RESIDUAL LAYERS AND FLOW CALLS ALONG A COMMON PATH

Anonymous authors
Paper under double-blind review

ABSTRACT

Residual layers and time-conditioned flow calls both build transformations through local updates, but more updates can encode a richer process or merely resolve the same process more finely. We compare them by the full path of intermediate transformations. For compactly supported, time-Lipschitz fields in fixed dimension d and spatial regularity $s \in \{1, 2\}$, we prove a sharp worst-case all-prefix information rate $\widetilde{\Theta}(\varepsilon^{-\max\{d/s,(1+d/s)/2\}})$ on sufficiently resolved Euler grids. When $d \geq s$, the path and endpoint have the same leading information power, while full-field reconstruction costs more. A finite time-conditioned network attains the path rate with its implementation costs counted; its encoded path can also be traversed with $O(\varepsilon^{-1/2})$ aligned calls and linear readout on the original grid. Conditional execution bounds and controlled examples show how call placement affects path error. The distinction separates information stored about a process from the resolution used to execute it.

1 INTRODUCTION

A deeper residual network can spend depth in two ways. Additional blocks can divide one transformation into smaller updates, or they can encode new intermediate changes. A time-conditioned flow can also spend depth on execution: a solver calls the same velocity network at more time points. These choices all add sequential computation, but they need not add the same information about the transformation. To compare residual depth with flow calls, we must distinguish a richer process from finer execution of one process.

Our theorem studies depth-normalized Euler teachers: at depth k of L, the residual increment is $r_k = L^{-1} f(\cdot, k/L)$ for one spatially regular, time-Lipschitz field f. Converting an arbitrary trained residual stack to this form is a separate question.

A residual branch adds an increment to the current state. A flow network instead gives a velocity from the current state and time; a solver converts it into an increment. For an Euler step, the updates read

$$x_{k+1} = x_k + r_{\theta_k}(x_k), \qquad y_{j+1} = y_j + h_j v_\phi(y_j, t_j). \tag{1}$$

Here r_{θ_k} is an increment, while v_ϕ is a rate of change: the step length h_j converts that rate into an increment. A continuous flow's exact increment instead integrates the velocity along its trajectory. This local-update viewpoint underlies the established connection between residual networks and differential equations (He et al., 2016; Chen et al., 2018; E et al., 2022).

Figure 1 makes the distinction visible: the same intermediate transformations can be stored at anchors and traversed with different numbers of updates.

The identity shortcut makes subdivision natural. In our normalized class, $r_k = hf(\cdot, t_k)$ with $h = 1/L$ and $\sum_k \mathrm{Lip}(r_k) = O(1)$; state propagation depends on this cumulative budget rather than layer count alone. Appendix E gives the bound and treats parameter learning separately. Additional blocks may still carry new path information.

Why make this correspondence quantitative? The residual stack may store a different rule at each layer, while a time-conditioned flow reuses one network at different times. Counting blocks and calls alone misses whether the path itself has changed. At a fixed path accuracy, representation

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.

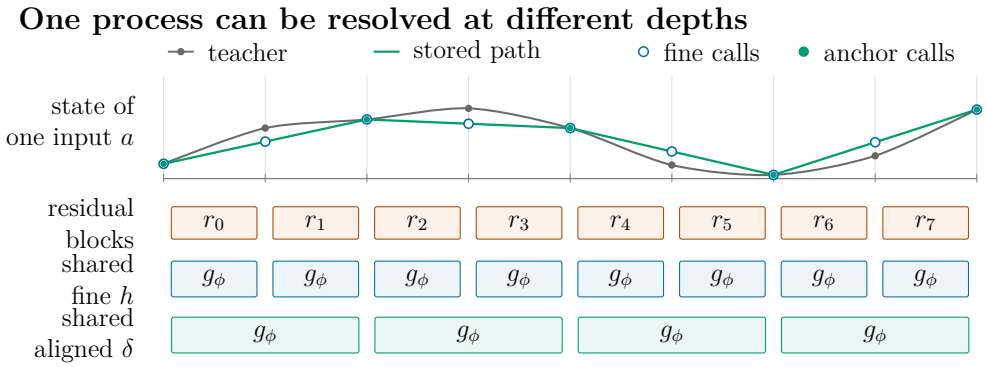


Figure 1: **One path, different execution depths.** Gray dots show teacher prefixes for one input; the green line joins stored anchors. Orange blocks use distinct rules, while blue and green rows call one conditioned evaluator at fine or aligned times. The drawing is schematic; path fidelity is defined over every allowed input and prefix.

capacity determines which intermediate changes can be retained; calls and their placement determine how the stored path is traversed. This also gives a fidelity target for conversion from layer-specific rules to shared execution: preserve the intermediate transformations and control the error of the updates used to follow them.

What should remain the same? To study this correspondence, we fix the transformation process before changing its implementation. Matching only the output would allow two computations to take entirely different intermediate routes. For example, one can move states away from their initial positions and then bring them back, while another leaves them fixed throughout. Their final maps agree, but their computations along depth differ. We therefore use the *transformation path*: the map from an input to its state at each stage, considered over a common domain of inputs. Preserving this path is the task we choose for understanding how computation unfolds with depth; a task that only needs final predictions may instead use endpoint fidelity.

This gives the central question: *when a residual stack and one time-conditioned network must preserve the same transformation process, what must they represent, and what must their execution resolve?* The first part asks how much information the process contains at a specified accuracy. The second asks whether a particular sequence of updates can reproduce it. We keep storage, work per call, and the number of calls separate when connecting the answers.

The answer developed here. We first quantify the information needed to preserve every intermediate transformation. Retaining a whole path imposes more constraints than retaining its endpoint, so an additional leading accuracy power is a reasonable possibility. For compactly supported, time-Lipschitz dynamics in fixed state dimension d, with spatial regularity $s = 1$ (Lipschitz) or $s = 2$ (a Lipschitz Jacobian), we show that when $d \geq s$, the path and endpoint have the same worst-case leading information power, up to logarithms, whereas reconstructing the full local field has a larger power. One finite time-conditioned network attains the path rate on the original Euler grid, including the cost of its clock, precision and inverse computations. The information lower bound allows any finite decoder, including compressed untied representations. The shared evaluator attains its leading accuracy power while receiving only current state and time.

Classical mixed-smoothness entropy already gives the scalar $s = 2$ information upper power after an Euler-prefix embedding (Temlyakov, 2016). Hierarchical stepping, anchor maps, and flow-map composition also provide substantive path tools (Liu et al., 2022; Stelter et al., 2026; Boffi et al., 2025). Here the contribution is the sharp information comparison for the stated teacher class together with a coherent shared evaluator whose online implementation is counted, and the aligned-call consequence of that code.

The construction also gives a direct layer–call correspondence. Its stored path joins anchor maps spaced $\delta \asymp \sqrt{\varepsilon}$ apart. At an anchor, one Euler call with step length δ performs the relative transport

to the next anchor. Thus $M = O(\varepsilon^{-1/2})$ aligned calls, followed by linear interpolation, retain every original residual prefix to accuracy ε once $L \gtrsim \varepsilon^{-1/2}$. A curved trajectory shows that this call power is necessary in the worst case for the specified linear-output rule. This is an execution consequence of the path representation; its storage and work per call still depend on spatial complexity.

Other schedules receive conditional path-error bounds in Section 4; Appendix E covers standard subdivision and gradient qualifications.

Appendix F.1 gives the Euler-prefix embedding used in the scalar entropy comparison.

2 A COMMON PATH FOR COMPARING LAYERS AND CALLS

The updates in Equation (1) give us a common starting point. We now fix what their accumulation must preserve. A transformation path records the state reached from each allowed input at each stage; Figure 1 illustrates how the same path can be represented and traversed at different resolutions. Throughout, L denotes residual layers and M denotes calls to a conditioned network. Our information theorem starts with $M = L$ on a common grid. Corollary 2 then changes the execution and readout rule explicitly, retaining every original prefix as an accuracy constraint.

We compare two executions on the same state space, horizon, initial-state domain, and Euler grid. Let L be dyadic, $h = 1/L$, and $t_k = kh$. A residual teacher has increments $r_k(x) = hf(x, t_k)$ and evolves as

$$X^f_{k+1}(a) = X^f_k(a) + hf(X^f_k(a), t_k), \qquad X^f_0(a) = a. \tag{2}$$

A conditioned evaluator g is executed with the same numerical update,

$$\widehat{X}^g_{k+1}(a) = \widehat{X}^g_k(a) + hg(\widehat{X}^g_k(a), t_k), \qquad \widehat{X}^g_0(a) = a. \tag{3}$$

The evaluator receives only the current state and depth coordinate; the initial state, execution history, and any uncharged persistent memory are unavailable.

Common state–depth object. Both descriptions live on the same product space $\mathbb{R}^d \times [0, 1]$: a residual block is attached to a discrete depth slice t_k, whereas the conditioned evaluator is queried at the pair (x, t_k). What we compare is the induced path of state transformations. Write

$$\Phi^f_k(a) := X^f_k(a), \qquad 0 \le k \le L, \tag{4}$$

for the teacher prefix map. This separates the transformation path $\{\Phi^f_k\}_{k=0}^L$ from the particular local field used to generate it.

Target class. For $s \in \{1, 2\}$, let $\mathcal{F}_{d,s}$ be the vector fields $f : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$ supported in the fixed cube $[-1, 1]^d$ and satisfying, uniformly in t,

$$\|f_t\|_\infty \le 1, \qquad \mathrm{Lip}(f_t) \le 1, \qquad \|f_t - f_u\|_\infty \le |t - u|. \tag{5}$$

For $s = 2$ we additionally assume $f_t \in C^{1,1}$ and $\mathrm{Lip}(Df_t) \le 1$ uniformly in time. Dimension d is fixed, and constants may depend on it. The shared representation uses the cross-depth regularity of f_t; for $s = 2$, this condition concerns field values and places no time regularity requirement on Df_t.

In residual-block notation these assumptions imply

$$\|r_k\|_\infty \le L^{-1}, \qquad \mathrm{Lip}(r_k) \le L^{-1}, \qquad \|r_{k+1} - r_k\|_\infty \le L^{-2}. \tag{6}$$

Thus the class consists of small, smoothly varying update functions across layers, with bounds uniform on the state domain. The common generating field is an assumption of the theorem, not a consequence of having small trained residual increments.

Three fidelity objects. The common transformation path is the sequence $\{\Phi^f_k\}_{k=0}^L$. We measure its reproduction by

$$d_{\mathrm{path}}(f, g) = \max_{0 \le k \le L} \sup_{a \in [-1,1]^d} \|\Phi^f_k(a) - \widehat{X}^g_k(a)\|. \tag{7}$$

Endpoint fidelity keeps only the term $k = L$. Full-field fidelity instead asks for pointwise reconstruction of the sampled local update rule on the common buffer cube $\Omega = [-2, 2]^d$,

$$d_{\mathrm{field}}(f, g) = \max_{0 \le k < L} \sup_{x \in \Omega} \|f(x, t_k) - g(x, t_k)\|. \tag{8}$$

The path metric measures how local updates accumulate, whereas the field metric compares the local rules themselves on the full sampled domain. Different rules need not be approximated to the same accuracy to meet a specified path tolerance. Requiring intermediate transformations is our chosen comparison task; endpoint fidelity remains appropriate when only the final map matters.

Fewer calls with linear output. For the coarse-execution result we choose original-grid knots $0 = t_0 < t_1 < \cdots < t_M = 1$, execute $y_{j+1} = y_j + (t_{j+1} - t_j) g(y_j, t_j)$, and linearly interpolate the resulting nodes to form $\bar{y}(t, a)$. The fidelity requirement is

$$\max_{0 \le k \le L} \sup_{a \in [-1,1]^d} \|\bar{y}(kh, a) - \Phi_k^f(a)\| \le \varepsilon. \tag{9}$$

This changes the execution and readout in (3) while evaluating error at every original prefix. An interval's outputs are emitted after its right node has been computed. Readout uses adjacent executed states; the neural evaluator continues to receive only current state and time. The description budgets below use the original $M = L$ execution.

Description budgets. To ask what the whole process needs to store, we measure finite description length in bits, taking the worst case over the field class. For fixed (ε, L), choose an encoder–decoder pair before the target is known. The encoder maps each $f \in \mathcal{F}_{d,s}$ to a finite binary string c_f; the decoder maps that string to a legal sampled evaluator. We write $B_{\mathrm{path,all}}^{\mathcal{F}_{d,s}}(\varepsilon, L)$ for the infimum, over all such target-independent representation contracts, of

$$\sup_{f \in \mathcal{F}_{d,s}} |c_f| \quad \text{subject to} \quad d_{\mathrm{path}}(f, \mathcal{D}(c_f)) \le \varepsilon \quad \text{for every } f \in \mathcal{F}_{d,s}. \tag{10}$$

The subscript "all" means that the decoded representation is otherwise unrestricted; in particular compressed untied representations are legal. The budget $B_{\mathrm{path,shared}}^{\mathcal{F}_{d,s}}$ imposes the additional requirement that $\mathcal{D}(c_f)$ be one finite sparse time-conditioned network used at every original Euler step. Endpoint and field budgets are defined analogously and are denoted by $B_{\mathrm{endpoint,all}}$ and $B_{\mathrm{field,all}}$.

Code length is target-dependent finite description length under this fixed decoder contract, not allocated parameter count. Thus target-independent compiler structure allowed by the public contract is side information, whereas every target-dependent coefficient, graph choice, clock datum, or other stored choice must be encoded in c_f. Separately, the shared upper construction below is given a conservative full sparse serialization: graph indices, all coefficient copies and their precision, clock logic, selectors, inverse iterations, and neural arithmetic are explicitly counted. That serialized realization has the same leading accuracy power as the target-dependent code. Per-call work, internal network depth, and the number L of outer Euler calls remain separate resources. The target may change the code and therefore the decoded network; the decoder contract itself may depend on the class, (ε, L), and fixed dimension, but not on the unknown target f.

Throughout, $\widetilde{O}$, $\widetilde{\Omega}$, and $\widetilde{\Theta}$ suppress factors polynomial in logarithms of the declared resolution parameters, in particular L/ε for the shared realization.

3 Information in the path and its shared execution

Consider first a field with a Lipschitz Jacobian ($s = 2$) in two state dimensions. Preserving its endpoint to accuracy ε has worst-case information power ε^{-1}. Our result shows that preserving every intermediate transformation has that same leading power, whereas uniformly reconstructing the local field has power ε^{-2} on a sufficiently fine common grid. In one state dimension the distinction changes: the endpoint power is $1/2$ and the path power is $3/4$. Thus retaining the process has a quantifiable cost, and that cost depends on spatial regularity and dimension.

The following theorem gives the general law. Spatial variation and coherent variation across depth produce two lower bounds; the larger exponent also controls a constructive shared upper bound.

Theorem 1 (All-prefix information and executable realization)**.** *Fix $d \geq 1$ and $s \in \{1, 2\}$. There are constants $\varepsilon_d, c_d, C_d > 0$ such that, for every $0 < \varepsilon \leq \varepsilon_d$ and every dyadic $L \geq C_d \varepsilon^{-1/2}$,*

$$\begin{aligned} c_d \varepsilon^{-\beta(d,s)} &\leq B^{\mathcal{F}_{d,s}}_{\mathrm{path,all}}(\varepsilon, L) \\ &\leq B^{\mathcal{F}_{d,s}}_{\mathrm{path,shared}}(\varepsilon, L) \qquad\qquad \beta(d,s) = \max\left\{\frac{d}{s}, \frac{1+d/s}{2}\right\}. \\ &\leq C_d \varepsilon^{-\beta(d,s)} \operatorname{polylog}(L/\varepsilon), \end{aligned} \tag{11}$$

The shared decoder returns one finite sparse time-conditioned neural evaluator used at all L original Euler steps. In addition to the target-dependent code bound in (11)*, the construction admits a complete explicit sparse serialization whose graph, copied coefficients, clock, precision, inverse computation, and neural arithmetic have the same leading upper power, up to polynomial logarithms. The network and code may depend on the target and L, while the decoder contract is fixed across the class. Its internal depth is polylogarithmic in L/ε, and its per-call sparse arithmetic has the same leading upper order as the serialized sparse graph, up to polynomial logarithms. This theorem uses L outer calls.*

For $s = 1$, the lower packing may be chosen so that every frozen-time teacher block $f(\cdot, t_k)$ is a finite ReLU network, and the shared upper construction is ReLU. For $s = 2$, the smoothness assumption excludes nonlinear finite ReLU blocks; finite squared-ReLU frozen-time blocks provide an activation-matched lower family, and a shared squared-ReLU realization attains the same leading upper power. The constants depend on d, and the rates suppress logarithmic factors, including at $d = s$.

The theorem becomes a calibration of depth once compared with the two adjacent fidelity contracts.

Corollary 1 (Endpoint, path, and field calibration)**.** *For fixed d and $s \in \{1, 2\}$, endpoint fidelity satisfies, uniformly over dyadic L,*

$$B^{\mathcal{F}_{d,s}}_{\mathrm{endpoint,all}}(\varepsilon, L) = \widetilde{\Theta}(\varepsilon^{-d/s}), \tag{12}$$

whereas in the regime of Theorem 1,

$$B^{\mathcal{F}_{d,s}}_{\mathrm{path,all}}(\varepsilon, L) = \widetilde{\Theta}\Big(\varepsilon^{-\max\{d/s, (1+d/s)/2\}}\Big). \tag{13}$$

On the common sampled field domain Ω, if $L \geq C_d \varepsilon^{-1}$,

$$B^{\mathcal{F}_{d,s}}_{\mathrm{field,all}}(\varepsilon, L) = \widetilde{\Theta}(\varepsilon^{-(1+d/s)}). \tag{14}$$

The endpoint leading rate is also attainable by a target-dependent shared conditioned evaluator, without an all-prefix guarantee.

The distinction between three fidelity requirements. Equations (12)–(14) give the qualitative boundary $d = s$. When $d < s$, temporal variation increases the leading all-prefix exponent above the endpoint exponent. When $d \geq s$, preserving every residual prefix and preserving only the endpoint have the same leading accuracy power, while full-field reconstruction is polynomially more expensive. The equality concerns the leading accuracy exponent, up to logarithmic factors. For the stated integer $d \geq 1$ and $s \in \{1, 2\}$, the case $d < s$ is $(d, s) = (1, 2)$.

Where the two lower-bound powers come from. A spatial packing with $\Theta(\varepsilon^{-d/s})$ independently signed cells contains $\exp(\Omega(\varepsilon^{-d/s}))$ distinguishable paths. For the joint state–depth packing, a depth block of length δ produces only order δ^2 prefix displacement under the time-Lipschitz condition. Taking $\delta \asymp \sqrt{\varepsilon}$ yields about $\varepsilon^{-1/2}$ depth blocks, with about $\varepsilon^{-d/(2s)}$ compatible spatial choices per block, producing the power $(1 + d/s)/2$. The maximum in Theorem 1 is therefore a competition between spatial complexity and additional depth variation.

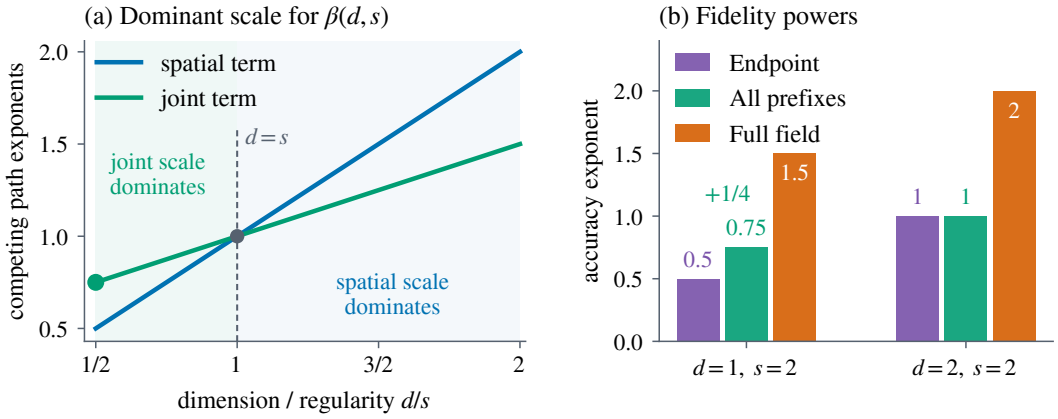


Figure 2: **The path power is set by the harder of two scales.** (a) Spatial variation contributes d/s and joint state–depth variation contributes $(1 + d/s)/2$; they tie at $d = s$. (b) For the allowed $d < s$ case $(d, s) = (1, 2)$, retaining all prefixes raises the endpoint exponent from $1/2$ to $3/4$. At $(d, s) = (2, 2)$ they both equal 1, while full-field reconstruction has exponent 2. The lines display the exponent formulas; theorem instances use integer $d \geq 1$ and $s \in \{1, 2\}$. Powers suppress logarithms; the path bound uses dyadic $L \gtrsim \varepsilon^{-1/2}$ and the field bound $L \gtrsim \varepsilon^{-1}$.

How a shared network executes the description. The construction stores local transports at several depth scales, reusing nearby portions of the path. At scale Δ, there are $O(1/\Delta)$ intervals and each local map needs $O((\Delta/\eta)^{d/s})$ nodal values. Thus, up to logarithmic precision factors, total storage is

$$S \lesssim \eta^{-d/s} \sum_{\Delta \geq \delta} \Delta^{d/s-1}, \qquad \delta \asymp \sqrt{\varepsilon}, \quad \eta \asymp \varepsilon / \operatorname{polylog}(1/\varepsilon).$$

The finest scale dominates when $d < s$, the coarsest when $d > s$, and a logarithm appears at $d = s$. Given a current state and time, inverse path coordinates recover the location needed to evaluate the next update without an initial-state oracle or persistent memory. Selectors, inverse iterations and arithmetic are compiled into one finite network. The original Euler grid resolves every segment of the constructed path, so the ideal updates follow that path and the compiler error can be controlled. Appendix A gives the proof and Appendix B details the realization costs.

The online interface matters here: the evaluator receives current state and time, not the initial input. Counted inverse computations recover the needed coordinates, so the shared network attains the unrestricted leading information power. Figure 3 shows this online call. Appendix C reports a finite compilation of one saved multiscale code into shared and untied graphs.

3.1 From anchor spacing to a call bound

The same construction gives a simple answer to how many calls suffice. It stores anchor maps Q_j at times $t_j = j\delta$, with $\delta \asymp \sqrt{\varepsilon}$, and joins them by linear interpolation. At an anchor, the ideal velocity is the relative displacement divided by δ:

$$g(y, t_j) = \frac{Q_{j+1} \circ Q_j^{-1}(y) - y}{\delta}, \qquad y + \delta g(y, t_j) = Q_{j+1} \circ Q_j^{-1}(y). \tag{15}$$

The evaluator is constructed from the encoded path: at anchor times, one update realizes the relative transport between adjacent anchors exactly before compiler error. Linear readout recovers the stored path between those nodes, and each call stays within one segment.

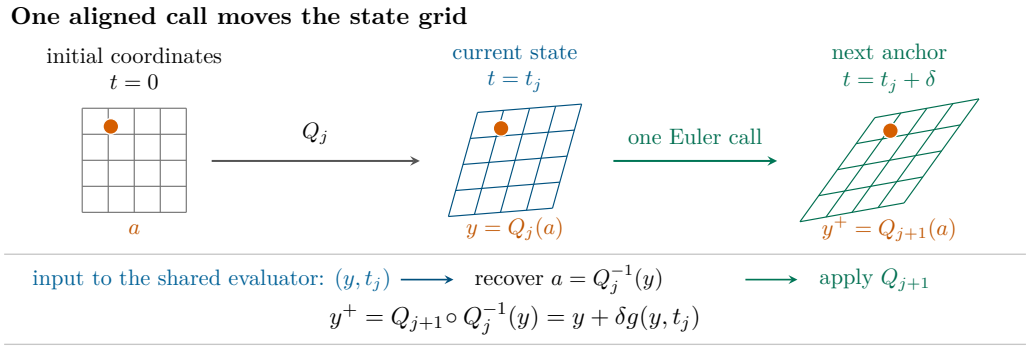


Figure 3: **One current-state call advances the encoded path.** The thin grids depict illustrative invertible anchor maps on many initial states; the orange point tracks one input. At t_j the shared evaluator receives (y, t_j), recovers $a = Q_j^{-1}(y)$ through its counted inverse circuit, and makes one aligned Euler call to $y^+ = Q_{j+1}(a)$. Both anchor maps use the same dyadic code. The grid deformation is schematic; linear readout between anchors supplies the original-prefix outputs in Corollary 2.

Corollary 2 (Aligned calls and linear-output resolution)**.** *Under the assumptions and resolved-grid regime of Theorem 1, with constants adjusted if necessary, the same shared construction admits a dyadic, original-grid-aligned execution with*

$$M \le C_d \varepsilon^{-1/2}, \qquad \max_{k \le L} \sup_{a \in [-1,1]^d} \|\bar{y}(kh, a) - \Phi_k^f(a)\| \le \varepsilon. \tag{16}$$

Here $\bar{y}$ is linear interpolation of the executed nodes as in (9)*. The complete sparse serialization remains $C_d \varepsilon^{-\beta(d,s)} \operatorname{polylog}(L/\varepsilon)$, and the evaluator receives only current state and time. Conversely, there is a target in $\mathcal{F}_{d,s}$ for which any piecewise-linear output with M segments and knots on the original grid satisfying* (9) *requires $M \ge c_d \varepsilon^{-1/2}$, even with arbitrary knot selection and node values.*

For the upper bound, the coordinate $Q_j^{-1}(y_j)$ drifts by at most $C\delta\nu$ per call when the compiled velocity error is ν. Across $M = 1/\delta$ calls the error is therefore $O(\nu)$; the already-counted finite precision suffices. The lower bound uses a quadratic trajectory: a straight segment spanning duration H has error of order H^2 at the sampled interior points. Appendix G proves both statements, including the domain and rounding budgets. The lower bound characterizes the resolution required by the linear output in Equation (9).

The correspondence is now explicit: once the residual grid resolves the requested accuracy, a shared representation of the whole path can be executed at an aligned resolution $M = O(\varepsilon^{-1/2})$, independently of further increases in L. Work per call is still charged, and emitting all $L + 1$ outputs takes $O(Ld)$ readout operations. This construction samples a network tailored to the original grid at its legal clock times. Next we diagnose call alignment on a fixed matrix path.

4 How should the stored path be executed?

Corollary 2 places calls at every stored anchor. Moving or removing them may cross a bend that one local velocity cannot resolve. To evaluate another clock, we compare each update with the displacement required by the path and accumulate the defects. We give a criterion for a fixed, known two-dimensional invertible matrix path, then examine saved ReLU executions. This diagnostic checks call alignment and conditional error allocation for those paths; the nonlinear theorem's worst-case rate rests on its proof. The criterion uses the proposed clock and path offline; the network still receives only current state and time. Its proof and nonlinear extension are in Appendix D.

Three resolutions of the same object. For a fixed two-dimensional path $D(t)$, let R_L linearly interpolate the nodes of an L-block residual reference using the exact relative maps of D. A code P_N interpolates $N+1$ anchors, and one graph approximating $P_N'(t)Q_4(P_N(t))y$ makes M Euler calls, where $Q_4(P) = \sum_{r=0}^{4}(I-P)^r$. Thus L, N and M resolve the reference, stored path and execution separately. Native Euler and direct-field controls are specified in Appendix C; the displayed residual reference uses exact relative maps.

Both outputs linearly interpolate their executed nodes. Write E_{pair} for their complete-time difference; E_{res} and E_{flow} for their errors to D; E_{code} for the code error; and E_{exec} for execution error to P_N. Then

$$E_{\mathrm{pair}} \le E_{\mathrm{res}} + E_{\mathrm{flow}}, \qquad E_{\mathrm{flow}} \le E_{\mathrm{code}} + E_{\mathrm{exec}}. \tag{17}$$

These bounds separate reference, representation and execution error.

A conditional error interval. At call times t_j, let $P_j = P_N(t_j)$ and $A_j = I + h_j F_j$ be the ideal linear update. Its path-coordinate defect is

$$C_j = P_{j+1}^{-1} A_j P_j - I$$

$C_j = 0$ means the update carries $P_j a$ exactly to $P_{j+1}a$. Form $Z_j = P_j(I + \sum_{i<j} C_i)$, retaining the directions of the accumulated defects. Let $\bar{Z}$ linearly interpolate these predicted node maps. For a reference map path B, set $q_B = \sup_{t,a\in\mathcal{A}} \|(\bar{Z}(t) - B(t))a\|$, where $\mathcal{A}$ is the declared initial-state domain.

Proposition 1 (Conditional full-path calibration)**.** *Suppose P_N is invertible with controlled norms on the required domains, $P_0 = I$, and the updates start at $a \in \mathcal{A}$. Both actual and predicted outputs use linear interpolation of their executed nodes. Let $\sigma_j = \sum_{i<j} \|C_i\|_2$, $R = \sup_{a\in\mathcal{A}} \|a\|_2$, and*

$$r = R \max_j \big\{ \|P_j\|_2 (e^{\sigma_j} - 1 - \sigma_j) \big\}.$$

If the neural output differs from the ideal arithmetic output by at most b_{comp} over the full output path, put $\eta = r + b_{\mathrm{comp}}$. Then its full-path error E_B against B obeys

$$\max(0, q_B - \eta) \le E_B \le q_B + \eta. \tag{18}$$

Taking $B = R_L$ and $B = D$ gives pair and target criteria: both upper bounds below ε imply pass, either lower bound above ε implies failure, and the rest remain undetermined. Reference, coding and compiler allocation gates are separate. A minimum passing call count within a candidate set also requires checking all smaller-call candidates.

The interval retains first-order defect directions and bounds higher-order products. On six near-identity targets, its decisions for 96 saved executions agree with measured outcomes at two reference depths and three thresholds. Appendix D gives the domain and arithmetic checks, then shows how larger defects make a separate matrix-path stress undetermined.

4.1 An alignment check

We isolate call placement by keeping one affine path code, its saved network and the multiset of step lengths fixed. Only the order of the steps changes. The broader reference-depth and coding-resolution matrix, including its allocation gates, is reported in Appendix C.

A step's position relative to the path matters. For exact inversion, Euler follows a single affine segment of P_N exactly: $x = P_N(t)a$ implies $x + hP_N'(t)P_N(t)^{-1}x = P_N(t+h)a$ whenever the step stays in that segment. This identity explains the aligned-grid plateau. Reordering the same steps changes error without changing the graph, call count or step-size histogram (Figure 4). Exact relative transports recover executed nodes, yet linear output across skipped bends still incurs full-path error. Both update and output resolution enter calibration.

Reference depth, stored path resolution and call placement affect distinct errors. The aligned construction gives an accuracy-dependent call scale; the defect interval evaluates other schedules for a specified path.

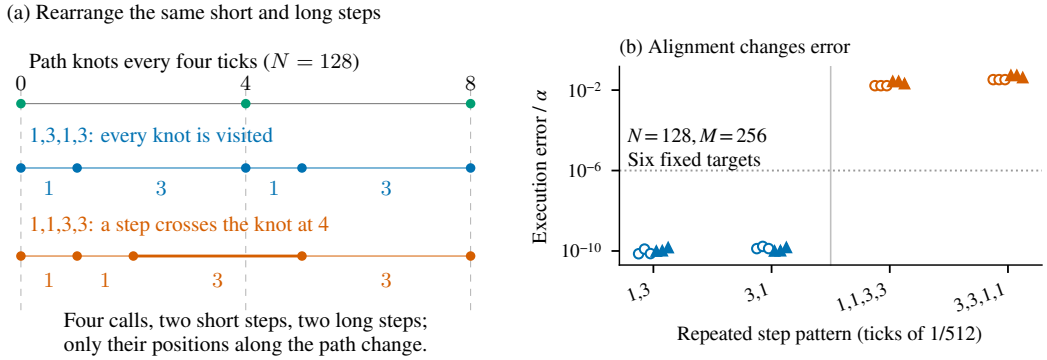


Figure 4: **Call placement changes execution error at fixed step multiset.** (a) Both schedules make four calls with the same two short and two long steps; only the orange schedule crosses the path knot. Vertical positions are schematic. (b) Saved ReLU executions at $N = 128, M = 256$ on six targets: each schedule has 128 one-tick and 128 three-tick steps, all on the valid $1/512$ clock lattice. Hollow circles and filled triangles mark the two phases. All six targets have $K = 8$ and amplitude $\alpha = 1/(4096K^2) = 2^{-18}$; the ordinate is full-path execution error normalized by this amplitude, E_{exec}/α. The dotted line marks 10^{-6}.

5 What the correspondence tells us about depth

The path-information law measures what must be represented, the aligned-call bound gives an execution resolution, and the conditional interval evaluates other schedules. A fixed smooth path can be subdivided without changing its intrinsic state propagation between fixed positions; depth-scaled residual increments have corresponding stability bounds. Thus more layers may refine execution rather than add transformation detail.

Forward path accuracy alone does not control gradients. Uniformly close paths in the theorem's $s = 1$ class can have different state Jacobians, and parameter optimization also depends on parameter features. Appendix E develops these distinctions and the depth-stability analysis (Hardt & Ma, 2017; Marion et al., 2024).

The correspondence makes resource allocation a precise question: how much path detail should a representation retain, and which calls resolve that description at the desired accuracy? It also specifies what a conversion from layer-specific updates to shared execution must preserve. The counted construction establishes such an execution for the stated class; optimizing the allocation or learning a conversion requires further work. For this class, one current-state/time representation reaches the optimal leading path-information power and, with linear readout, the optimal worst-case call resolution. How much a process stores and how finely it must be executed are related but distinct measures of depth.

Reproducibility Statement

Appendices A, B, F and G give the assumptions, proofs and implementation accounting. The anonymous supplementary materials contain the complete measured depth/call and step-order result tables, conditional-interval decisions, multiscale execution records, and sources for replaying Figures 2, 4(b) and 5. They also include the corrected amplitude-stress inputs, arithmetic source and dense-time verifier for Appendix D. The README gives setup, commands and expected outputs. Complete network reruns require saved neural graphs and full rollout arrays, which are outside this bundle.

AI USE STATEMENT

We used OpenAI Codex to help formulate and critique theoretical claims, inspect proof arguments and related literature, design and implement bounded numerical checks, create figures, interpret those checks, prepare reproducibility materials, and draft and edit the paper. The authors are responsible for the final claims, results, and manuscript.

REFERENCES

Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Jørn-Henrik Jacobsen. Invertible residual networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Nicholas M. Boffi, Michael S. Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *arXiv preprint arXiv:2406.07507*, 2025.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

Wolfgang Dahmen. Compositional sparsity, approximation classes, and parametric transport equations. *Constructive Approximation*, 2025. doi: 10.1007/s00365-024-09699-9.

Weinan E, Chao Ma, and Lei Wu. The barron space and the flow-induced function spaces for neural network models. *Constructive Approximation*, 2022.

Tomer Galanti and Lior Wolf. On the modularity of hypernetworks. In *Advances in Neural Information Processing Systems*, 2020.

Adityanand Guntuboyina and Bodhisattva Sen. Covering numbers for convex functions. *IEEE Transactions on Information Theory*, 2013.

Moritz Hardt and Tengyu Ma. Identity matters in deep learning. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1611.04231.

Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng. ReLU deep neural networks and linear finite elements. *arXiv preprint arXiv:1807.03973*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Zhongjie Li, Hao Liu, Carlangelo Liverani, and Enrique Zuazua. Universal approximation of dynamical systems by semi-autonomous neural odes and applications. *SIAM Journal on Numerical Analysis*, 2026.

Yuying Liu, J. Nathan Kutz, and Steven L. Brunton. Hierarchical deep learning of multiscale differential equation time-steppers. *Philosophical Transactions of the Royal Society A*, 380(2229): 20210200, 2022. doi: 10.1098/rsta.2021.0200.

Pierre Marion. Generalization bounds for neural ordinary differential equations and deep residual networks. In *Advances in Neural Information Processing Systems*, 2023.

Pierre Marion, Yu-Han Wu, Michael E. Sander, and Gérard Biau. Implicit regularization of deep residual networks towards neural ODEs. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.01213.

D. Stelter, T. Wilde, C. Rössl, and H. Theisel. Anchor flow maps: Efficient sampling and approximation of the entire flow map. *Computer Graphics Forum*, 2026. doi: 10.1111/cgf.70474.

Vladimir N. Temlyakov. On the entropy numbers of the mixed smoothness function classes. *arXiv preprint arXiv:1602.08712*, 2016. URL https://arxiv.org/abs/1602.08712.

Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

A PROOF OF THE CALIBRATION THEOREM

We prove Theorem 1 in five steps. Throughout, constants may depend on the fixed dimension d. All maps are evaluated on a fixed buffer cube containing the teacher trajectories. The local approximants below are extended to this cube, and iterative inverse queries are projected back to it; projection does not worsen their contraction estimates. Since $\|f_t\|_\infty \le 1$ and the horizon is one, the cube can be chosen independently of L and ε.

A.1 COHERENT MULTISCALE PATH COORDINATES

For a grid-aligned interval $I = [ph, qh]$ of duration $\Delta = (q-p)h$, let T_I denote the exact Euler transport from time ph to qh. Suppose a stored approximation has residual form

$$P_I = \mathrm{id} + r_I, \qquad \|P_I - T_I\|_\infty \le \eta, \qquad \mathrm{Lip}(r_I) \le \kappa\Delta. \tag{19}$$

Choose a fixed dyadic macroblock length Δ_0 so small that $a_0 = e^{\kappa\Delta_0} - 1 \le 1/5$. Let the anchor spacing δ be dyadic and grid aligned, $h \mid \delta \mid \Delta_0$. Each anchor prefix is represented by all completed macroblock roots together with the canonical dyadic decomposition of the current macroblock prefix. Denote the resulting map by Q_j at anchor time $j\delta$. The number of selected factors obeys

$$m \le C + \log_2(1/\delta), \tag{20}$$

and the durations of those factors sum to at most one.

Lemma 1 (Coherent anchor maps)**.** *If every selected local map satisfies* (19)*, then*

$$\|Q_j - \Phi^f_{j\delta/h}\|_\infty \le Cm\eta, \qquad \mathrm{Lip}(Q_j) + \mathrm{Lip}(Q_j^{-1}) \le C. \tag{21}$$

Moreover, for adjacent anchors,

$$R_j := Q_{j+1}Q_j^{-1} - \mathrm{id} \quad \textit{satisfies} \quad \mathrm{Lip}(R_j) \le \frac{1}{2}. \tag{22}$$

Proof. Replace the exact factors in the prefix composition one at a time. Teacher transport over duration u has Lipschitz constant at most e^u, while each approximate residual factor has Lipschitz constant at most $1 + \kappa\Delta$. Since the selected interval lengths sum to at most one, the standard telescoping composition estimate gives the first bound in (21).

Each $P_I = \mathrm{id} + r_I$ with $\mathrm{Lip}(r_I) < 1$ is invertible by contraction iteration, with $\mathrm{Lip}(P_I^{-1}) \le (1-\kappa\Delta)^{-1}$; this is the standard residual-inversion mechanism used by invertible residual networks (Behrmann et al., 2019). Therefore

$$\prod_I (1-\kappa|I|)^{-1} \le \exp\left(\frac{\kappa\sum_I |I|}{1-\kappa\Delta_0}\right),$$

which is uniformly bounded, proving the inverse-Lipschitz part of (21).

For adjacent anchors, factor out a common prefix so that $Q_j = H_j P$ and $Q_{j+1} = H_{j+1}P$. If both anchors lie in one macroblock, take P through that block's left boundary. If the second anchor is the right boundary of a macroblock, use the previous block's left boundary; the two suffix durations are then at most Δ_0. If the first anchor is a macroblock boundary, include the just-completed macroblock root in P; then $H_j = \mathrm{id}$ and H_{j+1} has duration δ. Thus in every case both suffixes have duration at most Δ_0. Hence $\mathrm{Lip}(H_i - \mathrm{id}) \le a_0$ and $\mathrm{Lip}(H_i^{-1}) \le (1-a_0)^{-1}$. Thus

$$R_j = (H_{j+1} - H_j)H_j^{-1}, \qquad \mathrm{Lip}(R_j) \le \frac{2a_0}{1-a_0} \le \frac{1}{2}.$$

□

For $t = j\delta + \lambda\delta$, $\lambda \in [0,1]$, interpolate adjacent anchors by

$$D_t = (\mathrm{id} + \lambda R_j)Q_j = (1-\lambda)Q_j + \lambda Q_{j+1}. \tag{23}$$

Lemma 1 gives uniform Lipschitz bounds for both D_t and D_t^{-1}. The teacher prefix maps have discrete second difference

$$\|\Phi^f_{k+2} - 2\Phi^f_{k+1} + \Phi^f_k\|_\infty \le 2h^2, \tag{24}$$

because f is one-Lipschitz in both state and time and each Euler displacement has norm at most h. Piecewise-linear interpolation between anchor prefixes, together with (21), therefore yields

$$\max_{0\le k\le L} \|D_{kh} - \Phi^f_k\|_\infty \le C(\delta^2 + m\eta). \tag{25}$$

Choose a dyadic $\delta \asymp \sqrt{\varepsilon}$ such that $h \mid \delta \mid \Delta_0$, and set $\eta \asymp \varepsilon/m$. Because both the outer grid and hierarchy are dyadic, such a scale exists after changing fixed constants whenever $L \ge C_d\varepsilon^{-1/2}$. Then (25) is at most $C\varepsilon$.

A.2 Cost of the local transports

Write $T_I = \mathrm{id} + U_I$. Spatial Lipschitzness of the velocity gives

$$\|U_I\|_\infty \le \Delta, \qquad \mathrm{Lip}(U_I) \le e^\Delta - 1 \le C\Delta. \tag{26}$$

For $s = 2$, differentiating the Euler recurrence and comparing Jacobians at two initial states gives the additional estimate

$$\mathrm{Lip}(DU_I) \le C\Delta. \tag{27}$$

Indeed, a duration-u prefix has Jacobian norm at most e^u, and the difference recurrence contains a geometric sum of $h\,\mathrm{Lip}(Df_t)e^{2u}$ multiplied by the remaining bounded Jacobian factors. No time regularity of Df_t is used.

Interpolate U_I on a regular simplicial mesh of the fixed buffer cube with width

$$\rho_I \asymp (\eta/\Delta)^{1/s}. \tag{28}$$

Because the teacher is continuous and supported in $[-1,1]^d$, the exact transport residual U_I vanishes outside that cube; extending it by zero therefore creates no boundary jump. For $s = 1$, the interpolation error follows from (26); for $s = 2$, barycentric interpolation cancels the affine Taylor term and (27) controls the remainder. Quantize each nodal vector value at resolution $c_d\eta$. The resulting piecewise-affine residual r_I satisfies

$$\|r_I - U_I\|_\infty \le \eta, \qquad \mathrm{Lip}(r_I) \le C_d\Delta, \qquad S_I \le C_d(\Delta/\eta)^{d/s}, \tag{29}$$

where S_I is the number of stored nodal degrees of freedom, up to fixed dimension-dependent factors. Shape regularity gives the Lipschitz estimate; quantization contributes at most $C_d\eta/\rho_I \le C_d\Delta$.

Regular simplicial nodal functions have exact finite ReLU realizations (He et al., 2018); controlled-Lipschitz value approximation is also an established ingredient in approximation-class analyses of transport (Dahmen, 2025). We use these only as local representation ingredients. The whole-path rate comes from how the local maps are organized and executed. In particular, (29) verifies (19) with a fixed $\kappa = C_d$; Δ_0 is chosen once for this constant.

At scale Δ there are $O(1/\Delta)$ dyadic intervals. Summing (29) over scales from Δ_0 down to δ gives

$$S \le C_d\eta^{-d/s} \sum_{\Delta\in\{\Delta_0,\Delta_0/2,\dots,\delta\}} \Delta^{d/s-1}. \tag{30}$$

The sum is bounded by a constant if $d > s$, is logarithmic if $d = s$, and is of order $\delta^{d/s-1}$ if $d < s$. With $\delta \asymp \varepsilon^{1/2}$ and $\eta \asymp \varepsilon/m$,

$$S \le \varepsilon^{-\max\{d/s,(1+d/s)/2\}}\,\mathrm{polylog}(1/\varepsilon). \tag{31}$$

A.3 From path coordinates to one sampled evaluator

The path code is indexed by initial coordinates, whereas the online network receives only the current state. For $t = j\delta + \lambda\delta$, define

$$g(y,t) = \delta^{-1}R_j(z), \qquad z = (\mathrm{id} + \lambda R_j)^{-1}(y). \tag{32}$$

The inverse is well defined by (22). If $y = D_t(a)$, then $z = Q_j(a)$, and hence an aligned Euler step satisfies

$$y + hg(y,t) = D_{t+h}(a) = D_{t+h}D_t^{-1}(y). \tag{33}$$

Thus the ideal evaluator executes the interpolated path exactly on the original Euler grid without carrying a as persistent state.

Now approximate g uniformly at sampled times by a finite neural circuit $\widehat{g}$ with velocity error at most ν on the buffer cube. Let $\widehat{x}_k$ be its Euler trajectory and put $a_k = D_{kh}^{-1}(\widehat{x}_k)$. From (33) and the uniform inverse-Lipschitz bound,

$$\|a_{k+1} - a_k\| \le Ch\nu, \qquad \max_k \|\widehat{x}_k - D_{kh}(a_0)\| \le C\nu. \tag{34}$$

This avoids an exponential stability penalty: the ideal steps telescope in the common path coordinates. Taking $\nu = c\varepsilon$ and combining (34) with (25) gives all-prefix error at most ε after adjusting constants.

It remains to count the circuit. A prefix query selects at most one stored map at each dyadic level plus a fixed number of completed macroblock roots. The inverse query reverses those stages and unrolls contraction iterations. A second contraction inversion computes $(\mathrm{id} + \lambda R_j)^{-1}$. Clock transitions are placed in $[t_j - h/4, t_j)$, which contains no original sample, so the right branch is exact at every grid time t_j. Bounded scalar products can be implemented by standard logarithmic-size ReLU multiplication circuits (Yarotsky, 2017).

More explicitly, at each fine dyadic level only the complete left sibling of the anchor's active cell can appear in the canonical prefix decomposition. Its selector is a difference of two clock steps; completed macroblock roots use one step selector. Therefore all interval circuits at a level can be evaluated under mutually exclusive selectors, and each stored interval circuit appears only once in a prefix-query graph. A query for Q_{j+1} uses the same graph with the clock shifted by δ and saturated at one. If b ranges over internal anchor times, the within-interval coordinate at sampled times can be written

$$\lambda(t) = \frac{t - \delta \sum_b H_b(t)}{\delta}, \tag{35}$$

where each H_b transitions only inside $[b - h/4, b)$. Thus the clock has $O(N)$ breakpoints and logarithmic-bit dyadic coefficients rather than an L-entry target-dependent table.

For completeness, let $N = 1/\delta$, let S be the local-map dictionary size, and put

$$S_0 = S + Nd + md. \tag{36}$$

Let $q_0 = \kappa\Delta_0 < 1/4$, which follows from the choice of Δ_0. If μ bounds each approximate scalar-product output, then the selected forward query and an R-step inverse query satisfy

$$e_{\mathrm{forward}} \le CS_0\mu, \qquad e_{\mathrm{inverse}} \le C\big(mq_0^R + S_0\mu\big). \tag{37}$$

The inverse estimate uses geometric accumulation inside each contraction and the uniformly bounded products of forward and inverse Lipschitz constants from Lemma 1. Consequently R inner inverse iterations, T outer inverse iterations, and scalar product error μ give

$$\|\widehat{g} - g\|_\infty \le \frac{C}{\delta}\left(2^{-T} + mq_0^R + S_0\mu\right). \tag{38}$$

Choose

$$T = \left\lceil \log_2 \frac{C}{\delta\nu} \right\rceil, \qquad R = \left\lceil \log_{1/q_0} \frac{Cm}{\delta\nu} \right\rceil, \qquad \mu \le \frac{\delta\nu}{CS_0}. \tag{39}$$

The compiled graph then has at most $S_0\,\mathrm{polylog}(L/\varepsilon)$ nonzero coefficients and polylogarithmic depth. Since $N = O(\varepsilon^{-1/2})$ is lower order than the exponent in (31), its sparse size has the same leading power as S.

Finally, if the compiled graph has U coefficients, depth D, and coefficient magnitude bounded by W, then $\log U + \log W = \mathrm{polylog}(L/\varepsilon)$. Choosing coefficient precision

$$b \ge C\left[\log \frac{L}{\varepsilon} + D\log(UW + C)\right] \tag{40}$$

covers coefficient rounding and an $O(\varepsilon/L)$ arithmetic allowance per outer update. Graph indices cost $O(\log U)$ bits each, so the total finite description remains $\varepsilon^{-\beta(d,s)}\,\mathrm{polylog}(L/\varepsilon)$. This proves the shared upper bound in Theorem 1.

A.4 MATCHING LOWER PACKINGS

We use two finite families. Let $w > 0$ be dyadic and place $n_{\mathrm{cell}} \asymp w^{-d}$ disjoint spatial cells in an interior cube. Let ϕ_i be a unit plateau supported in cell i and equal to one near its center c_i. For $s = 1$ choose a piecewise-linear plateau; for $s = 2$ choose a $C^{1,1}$ piecewise-polynomial plateau. A sufficiently small constant $a_d > 0$ makes every field below belong to $\mathcal{F}_{d,s}$.

For the purely spatial family, set

$$f_\sigma(x,t) = a_d w^s \sum_{i=1}^{n_{\mathrm{cell}}} \sigma_i \phi_i(x) e_1, \qquad \sigma \in \{-1,1\}^{n_{\mathrm{cell}}}. \tag{41}$$

Starting at c_i, the trajectory remains in the plateau and every Euler step is a constant translation. Opposite signs in cell i therefore create endpoint separation $2a_d w^s$. Taking $w^s \asymp \varepsilon$ gives a $2^{n_{\mathrm{cell}}}$-element endpoint packing with $n_{\mathrm{cell}} \asymp \varepsilon^{-d/s}$. Any ε-accurate decoder must assign distinct codes to packing elements separated by more than 2ε, hence

$$B^{\mathcal{F}_{d,s}}_{\mathrm{path,all}}(\varepsilon, L) \geq B^{\mathcal{F}_{d,s}}_{\mathrm{endpoint,all}}(\varepsilon, L) \geq c_d \varepsilon^{-d/s}. \tag{42}$$

For the joint state–depth family, put $\delta = w^s$ and divide depth into blocks of length δ. Let ψ be the unit-height triangular tent on $[0,1]$, zero at both endpoints, and set on block j

$$f_\sigma(x,t) = a_d \delta \sum_{i=1}^{n_{\mathrm{cell}}} \sigma_{i,j} \phi_i(x) \psi\left(\frac{t - j\delta}{\delta}\right) e_1. \tag{43}$$

The time-Lipschitz constant is at most $2a_d$, and the spatial constraints are uniform because the cells are disjoint. Center trajectories stay within their own plateaus since the total displacement over the horizon is $O(a_d\delta)$. For every dyadic $L \geq C_d \varepsilon^{-1/2}$, choose the dyadic $\delta \asymp \sqrt{\varepsilon}$ so that δ/h is an even integer; changing the fixed constants suffices. Thus each block contains an even number of Euler steps. The exact left-sampled discrete integral of the triangular tent over a block is $\delta/2$. For any two distinct sign arrays, choose a cell i on which their time words differ and let j be the first differing block for that cell. Their states from initial point c_i agree before that block and differ by $a_d\delta^2$ at its end; differences in other cells do not affect this trajectory because the supports are disjoint. The choice $\delta \asymp \sqrt{\varepsilon}$ therefore yields a $2^{n_{\mathrm{cell}}/\delta}$-element path packing after fixing constants so that the separation exceeds 2ε. Consequently

$$B^{\mathcal{F}_{d,s}}_{\mathrm{path,all}}(\varepsilon, L) \geq c_d \frac{n_{\mathrm{cell}}}{\delta} = c_d \varepsilon^{-(1+d/s)/2}. \tag{44}$$

Combining (42) and (44) proves the lower bound in Theorem 1.

A.5 ACTIVATION MATCHING AND ADJACENT FIDELITY LAWS

For $s = 1$, the plateau in the lower packing can be chosen piecewise linear, so every frozen-time teacher block is a finite ReLU network; the finite-element upper construction is ReLU as well. For $s = 2$, choose tensor products of univariate compactly supported $C^{1,1}$ piecewise-polynomial plateaus. They admit finite squared-ReLU realizations: with $\sigma_2(z) = [z]_+^2$, one has $z^2 = \sigma_2(z) + \sigma_2(-z)$ and therefore exact multiplication $uv = ((u+v)^2 - (u-v)^2)/4$; truncated cubic pieces follow from $z\sigma_2(z) = [z]_+^3$, and finite tensor products can then be built exactly. To match the upper construction to squared-ReLU activation, replace each ReLU gate by

$$q_\tau(z) = \frac{[z+\tau]_+^2 - [z-\tau]_+^2}{4\tau}, \qquad \|q_\tau - [\cdot]_+\|_\infty \leq \tau/4. \tag{45}$$

Choosing dyadic τ within the polynomial-logarithmic circuit error budget changes sparse size only by a constant factor and precision only by logarithmic terms.

For endpoint fidelity, the exact endpoint map has uniformly bounded Lipschitz norm when $s = 1$ and uniformly bounded $C^{1,1}$ norm when $s = 2$. Direct nodal interpolation on a mesh of width $O(\varepsilon^{1/s})$, together with finite quantization, gives

$$B^{\mathcal{F}_{d,s}}_{\mathrm{endpoint,all}}(\varepsilon, L) \leq \varepsilon^{-d/s} \operatorname{polylog}(1/\varepsilon). \tag{46}$$

Indeed, the unrestricted decoder can turn the stored endpoint approximant P into a legal sampled evaluator by using the update $g(x, t_0) = h^{-1}(P(x) - x)$ at the first grid point and zero thereafter; L is part of the fixed representation contract. The spatial packing (41) matches this rate. A shared endpoint evaluator with the same leading power follows by retaining only the fixed number of macroblocks, approximating their maps to $O(\varepsilon)$, and using the same inverse compiler. This construction is not required to reproduce the intermediate teacher prefixes.

For field fidelity, sample time at spacing $O(\varepsilon)$ and space at mesh width $O(\varepsilon^{1/s})$. When $L \geq C_d/\varepsilon$, this gives

$$B^{\mathcal{F}_{d,s}}_{\mathrm{field,all}}(\varepsilon, L) \leq \varepsilon^{-(1+d/s)} \operatorname{polylog}(1/\varepsilon). \tag{47}$$

For the lower bound, reuse (43) with $\delta = w^s \asymp \varepsilon$. For dyadic $L \geq C_d/\varepsilon$, choose this scale so δ/h is even; the grid then contains every tent midpoint. At such a midpoint two opposite signs differ in field value by order ε. The n_{cell}/δ independent signs therefore force $c_d\varepsilon^{-(1+d/s)}$ bits. Equations (46)–(47), together with the two packings, prove Corollary 1.

B Resource accounting contract

The calibration keeps the following resources distinct throughout the paper: stored target-dependent bits, finite precision, learned or generated clock machinery, sparse network size, per-call arithmetic, internal network depth, and the number M of outer Euler calls. State augmentation and persistent execution memory must be declared and charged. The main theorem leaves the state unchanged and retains all original Euler calls. Its aligned-execution corollary uses the same representation with $M = O(\varepsilon^{-1/2})$ and the linear readout specified in (9).

For clarity, the shared construction uses the following accounting convention.

Resource	What is charged
Information length	all target-dependent bits under the fixed public decoder contract
Explicit sparse storage	graph indices, quantized coefficient copies, stored local maps, clock data
Per-call work	selectors, local-map queries, inverse iterations, products, final evaluation
Internal depth	depth of the finite circuit used for one call to the conditioned evaluator
Numerical depth	L in Theorem 1; aligned M in Corollary 2
State	the original d-dimensional state; inverse coordinates are temporary activations
Precision	enough bits for circuit error and $O(\varepsilon/L)$ outer-update arithmetic error
Readout	adjacent-node buffer and interpolation; $O(Ld)$ work for all prefix outputs

Inverse computations and depth-dependent selection are compiled into the finite graph. Their target-dependent choices enter the information code, and their circuitry enters explicit sparse storage. Repeated evaluation contributes to per-call work and the number of outer calls.

The unrestricted information budget permits any decoded representation and is therefore also a lower bound for compressed untied realizations. For each target and grid, the construction produces one conditioned network with a full explicit sparse serialization at the same leading accuracy power. Target-dependent information, realized sparse storage, per-call work and total execution work remain distinct resources.

B.1 Realization mechanism

The architectural question asks how a short description of the path becomes an evaluator queried with current state and depth. The upper bound in Theorem 1 uses three ingredients.

Multiscale local transports. Short Euler transports are close to the identity. Temporal regularity permits nearby prefixes to reuse local transport information, while spatial regularity controls the description cost of each local map. A dyadic hierarchy stores the maps needed to reconstruct arbitrary prefixes without keeping an independent approximation for every layer.

Coherent inverse coordinates. The stored path maps are naturally indexed by an initial path coordinate, but the online evaluator is queried at the current state. For a prefix map D_t, the local velocity can therefore be written schematically through $D_t^{-1}(x)$. The construction maintains each short residual map close enough to the identity that the inverse is stable and can be computed iteratively. This removes any need for an initial-state oracle or persistent execution history.

Same-grid neural compilation. The multiscale selectors, local map evaluations, inverse iterations, products, and clock transitions are compiled into one finite time-conditioned network. In Theorem 1, all original Euler samples are kept. Transition bands are placed away from sampled clock values so that the network agrees with the intended discrete evaluator at every original step. Target-dependent quantities are included in the information code, while a conservative explicit serialization also counts finite precision, graph indices, and every copied coefficient. The internal network depth and per-call work grow only polylogarithmically on top of the sparse serialization order.

Metric entropy gives a compact description of the path; inverse geometry and neural compilation make that description executable by the conditioned architecture under the same state and numerical depth. The full technical construction and stability estimates are in Appendix A. Appendix G shows how the same compiled evaluator executes one whole anchor interval per call, with separate readout costs.

C FINITE CALIBRATION PROTOCOL

The saved affine call-calibration study includes a depth-by-call matrix and a matched-step-order follow-up. Their models, time nodes, trajectories, metrics and scientific specifications are described below. Complete measured result tables and the figure-replay source are included in the anonymous supplementary materials.

Saved multiscale execution. In a separate 32-step shared-code execution check, the same reloaded dyadic code drove one current-state/time affine/ReLU graph and 32 frozen state-only graphs. The two nonlinear teacher fields are as follows. For the one-dimensional pulsed plateau field, put $w = 1/8$, $c \in \{-3/8, 0, 3/8\}$, $b_w(z) = \min\{1, [2 - 2|z|/w]_+\}$ and $\psi(u) = 2\min\{u, 1-u\}$. With $j = \min\{\lfloor 4t \rfloor, 3\}$,

$$f_{\mathrm{pulse}}(x,t) = \frac{1}{2048}\psi(4t-j)\sum_c \sigma_{j,c} b_w(x-c), \qquad \sigma_{j,c} \in \{-1,1\} \quad \text{(saved seed 1703)}.$$

For the two-dimensional localized ReLU field, let $b_\infty(z) = [1 - 2\|z\|_\infty]_+$ and set

$$f_{\mathrm{local}}(x,t) = \frac{1}{16}\big(t b_\infty(x - (1/4,0)), (1-t) b_\infty(x - (-1/4,0))\big).$$

Both teachers use the original $L = 32$ Euler grid on $[-1,1]^d$. The saved code stores every dyadic local transport from level 2 through $\log_2 N$, with 24-bit nodal values. The pulsed field uses $N = 8$ anchors and 32 spatial intervals; the localized field uses $N = 4$ and a four-interval spatial grid per coordinate. Anchors compose those local maps and the code path interpolates between anchors.

The errors below maximize over all 33 original prefixes and 513 pulsed-field or 1345 localized-field initial points: respectively a 257-point or 33^2-point regular grid, plus 256 fixed random points in $[-1,1]^d$. Here E_{code} compares the arithmetic code path with the teacher; "compiler added" compares the reloaded shared graph with 64-iteration arithmetic execution of that same code. These are sampled errors, not uniform certificates.

Known coupled affine paths. For $K \in \{4, 8, 16\}$ let $\alpha_K = 1/(4096K^2)$, and define

$$S(z) = \mathrm{clip}(z,0,1)^2\big(3 - 2\,\mathrm{clip}(z,0,1)\big), \qquad \psi(u) = S(4u) - S(4u-3).$$

With $j = \lfloor Kt + \varphi \rfloor \bmod K$ and $u = Kt + \varphi - \lfloor Kt + \varphi \rfloor$, set

$$A(t) = I + \alpha_K \psi(u)(\sigma_{j,1}E_{12} + \sigma_{j,2}E_{21}), \qquad D(t) = A(t)A(0)^{-1}.$$

The initial-state box is $[-1/4, 1/4]^2$ and the execution buffer is $[-1/2, 1/2]^2$. The examples use affine paths on these bounded domains. All signs are stored as actual values in the serialized targets.

Table 1: **Finite execution of one reloaded dyadic path code.** Graph sizes are decimal MB. The shared graph includes clock and inverse circuits; the untied graph stores all 32 frozen-time copies. The localized field's coarse code has a large teacher error, and both shared graphs are larger.

Fixture	E_{code}	Compiler added	Shared MB	Untied MB
Pulsed 1D	7.69×10^{-6}	8.33×10^{-15}	27.73	12.84
Localized 2D	1.70×10^{-2}	2.57×10^{-10}	28.27	10.33

The first campaign uses two phases, 0 and $1/3$, and seeds 1121–1123, with independent words and paired words that repeat the first sign pair. There are 36 labeled targets and 30 distinct target paths; repeated labels share their underlying path.

Fixed graphs and resolution axes. Residual reference depths are $L \in \{16, 32, 64, 128, 256, 512\}$, call counts $M \in \{1, 2, 4, \ldots, 512\}$, and path resolutions $N \in \{8, 32, 128, 512\}$. Every path or field graph is frozen across all its executions. Sparse affine/ReLU DAGs serialize skip edges, coefficients, biases and indices. A bounded ReLU product circuit and fourth-order Neumann truncation implement current-state inverse coordinates. Computation, internal graph depth, precision, graph bytes and outer calls are separately recorded. Exact relative residual maps, native left-Euler residual maps, the path graph and a separately compiled instantaneous-field graph are distinct controls. Macro residuals are executed from expanded, gzip and actual-sign-generator formats, with decoding costs exposed.

The main campaign comprises 1440 path-graph executions, 360 field-graph executions, 1080 macro residual executions across three formats, and 216 native Euler residual executions. Its 36 initial points are the four box corners and 32 scrambled Sobol points (seed 11301); 128 additional points (seed 11302) check compiler agreement at all 512 declared clock values. The maximum rollout compiler discrepancy divided by α_K is 8.70×10^{-8} on those queries in float64 arithmetic.

Errors and gates. The maximum difference of two piecewise-linear trajectories occurs at a knot of their union, giving the full-time polygon error on the evaluated initial states. Errors to D use 65536 time intervals, with a half-cell Lipschitz remainder based on the analytic target derivative and actual polygon slopes. Intermediate output uses linear interpolation of executed nodes. Total fidelity requires $E_{\mathrm{pair}}, E_{\mathrm{flow}} \le \varepsilon$; the original diagnostic additionally requires $E_{\mathrm{res}}, E_{\mathrm{code}} \le \varepsilon/4$ and compiler error $\le \varepsilon/16$. Relative thresholds are $\varepsilon/\alpha_K \in \{.10, .05, .02\}$. Common absolute thresholds $\alpha_{16}\{.10, .05, .02\}$ are reported separately in the raw results. Product depth was chosen for the relative thresholds. For the $K = 4$ absolute controls, the reported compiler checks are measured values.

The displayed example uses $K = 8$, independent word, phase $1/3$, seed 1121, $L = 256$, and coding resolution $N = 128$. It meets the total $.05\alpha_K$ tolerance for sufficiently many calls but exceeds the separate coding allocation. Figure 5 also shows the $N = 512$ graph as a distinct fixed-capacity curve. For this example, native Euler at $L = 512$ has reference error about $.017516\alpha_K$, above the $.0125\alpha_K$ reference allocation. The main heatmap consequently uses the exact relative-map reference.

Information and execution diagnostic. At fixed $N = L = 512$, all paired independent and repeated sign words have the same minimum passing tested M at each relative threshold. Their structural sign contents are $2K$ and 2 bits, respectively, while complete target files also charge metadata. This observation distinguishes specified sign-code content from the temporal resolution needed to execute a path. The maximum over time is controlled by a local pulse shared by both constructions.

Alignment intervention. The follow-up retains the six independent-word $K = 8$ targets and their original $N = 128, 512$ graphs. At $M = 128$ it compares tick patterns $[4]$, $[3, 5]$ and $[5, 3]$. At $M = 256$ it compares $[2]$, $[1, 3]$, $[3, 1]$, $[1, 1, 3, 3]$ and $[3, 3, 1, 1]$, repeated to fill the horizon. All timestamps are original $1/512$ lattice points; there is no recompilation. The 96 neural trajectories have 96 arithmetic Q4 counterparts and 96 independently executed exact P_N-transport controls. The uniform trajectories reproduce the retained original trajectories bit for bit. At $N = 128, M = 256$,

the four nonuniform schedules have identical step multisets, but only the alternating patterns visit every path knot. Both grouped patterns cross 64 knots, yet give different errors; crossing count alone is insufficient.

For the fixed displayed target, the grouped patterns have neural node errors about $.026803\alpha_K$ and $.053606\alpha_K$. Exact transport controls have negligible node error but about $.017323\alpha_K$ full-path error from linear interpolation across skipped bends. A standard local-defect propagation bound covers all observed trajectories with a conservative margin.

Execution can saturate at the representation error. Figure 5(b) exposes two error regimes. For $N = 128$, increasing M from 128 to 512 leaves error near $.0151\alpha$, the coding floor. For $N = 512$, the error at $M = 128$ is $.0650\alpha$ and falls to $.00127\alpha$ at $M = 512$. A finer stored path changes the local velocity that must be integrated. The coding allocation and the total fidelity criterion identify separate requirements in the displayed example.

C.1 Reference, code and call resolution

Figure 5 shows the three resolution axes in one saved affine example. Its residual reference uses the exact relative maps of D, as specified above; the native Euler control has a different reference error and is reported separately in the raw results.

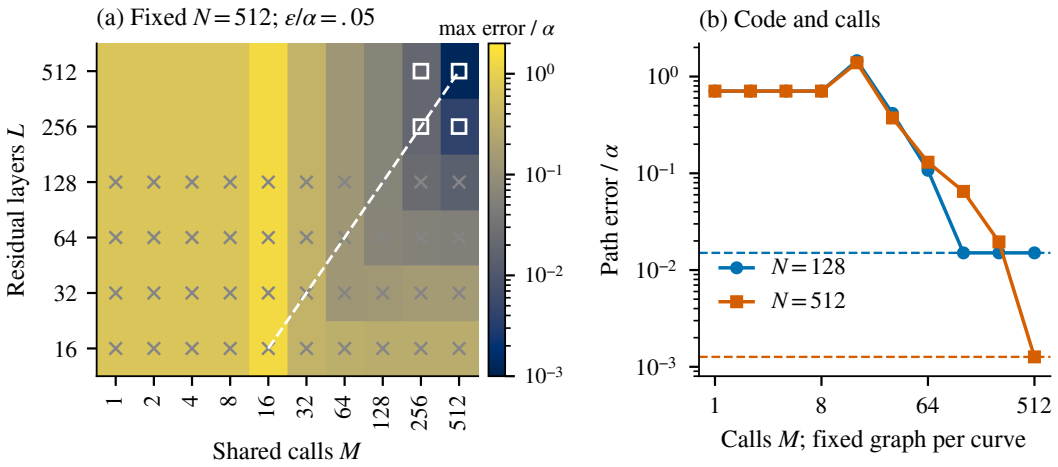


Figure 5: **Reference depth, path-code resolution, and calls are separate axes.** Saved ReLU executions on the $K = 8$, phase $1/3$, seed 1121 path, using the exact relative-map residual reference. (a) Every cell uses the same $N = 512$ graph. Color gives $\max(E_{\text{pair}}, E_{\text{flow}})/\alpha$; white squares meet the total tolerance $\varepsilon = .05\alpha$ and the allocation gates, while gray crosses mark $E_{\text{res}} > \varepsilon/4$. For $L = 256$ and 512, the smallest passing tested M is 256; the dashed line is $M = L$. (b) Each curve fixes a graph and varies its calls; dashed horizontal lines show coding error. At $M = 128$, the finer $N = 512$ code has higher executed error than $N = 128$, while at $M = 512$ it has lower error. The $N = 128$ plateau of $.0151\alpha$ meets the total tolerance but exceeds the coding allocation $\varepsilon/4 = .0125\alpha$. Results use four corners and 32 Sobol initial states, with a dense-grid Lipschitz remainder for reference errors.

The fixed-network heatmap has the same smallest passing tested call count at $L = 256$ and 512. Once $E_{\text{res}} \le \varepsilon/4$, any execution with $E_{\text{flow}} \le 3\varepsilon/4$ satisfies the pair criterion by Equation (17). Thus both residual depths resolve the common path to the allocated reference accuracy. Panel (b) separates the coding floor from execution error: more calls can resolve a stored path more accurately, but cannot remove its fixed coding error. A finer code also changes the velocity to be executed and may require a different call schedule.

D CONDITIONAL EXECUTION ANALYSIS

This appendix derives Proposition 1 for a known path, fixed numerical update and declared linear output. The online network receives current state and time.

D.1 DEFECTS IN NONLINEAR PATH COORDINATES

Let $P_j = P(t_j)$ be invertible maps with $P_0 = \mathrm{id}$, and let U_j be the ideal update from t_j to t_{j+1}. Define

$$\Theta_j = P_{j+1}^{-1} \circ U_j \circ P_j = \mathrm{id} + c_j.$$

The common coordinate domain must contain the initial set $\mathcal{A}$, all actual pulled-back states, and the predicted coordinates below. All maps and inverse queries must be defined on the corresponding domains. Suppose P_j, P_j^{-1} and c_j have Lipschitz bounds κ_j, λ_j and ℓ_j, respectively. These bounds apply throughout the interpolated path as well as at its anchors.

For $y_0(a) = a$, put $u_j = P_j^{-1}(y_j)$, so that $u_{j+1} = u_j + c_j(u_j)$. Set

$$S_j(a) = \sum_{i<j} c_i(a), \qquad s_j = \sup_{a\in\mathcal{A}} \|S_j(a)\|, \qquad z_j(a) = P_j(a + S_j(a)).$$

If $b_0 = 0$ and

$$b_{j+1} = (1 + \ell_j)b_j + \ell_j s_j, \tag{48}$$

then $\sup_a \|y_j(a) - z_j(a)\| \le \kappa_j b_j$. Indeed, with $e_j = u_j - a - S_j(a)$,

$$e_{j+1} = e_j + c_j(a + S_j(a) + e_j) - c_j(a),$$

whose norm is bounded by $(1 + \ell_j)\|e_j\| + \ell_j s_j$. Induction and the Lipschitz bound of P_j prove the assertion. The signed sum S_j retains cancellations before the remainder is bounded.

For perturbed updates $\widehat{U}_j = U_j + h_j\xi_j$ with $\|\xi_j\| \le \nu_j$, add $\lambda_{j+1}h_j\nu_j$ to the right-hand side of Equation (48). At each actual input, both $U_j(\widehat{y}_j)$ and $\widehat{U}_j(\widehat{y}_j)$ must belong to a valid domain for the same inverse-Lipschitz bound. Clock or floating-point errors not included in ν_j need their own bounds. Linear interpolation of the actual and predicted *node values* extends the maximum nodal bound to the entire output path by convexity of the norm. Interpolating P and S separately and then composing them would be a different output rule. Useful nonlinear certificates still require computable domain, conditioning and defect bounds.

D.2 ORDERED MATRIX PRODUCTS AND THE ERROR INTERVAL

For the matrix paths in the main proposition, define

$$C_j = P_{j+1}^{-1}A_jP_j - I, \quad S_j = \sum_{i<j} C_i, \quad \sigma_j = \sum_{i<j} \|C_i\|_2.$$

The exact arithmetic node maps and their first-order predictions are

$$Y_j = P_j(I + C_{j-1}) \cdots (I + C_0), \qquad Z_j = P_j(I + S_j).$$

No commutativity is assumed. In the ordered product, the degree-zero and degree-one terms are I and S_j. Submultiplicativity bounds the remaining terms by the elementary symmetric sums of $\|C_i\|_2$, hence

$$\left\| \prod_{i=j-1}^{0} (I + C_i) - I - S_j \right\|_2 \le \prod_{i<j}(1 + \|C_i\|_2) - 1 - \sigma_j \le e^{\sigma_j} - 1 - \sigma_j.$$

Multiplication by $R\|P_j\|_2$ bounds the output remainder. Taking the maximum over nodes and interpolating gives the r in the proposition. The additional compiler path bound yields $\sup_{t,a} \|\widehat{y}(t, a) - \bar{Z}(t)a\| \le r + b_{\mathrm{comp}}$. Applying the triangle and reverse-triangle inequalities to any reference $B(t)a$ proves Equation (18). The expansion and exponential remainder give a prediction from prefix sums of defects and a bound for the omitted ordered products.

For $B = P_N$ this gives execution error; for $B = R_L$ it gives residual/shared pair error; for $B = D$ it gives error to the common target. If only lower and upper enclosures $q_B^- \le q_B \le q_B^+$ are computable, use $\max(0, q_B^- - \eta)$ and $q_B^+ + \eta$ as the interval endpoints. This matters for continuous-time comparisons to the smooth target.

D.3 WHICH UPDATE DIFFERENCES MATTER?

Let $V_j = P'_+(t_j)$ for a piecewise-affine matrix path and define

$$\Delta_j = P_{j+1} - P_j - h_j V_j.$$

If J_τ is the slope jump at an internal knot τ, piecewise integration gives

$$\Delta_j = \sum_{t_j < \tau < t_{j+1}} (t_{j+1} - \tau) J_\tau. \tag{49}$$

With the exact inverse velocity, $C_j = -P_{j+1}^{-1}\Delta_j$. Thus a knot's location, jump size and direction all matter; knot count or a step-size histogram alone loses relevant information. For the implemented polynomial $Q_4(P) = \sum_{r=0}^{4}(I-P)^r$, the identity $Q_4(P)P = I - (I-P)^5$ gives instead

$$C_j = -P_{j+1}^{-1}\big[\Delta_j + h_j V_j (I - P_j)^5\big].$$

This already includes inverse-truncation error. When exact-inverse Euler steps stay inside path segments, $\Delta_j = 0$ and they execute the encoded path exactly. Steps crossing a bend can also introduce interpolation error between otherwise exact nodes. Both effects are included by comparing the complete predicted polygon with the reference.

This connects to the main theorem's mechanism: its original grid resolves the encoded segment boundaries. A proposed clock can be evaluated by its weighted knot defects and the resulting complete output path.

D.4 COMPILER DOMAIN AND FINITE CHECKS

The retained check uses all 96 configurations of the six $K = 8$ targets in Appendix C, with $N = 128,512$ and the existing legal clocks. For $\alpha = 2^{-18}$, the encoded paths satisfy

$$\|P(t) - I\|_2 \le \rho = \frac{2\alpha}{1-\alpha} < 7.630 \times 10^{-6}.$$

Convex interpolation preserves this bound, so it guarantees invertibility throughout each segment as well as at anchors. The original product-circuit precision is $\xi_{\mathrm{prod}} = 16 \cdot 4^{-18} = 2^{-32}$. In the valid state buffer and on the original $1/512$ clock lattice, the real-arithmetic field budget is $\nu = 2048K\alpha\xi_{\mathrm{prod}} = 2^{-36}$ and a path budget is $b_{\mathrm{comp}} = 2\nu = 4096K\alpha\xi_{\mathrm{prod}}$. To close the domain condition, put $\lambda = 7K\alpha$ and $R = \sqrt{2}/4$. The ideal Q4 field is homogeneous linear and satisfies the corresponding Lipschitz bound. For any positive steps totaling one,

$$\|\widehat{y}_{j+1}\| \le (1 + h_j\lambda)\|\widehat{y}_j\| + h_j\nu, \qquad \max_j \|\widehat{y}_j\| \le e^{\lambda}(R + \nu) < .354 < .5.$$

Propagation of neural/arithmetic discrepancies gives $e^{\lambda}\nu < 2\nu$. These bounds apply on the valid compiler domain and original clock lattice. The numerical comparisons use float64 and a physical tolerance of 2×10^{-13}.

Predictions use the target descriptions, path resolution, clock and fixed compiler contract, then are compared with the saved trajectories. For linear arithmetic maps, the maximum over the initial box is attained at a corner, and polygon comparisons need only the union of time knots. The neural diagnostics use four corners and 32 Sobol initial queries. Comparisons to D use the existing 65536-interval time grid with a Lipschitz remainder to enclose the continuous-time maximum.

All 96 remainder checks pass. The maximum defect sum is approximately 2.289×10^{-5}, making the second-order remainder small. Across two residual reference depths and three thresholds, the 576 related total-fidelity decisions comprise 388 passes and 188 rejections, with no undetermined cases, and agree with the saved labels. Total fidelity and the separate coding allocation are evaluated independently. The maximum execution-error difference from the predicted center is approximately $3.414 \times 10^{-7}\alpha$; the maximum interval width is $6.382 \times 10^{-5}\alpha$. The small defects in these near-identity examples explain the narrow intervals.

An amplitude stress for the interval. To expose the other regime, let $P_\alpha(j/8) = I + \alpha \left(\begin{smallmatrix} 0 & u_j \\ v_j & 0 \end{smallmatrix} \right)$ with $u = (0, 1, 1, 0, -1, -1, 0, 1, 0)$ and $v = (0, 0, 1, 1, 0, -1, -1, 0, 0)$, and interpolate linearly between the nine knots. For $\alpha < 1$, $\|P_\alpha(t) - I\|_2 \le \alpha$, so every inverse is defined and the condition number is at most $(1+\alpha)/(1-\alpha)$. We execute the ideal field $P'_\alpha(t)P_\alpha(t)^{-1}y$ from the $[-1/4, 1/4]^2$ input box, using the full eight-call anchor schedule, two four-call schedules with the same long/short step multiset, four uniform calls, or two uniform calls. The reference is $B = P_\alpha$ and $b_{\mathrm{comp}} = 0$. For this piecewise-linear matrix experiment, all full-path errors are attained at a path knot and an input-box corner.

Each row below aggregates the interval decisions for five fixed schedules at the three absolute tolerances .02, .05, .10 (15 decisions per amplitude).

Table 2: **Amplitude stress of the conditional error interval.** Counts are pass, reject, and undetermined decisions across the 15 schedule–tolerance pairs at each amplitude; the condition column is the bound $(1 + \alpha)/(1 - \alpha)$.

α	condition bound	pass	reject	undetermined
.02	1.041	12	2	1
.05	1.105	7	3	5
.10	1.222	3	3	9
.20	1.500	3	0	12
.35	2.077	3	0	12
.50	3.000	3	0	12
.65	4.714	3	0	12

All 35 direct polygon errors lie in the predicted intervals. As amplitude, conditioning and defect sums grow together, the remainder widens: for $\alpha \ge .35$, only the three exactly aligned cases are certified and all twelve other decisions are undetermined. This arithmetic stress tests the scope of the conditional interval; it uses no finite neural compiler. The protocol, complete 105-decision record, corrected arithmetic source and dense-time verifier are included in the anonymous supplementary materials.

A first-order sum can miss interactions. For general updates with $P = I$, two defects $C_0 = aI$ and $C_1 = -aI$ sum to zero but have product $(1 - a^2)I$. Conversely, multiples of a matrix H with $H^2 = 0$ have no higher-order cross terms. The remainder and compiler conditions distinguish these cases when evaluating a candidate schedule.

E Residual refinement, state gradients and parameter learning

The following stability bounds and examples develop the depth interpretation in Section 5. They distinguish propagation along a fixed path, fidelity of its derivatives, and parameter updates for separately parameterized blocks and a time-conditioned network.

E.1 A cumulative increment budget controls propagation

Suppose $U_k(x) = x + r_k(x)$ is differentiable on the relevant domains and $\|Dr_k(x)\|_2 \le \rho_k \le q < 1$. For ReLU maps, Jacobian statements apply at points where the relevant derivatives exist; distance bounds can instead use global Lipschitz constants. This assumes identity shortcuts without an additional nonidentity gate after addition. Downsampling or post-addition activations require their own analysis.

For $J_{j:i}$, the state Jacobian from layer i to layer j, put $S_{i:j} = \sum_{k=i}^{j-1} \rho_k$. Singular-value product bounds give

$$\prod_{k=i}^{j-1}(1 - \rho_k) \le \sigma_{\min}(J_{j:i}) \le \sigma_{\max}(J_{j:i}) \le \prod_{k=i}^{j-1}(1 + \rho_k), \tag{50}$$

$$e^{-S_{i:j}/(1-q)} \le \sigma_{\min}(J_{j:i}) \le \sigma_{\max}(J_{j:i}) \le e^{S_{i:j}}. \tag{51}$$

The second line follows from the scalar logarithm bounds. An endpoint loss has state adjoint $p_i = J_{j:i}^{\mathsf{T}} p_j$, with the same norm bounds; a nonzero endpoint adjoint is needed for a nontrivial lower bound. Parameter gradients also involve the derivative of each update with respect to its parameters.

For $r_k = h_k f_k$, $\sum h_k = T$ and $\mathrm{Lip}_x f_k \le K$, $S \le KT$ provided $h_{\max} K \le q$. The bound need not grow with layer count, although e^{KT} can itself be large. Our normalized teacher class satisfies this sufficient condition with $T = K = 1$ and $q = 1/2$ for $L \ge 2$. A shared compiler needs the corresponding derivative control to inherit the same bound. For example, $r_k(x) = x$ gives $J = 2^L I$. Increasing the time horizon or changing the target also changes the comparison.

For one fixed field, assume additionally $\|f\| \le B$ and time-Lipschitz constant A, with exact and discrete trajectories in a common valid domain. The uniform Euler local defect against the continuous solution is at most $(A + KB)h^2/2$. Discrete Gronwall gives

$$\max_k \|x_k - P_{t_k}(a)\| \le \frac{C_E}{L}, \qquad C_E = \frac{T(A + KB)(e^{KT} - 1)}{2K}.$$

For $K = 0$ take $C_E = AT^2/2$. Thus $L \ge \max\{KT/q, C_E/\varepsilon\}$ is a sufficient choice for both the stated propagation condition and forward accuracy. This continuous-solution comparison illustrates refinement of a fixed field.

E.2 Exact subdivision leaves intrinsic propagation unchanged

For a C^1 diffeomorphic path, the chain rule applied to $U_k = P_{t_{k+1}} \circ P_{t_k}^{-1}$ gives

$$J_{j:i} = DP_{t_j}(a)\,[DP_{t_i}(a)]^{-1}. \tag{52}$$

If $\sup_t \|DP_t\| \le K_P$ and $\sup_t \|(DP_t)^{-1}\| \le \Lambda_P$, its segment singular values lie between $(K_P\Lambda_P)^{-1}$ and $K_P\Lambda_P$. The same identity holds for any exact implementation of the same path, regardless of how its maps are stored.

Refinement cannot remove intrinsic ill-conditioning. For example, $P_t = \mathrm{diag}(e^{a\sin(2\pi t)}, e^{-a\sin(2\pi t)})$ returns to the identity while attaining maximum singular value $e^{|a|}$ in between. Conversely, a planar rotation field $f(x) = \omega Q x$, with $Q^{\mathsf{T}} = -Q$ and $Q^2 = -I$, has an exact norm-preserving path. Its L uniform Euler steps have equal singular values

$$\left(1 + \frac{\omega^2}{L^2}\right)^{L/2}, \qquad \log \sigma_{\max} \le \frac{\omega^2}{2L}.$$

Finer execution removes artificial amplification. The condition number is one throughout, so checking it alone would miss that amplification.

E.3 Derivative defects are needed for gradient fidelity

Use the nonlinear coordinates of Appendix D, with $\Theta_k = \mathrm{id} + c_k$. Suppose $\|Dc_k\| \le \gamma_k \le q < 1$, put $G_{i:j} = \sum_{k=i}^{j-1} \gamma_k$, and bound $\|DP_k\| \le \kappa_k$, $\|(DP_k)^{-1}\| \le \lambda_k$ on all queried domains. Differentiating the coordinate recursion yields

$$D_a y_j = DP_j(u_j)(I + Dc_{j-1}(u_{j-1})) \cdots (I + Dc_0(u_0)).$$

For propagation from node i to node j this product also includes $[DP_i(u_i)]^{-1}$ on the right, and consequently

$$\sigma_{\max}(J_{j:i}^U) \le \kappa_j \lambda_i e^{G_{i:j}}, \qquad \sigma_{\min}(J_{j:i}^U) \ge \frac{e^{-G_{i:j}/(1-q)}}{\lambda_j \kappa_i}.$$

If $\|u_j - a\| \le e_j$ and DP_j has Lipschitz constant H_j on the required connecting region, expansion of the product gives

$$\|D_a y_j - DP_j(a)\| \le \kappa_j(e^{G_{0:j}} - 1) + H_j e_j.$$

Small state defects control e_j; derivative defects control $G_{0:j}$. In the affine arithmetic case $c_k(u) = C_k u$, the matrix budget also controls derivatives. For the nonlinear ReLU graph, gradient fidelity additionally requires bounds on its spatial derivative.

E.4 Uniform path values need not determine input Jacobians

Here is a counterexample inside the main theorem's $s = 1$ class. Define the odd, compactly supported piecewise-linear function ψ by $\psi(x) = x$ for $0 \le x \le 1/2$, $\psi(x) = 1 - x$ for $1/2 \le x \le 1$, and $\psi(x) = 0$ for $x \ge 1$. It has Lipschitz constant one, $\|\psi\|_\infty = 1/2$, and $\psi'(0) = 1$, and has a finite ReLU representation. Fix $c = 1/2$ and set

$$f_n(x,t) = \frac{c}{n}\psi(nx), \qquad f_0(x,t) = 0.$$

For $n \ge 1$, these fields share support inside $[-1,1]$, have spatial Lipschitz constant at most c, and are constant in time. Every Euler increment has magnitude at most $hc/(2n)$, hence uniformly over initial states and all prefixes,

$$\max_{k \le L} \sup_{a \in [-1,1]} |X_k^{f_n}(a) - a| \le \frac{c}{2n} \longrightarrow 0.$$

The trajectory from zero stays at zero, and all relevant derivatives there exist. Nevertheless,

$$\partial_a X_L^{f_n}(0) = \left(1 + \frac{c}{L}\right)^L \ge 1 + c, \qquad \partial_a X_L^{f_0}(0) = 1.$$

The Jacobian gap persists even on increasingly resolved common grids. Thus uniform prefix-value convergence alone permits a fixed gradient discrepancy.

For smoother paths, if two path maps also have uniformly Lipschitz derivatives, put $E = P - Q$, $\|E\|_\infty \le \varepsilon$ and $\mathrm{Lip}(DE) \le H$. Taylor's formula along a valid line segment of length t gives $\|DE(x)v\| \le 2\varepsilon/t + Ht/2$ for unit v. Choosing $t = 2\sqrt{\varepsilon/H}$ yields $2\sqrt{H\varepsilon}$ when that length fits inside the domain. Otherwise minimize over valid lengths. The bound uses derivative regularity of both the target and its approximation.

E.5 State stability and the parameter metric

For endpoint loss ℓ, untied Euler parameters have gradients

$$\nabla_{\theta_k} \ell = h_k [D_\theta f_{\theta_k}(x_k)]^\mathsf{T} p_{k+1}, \qquad p_k = (I + h_k D_x f_{\theta_k})^\mathsf{T} p_{k+1}.$$

For shared parameters these contributions add across calls: $\nabla_\phi \ell = \sum_k h_k [D_\phi g_\phi(x_k, t_k)]^\mathsf{T} p_{k+1}$. They may reinforce or cancel. Stable state adjoints alone do not determine optimization speed.

An exact quadratic example isolates the parameter metric. Let $h = 1/L$, $b(t) = 1 + t$, $f_k(x) = b(t_k)\theta_k$, and $g_\phi(x,t) = b(t)\phi$. At $\theta_k = \phi$, the two implementations follow the same path, and every state Jacobian is one. With $x_0 = 0$ and loss $\frac{1}{2}(x_L - y)^2$, define $a_k = hb(t_k)$ and $\bar{b} = \sum_k a_k$. The squared output-parameter Jacobian norms are

$$K_{\text{untied}} = \sum_k a_k^2 = \Theta(L^{-1}), \qquad K_{\text{shared}} = \bar{b}^2 = \Theta(1).$$

Under Euclidean gradient descent, the output error obeys exactly $e^+ = (1 - \eta K)e$. At a fixed stable learning rate, attaining a fixed relative error thus takes $\Theta(L)$ iterations for the first coordinates and $O(1)$ for the second. Under the discrete time-L^2 metric $\|\delta\theta\|_h^2 = \sum_k h|\delta\theta_k|^2$, the inverse metric multiplies each update by $1/h$, and the effective untied kernel becomes $\sum_k hb(t_k)^2 = \Theta(1)$. This removes the depth factor in this example. The example uses endpoint supervision and constant translation fields to isolate parameter scaling. Depth-normalized parameter dynamics have direct precedent in Marion et al. (2024).

More generally, for squared loss and parameter metric M, a uniform bound $J_\theta M^{-1} J_\theta^\mathsf{T} \succeq \mu I$ along gradient flow gives $\ell(t) \le e^{-2\mu t}\ell(0)$. This follows by differentiating the loss. Optimization therefore also depends on the parameter-feature bound; discrete updates additionally require step-size and smoothness conditions. Near-identity training results such as Hardt & Ma (2017) also require their stated optimization conditions.

Finally, a linear residual block $I + hA_k$ can be written as a plain block W_k. With matched initial functions and $W_k = I + hA_k$, $\nabla_A \ell = h \nabla_W \ell$. Vanilla SGD without weight decay produces identical function updates when $\eta_W = h^2 \eta_A$. The equality follows from this linear parameterization and matched learning-rate scaling.

F Related work and positioning

Residual networks and differential equations have long been connected through their local updates (He et al., 2016; Chen et al., 2018; E et al., 2022). We fix an Euler residual teacher and compare its endpoint, every prefix, and sampled field as three distinct reconstruction tasks. This makes the accuracy target explicit when residual layers are represented by one time-conditioned evaluator.

Conditioned networks and hypernetworks encode families of functions; their approximation and modularity properties have been studied by Galanti & Wolf (2020). Flow-map formulations emphasize intermediate-time transformations and composition (Boffi et al., 2025). Hierarchical neural time-steppers compose maps at several scales and interpolate intermediate states (Liu et al., 2022); anchor flow maps factor the full two-time map through a common time to reduce sampled storage (Stelter et al., 2026). Time-conditioned neural ODEs also admit uniform trajectory approximation in broad classes (Li et al., 2026). For the Euler-prefix task, our current-state/time evaluator executes one shared code, with clock, inverse queries, copied coefficients and finite arithmetic charged.

The proof draws on established approximation and inversion tools. Finite-element hat functions have exact ReLU realizations (He et al., 2018); invertible residual maps can be inverted by contraction (Behrmann et al., 2019); and bounded multiplication has logarithmic-size ReLU approximations (Yarotsky, 2017). These tools support the geometry-preserving path code and its finite neural compiler.

Entropy and characteristic approximation. Mixed-smoothness theory quantifies the information in coupled coordinate variation. For $(d, s) = (1, 2)$, Theorem 1.8 of Temlyakov (2016), together with the Euler-prefix embedding below, gives the $\widetilde{O}(\varepsilon^{-3/4})$ unrestricted information upper bound. The constructive result in Theorem 1 attains this power with coherent invertible path maps and a fully counted shared evaluator; its lower bound uses a packing within the admissible teacher class. Classical entropy calculations also exhibit half-power laws in one-dimensional convex classes (Guntuboyina & Sen, 2013).

Theorem 4.2 of Dahmen (2025) gives a uniform DNN approximation of transport characteristics under a compositional approximation-class premise. Its approximated map takes initial position and time as inputs. Executing a path from current state and time also requires invertible approximate maps with controlled inverse Lipschitz constants, together with a circuit that evaluates those inverses within the storage and error budgets. Appendix A provides this geometry and charges the query circuit, coefficient copies and precision at the path-information power. The same evaluator supports the aligned execution in Corollary 2.

Regularity-based studies also connect residual depth with statistical complexity (Marion, 2023). Our quantity is the target-dependent description length of deterministic prefix maps. For the interpretation of depth refinement, Hardt & Ma (2017) analyzes near-identity linear residual parameterizations, and Marion et al. (2024) studies depth-scaled residual training dynamics. The state-propagation and parameter-metric examples in Appendix E place the forward path results alongside those optimization analyses.

F.1 The scalar path upper bound from mixed-smoothness entropy

In the two-variable notation of Temlyakov (2016), let $B_{p,q}$ be the orthogonal dyadic Fourier projections and $F_\ell = \sum_{p+q=\ell} B_{p,q} F$. The periodic class $\mathbf{W}_2^{a,b}$ is defined by $\|F_\ell\|_2 \le 2^{-a\ell} \max(1, \ell)^b$. The two variables here are initial state and time; the state dimension remains one.

Lemma 2 (Uniform extension of scalar Euler prefixes). *For $f \in \mathcal{F}_{1,2}$ and $h = 1/L$, the displacements $U_k(a) = \Phi_k^f(a) - a$ admit an extension $H_L(a, t)$ satisfying $H_L(a, kh) = U_k(a)$ and, in the weak-derivative sense,*

$$\|H_L\|_\infty + \|\partial_t H_L\|_\infty + \|\partial_a^2 H_L\|_\infty + \|\partial_t \partial_a^2 H_L\|_\infty + \|\partial_t^2 H_L\|_\infty \le C. \tag{53}$$

After extension to a fixed larger rectangle, cutoff and rescaling to a fixed torus, these functions belong to $C\mathbf{W}_2^{4/3,0}$. All constants are independent of f and L.

Proof. Set $v_k = f(\Phi_k^f, kh)$ for $0 \le k \le L$, using the value at time one only for interpolation. The Euler recurrence and the class assumptions give

$$U_{k+1} - U_k = hv_k, \qquad \|v_{k+1} - v_k\|_\infty \le 2h, \qquad \|U_k\|_{W^{2,\infty}} + \|v_k\|_{W^{2,\infty}} \le C. \tag{54}$$

For the last estimate, the Jacobian recurrence bounds $\|D\Phi_k^f\|_\infty$ by e, and its spatial Lipschitz recurrence bounds $\mathrm{Lip}(D\Phi_k^f)$ uniformly over the unit horizon. Composition with f_{kh} then bounds v_k in $C^{1,1}$.

On $t = kh + uh$, $0 \le u \le 1$, use the cubic interpolant

$$H_L(a,t) = U_k(a) + h\big[uv_k(a) + c(u)(v_{k+1}(a) - v_k(a))\big], \qquad c(u) = u^2(u-1). \tag{55}$$

It matches U_k, U_{k+1} and has endpoint time derivatives v_k, v_{k+1}, so $\partial_t H_L$ joins continuously between cells. Within each cell,

$$\partial_t H_L = v_k + c'(u)(v_{k+1} - v_k), \qquad \partial_t^2 H_L = h^{-1}c''(u)(v_{k+1} - v_k).$$

Equation (54) proves (53); the mixed derivative uses only the uniform spatial bounds of the two slopes. No time regularity of Df_t is needed. Continue H_L affinely in time outside $[0,1]$ using endpoint slopes v_0, v_L, and multiply by a fixed smooth cutoff equal to one on $[0,1]$ and supported in $(-1,2)$. Values and first time derivatives match at the endpoints. Also, U_k and v_k vanish outside $[-1,1]$. Periodization on $(-2,2)\times(-2,3)$ and fixed coordinate rescaling therefore preserve the bounds and all original-node values.

Write F for the resulting periodic function. For $p, q \ge 1$, the mixed and temporal derivative bounds give

$$\|B_{p,q}F\|_2 \le C\min\{2^{-2p-q}, 2^{-2q}\}, \qquad \max\{2p+q, 2q\} \ge \tfrac{4}{3}(p+q).$$

On each shell, Parseval's identity with the two derivative multipliers yields

$$\left\| \sum_{\substack{p+q=\ell \\ p,q\ge 1}} B_{p,q}F \right\|_2^2 \le C2^{-8\ell/3}\big(\|\partial_a^2\partial_t F\|_2^2 + \|\partial_t^2 F\|_2^2\big).$$

The $q = 0$ and $p = 0$ rectangles have L_2 norm at most $C2^{-2\ell}$ by the pure spatial and temporal second derivatives, respectively. The constant mode is bounded as well. Hence $\|F_\ell\|_2 \le C2^{-4\ell/3}$, which is the claimed class membership. □

Theorem 1.8 of Temlyakov (2016) gives, for this two-variable class,

$$e_n \lesssim n^{-4/3}(\log(2+n))^{11/6}, \qquad H_\varepsilon \lesssim \varepsilon^{-3/4}(\log(2/\varepsilon))^{11/8}. \tag{56}$$

Here e_n uses 2^n covering balls and H_ε is the base-two logarithm of the covering number. The extension family consists of continuous functions agreeing exactly with the original nodes. We transfer its cover to legal sampled evaluators as follows.

To obtain a legal decoder, take a cover of radius $\varepsilon/4$ and choose one teacher representative from every occupied ball. The target and representative are continuous, so their mutual L_∞ bound holds pointwise and gives $\varepsilon/2$ error at every original node. A finite decoder may store, for each representative, rational piecewise-affine approximations to its L frozen-time fields with uniform error ζ, extended by zero outside the fixed buffer. Euler stability bounds the additional path error by $((1+h)^L - 1)\zeta \le (e-1)\zeta$. Choosing $\zeta \le \varepsilon/[2(e-1)]$ completes an ε-accurate finite codebook, fixed before the target is known. Its entry index has the bit length in (56), proving $B^{\mathcal{F}_{1,2}}_{\mathrm{path,all}}(\varepsilon, L) = \widetilde{O}(\varepsilon^{-3/4})$. This information argument allows a large public decoder. The full sparse serialization and current-state execution bounds are supplied by the coherent construction in Appendix A.

G Aligned execution and the resolution of linear output

We prove Corollary 2 using the same path code and finite evaluator as Appendix A. The original theorem uses L Euler calls. Here we change the execution and readout rule to (9); the teacher and all original prefix constraints remain fixed. Constants below may depend on the fixed dimension.

G.1 One call per anchor interval

Choose a dyadic $\delta \asymp \sqrt{\varepsilon}$ with $h \mid \delta \mid \Delta_0$ and put $N = 1/\delta \le L$. Recall $Q_0 = \mathrm{id}$, the uniform Lipschitz bounds for Q_j and Q_j^{-1}, and the interpolant D_t from Lemma 1 and (23). At an anchor $t_j = j\delta$, the right-hand branch has $\lambda = 0$, so (32) gives, for every admissible current state y,

$$g(y, t_j) = \delta^{-1} R_j(y), \qquad y + \delta g(y, t_j) = Q_{j+1} Q_j^{-1}(y). \tag{57}$$

Thus the ideal coarse Euler execution visits $Q_j(a)$ at every anchor. Each step spans exactly one affine segment of D_t, so the ideal Euler update follows that segment exactly.

Let $\widehat{g}$ be the compiled evaluator with uniform velocity error at most ν on the relevant buffer at legal sampled times, including coefficient rounding and evaluator arithmetic. Reserve τ for the outer multiplication and addition error:

$$\widehat{y}_{j+1} = \widehat{y}_j + \delta \widehat{g}(\widehat{y}_j, t_j) + e_j, \quad \|e_j\| \le \tau, \qquad \widehat{y}_0 = a. \tag{58}$$

For $a_j = Q_j^{-1}(\widehat{y}_j)$, uniform inverse conditioning gives

$$\|a_{j+1} - a_j\| \le C\big(\delta\nu + \tau\big), \tag{59}$$

$$\max_{j \le N} \|\widehat{y}_j - Q_j(a)\| \le C\big(\nu + N\tau\big). \tag{60}$$

The first inequality compares Q_{j+1}^{-1} at the ideal image $Q_{j+1}Q_j^{-1}(\widehat{y}_j)$ and at the actual next state. The second sums the first using $N\delta = 1$ and applies the uniform Lipschitz bound for Q_j. In particular, anchor approximation errors are not added independently at every call: the same fixed maps cancel in the coordinate update.

Linearly interpolate the actual nodes to obtain $\bar{y}$. Convexity of the norm bounds its error to $D_t(a)$ by the maximum nodal error. With (25),

$$\max_{k \le L} \sup_a \|\bar{y}(kh, a) - \Phi_k^f(a)\| \le C\big(\delta^2 + m\eta + \nu + N\tau\big). \tag{61}$$

Choose $\delta \le c\sqrt{\varepsilon}$, $\eta \le c\varepsilon/m$, $\nu \le c\varepsilon$, and $\tau \le c\varepsilon/L$. For a sufficiently small fixed c and a sufficiently large resolved-grid constant, a dyadic δ exists and $N \le L$. Equation (61) is then at most ε, with a fixed fraction reserved for readout rounding if outputs are materialized in finite precision.

Legal clocks, domains, and precision. Every $j\delta$ is an original-grid time. The clock selectors of Appendix A choose the right-hand branch exactly there, including at internal anchors, so no guarantee between legal clock samples is invoked. The terminal time is used for output, not for an extra call.

Use the fixed buffer and local-map extensions of the original realization, with positive margin around the teacher tube and its coherent anchor images. If necessary enlarge the fixed outer and inverse-query buffers by a dimension-dependent constant before compiling; the local approximation bounds retain their rates on such fixed cubes. We use the original compiler's containment guarantee for contraction fixed points and its projected inverse-query bound (37), with query buffers containing those fixed points. For the outer execution, suppose nodes through j are valid. Before applying the next inverse, (57) gives

$$\widehat{y}_{j+1} = Q_{j+1}(a_j) + \xi_j, \qquad \|\xi_j\| \le \delta\nu + \tau,$$

and hence

$$\|\widehat{y}_{j+1} - Q_{j+1}(a)\| \le C\|a_j - a\| + \delta\nu + \tau.$$

The accumulated coordinate bound through j puts this below the fixed buffer margin for small ε. The next state is therefore in the valid domain before Q_{j+1}^{-1} is invoked. Induction now closes (59)–(60) and all outer calls remain covered by the original compiler guarantees.

Use (38)–(39) with $\nu = c\varepsilon$. Coefficient and operation precision from (40) already allows an $O(\varepsilon/L)$ outer-update error. Applying the larger dyadic multiplier $\delta \le 1$ on the same bounded states and velocities needs at most an additional $O(\log L)$ precision allowance relative to the original step; this is absorbed by the existing bit bound. Thus we can keep $\tau \le c\varepsilon/L$, with total $N\tau = O(\varepsilon)$. The dyadic step δ introduces no larger clock bit scale. All inverse iterations, coefficient copies and $O(N)$ selectors were charged before; therefore the full sparse serialization retains $\varepsilon^{-\beta(d,s)}\,\mathrm{polylog}(L/\varepsilon)$. This uses the finite-network compiler of Appendix A.

Readout resources. An interval's interpolated outputs can be emitted once its right node is available, using two adjacent states and $O(d)$ scalar registers, or $O(db)$ bits at b-bit precision. Storing every coarse node for later random access instead costs $O(Md)$ registers. Materializing all $L+1$ prefix outputs takes $O(Ld)$ interpolation operations; sufficient precision for an $O(\varepsilon)$ uniform readout budget fits within the declared bit scale. These costs are separate from the $M = N$ neural calls; the evaluator continues to receive current state and time.

G.2 A MATCHING OBSTRUCTION FOR LINEAR OUTPUT

Choose a fixed $C^{1,1}$ cutoff χ, supported in $[-1,1]^d$ and identically one on a neighborhood of the origin. For a sufficiently small constant $c_d > 0$, the field

$$f(x,t) = c_d t \chi(x) e_1 \tag{62}$$

belongs to both $\mathcal{F}_{d,1}$ and $\mathcal{F}_{d,2}$. The choice of c_d controls the field magnitude, its spatial and temporal Lipschitz constants, and the Lipschitz constant of its Jacobian. It also keeps the Euler trajectory from $a = 0$ inside the plateau throughout the unit horizon. Its only nonzero coordinate is

$$q(kh) = \frac{c_d}{2} h^2 k(k-1), \qquad q(t) = \frac{c_d}{2}(t^2 - ht). \tag{63}$$

Consider any continuous polygon with M segments whose knots lie on the original grid and include zero and one. Its node values and knots may be chosen arbitrarily for this target. A longest segment $[u,v]$ has length $H \geq 1/M$. If $M \leq L/2$, then $H \geq 2h$. Choose a grid point w nearest the midpoint and set $\theta = (w-u)/H$. This is an interior original prefix, and $|w - (u+v)/2| \leq h/2$. The first coordinate p of the polygon is affine on $[u,v]$, whereas

$$(1-\theta)q(u) + \theta q(v) - q(w) = \frac{c_d}{2}(w-u)(v-w). \tag{64}$$

Uniform prefix error at most ε bounds the left-hand side by 2ε, since the corresponding affine combination of p is zero. It follows that

$$\varepsilon \geq \frac{c_d}{4}(w-u)(v-w) \geq \frac{c_d}{16}(H^2 - h^2) \geq \frac{3c_d}{64M^2}. \tag{65}$$

If instead $M > L/2$, the resolved-grid assumption $L \geq C_d \varepsilon^{-1/2}$ directly implies the desired lower bound. Thus in either case $M \geq c'_d \varepsilon^{-1/2}$.

One initial state suffices to obstruct the all-input guarantee. The argument even allows arbitrary, target-dependent node values, so it applies to any network execution with this readout rule. Combined with the constructive upper bound, it gives the worst-case $\Theta(\varepsilon^{-1/2})$ segment/call rate under the aligned linear-output contract.