes, and it sits between the 4M and 22M released sizes. The exponents of Section 4.4 are properties of the block structure and do not depend on the width or the number of layers.

Benchmark. GIFT-Eval (Aksu et al., 2024) evaluates on 23 datasets, about 144,000 series and 177 million points across seven domains and ten sampling frequencies, grouped into 97 triplets of dataset, frequency, and prediction length; the medium and long horizon settings extend the short one by factors of 10 and 15, giving horizons from 6 to 900 steps. Its pretraining corpus is separate from the evaluation split, with 71 univariate and 17 multivariate datasets over 4.5 million series and 230 billion points, and is constructed so that models pretrained on it can be evaluated on the benchmark without leakage.