ALIGNING AND FOLDING: WHAT NETWORKS DO BEFORE THEY LEARN

Anonymous authors
Paper under double-blind review

ABSTRACT

Architectures are usually organized by name, which hides an elementary question: what are the reusable pieces from which they are assembled? We answer that a network is an alternation of two processes. An *align* exposes patterns on arbitrary supports in comparable coordinates; a *fold* combines responses and may identify distinctions. Convolutional networks, transformers, and state-space mixers are subfamilies of this single class: residual blocks, a pre-norm transformer block, and the mixer of an official Mamba-130M agree with their translated programs to numerical precision on the order of 10^{-10} in forward values and per-parameter gradients, and covering the state-space mechanism costs one additional align primitive. The class is built to answer two questions before training: whether a given architecture can realize a task within a declared budget, and which architecture in a predeclared family should be chosen. Two empirical principles make this precise. *Information is pattern*: a task is generated by finitely many admissible patterns of a label-independent grammar. *Failure is cost*: once an architecture can jointly expose those patterns, an exact realization of the task is reachable from some admissible initialization whenever the declared budgets suffice. Each has a stated falsifier. For **diagnosis**, a representation admits an unrestricted deterministic readout precisely when its fibers refine the label partition, and cross-label collisions are irreversible downstream. We define a capacity that counts, as a formal upper bound, the ways an architecture can transform the basis of the patterns it jointly exposes. Predictions made before training hold on controlled tasks, including a certified impossibility and a targeted repair. For **selection**, the same capacity and cost are computed for every member of a predeclared family without training candidates, returning a finite cost-Pareto set; the framework finds a vision transformer with 25% fewer parameters than ViT-Lite-7/4 that matches ViT-Lite-6/4, and, among inherited-weight subnets of AutoFormer's ImageNet search space, one with fewer parameters matching the reported accuracy of AutoFormer-T.

1 INTRODUCTION

Choosing a convolutional network, a transformer, or a state-space model—and finding a cheaper variant—often requires costly training and evaluation. We ask what can be decided beforehand: can an architecture support a task within a declared learning budget, and which architecture should be constructed within a declared resource budget? Our approach examines which input patterns an architecture can expose jointly, and at what cost.

We start from two empirical principles. *Information is pattern* (EP1): a task is determined by finitely many patterns of the input, and a pattern may draw on any set of input sites, however distant. *Failure is cost* (EP2): once an architecture exposes those patterns jointly, training can reach an exact realization whenever the declared learning resources suffice. The first fixes what a task needs; the second connects exposing it to learning. Both are hypotheses, each with a stated falsifier.

We describe architectures using two recurring roles. An *align* exposes pattern instances in comparable coordinates; a *fold* combines responses and may identify distinctions among its inputs. Their directed composition defines an open architecture class. Convolutions, attention, residual blocks, and state-space mixers are particular programs in this class, with their usual implementation costs. A new architecture combines existing primitives; a new primitive extends the dictionary.

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.

Within this class, a ledger read from the architecture alone counts the ways a construction can transform the basis of the patterns it jointly exposes. Each fold contributes a few local choices, a selection among its input maps or a joint merge of its branches; treating them as independent gives a formal count C and its logarithm L, which complement the familiar implementation costs. What is counted are joint constructions, not patterns obtainable one at a time.

These pieces answer the two questions. *Can an architecture support a task?* By EP2, it can once it jointly exposes the task's patterns and the declared budgets suffice; when those patterns are known, an explicit realization certifies this, and a collision that holds for every weight rules it out. *Which architecture should be built?* When the patterns are unknown, L calibrated on a reference screens a declared family, and the cheapest candidates that pass are chosen. Both answers come from the construction alone: no candidate is instantiated, run forward, or trained before it is chosen; training only tests the choice. The experiments test each link: exact certificates on enumerable tasks and natural images, an attempt to falsify EP2, fourteen residual blocks in two matched-cost groups separated by the ledger, and cheaper selections of residual blocks and of vision-transformer subnets in an ImageNet search space at accuracy close to their references.

Relation to existing approaches. Nuisance quotients and equivariant architectures (Soatto & Chiuso, 2016; Mallat, 2012; Bruna & Mallat, 2013; Cohen & Welling, 2016; Bronstein et al., 2021) organize representations around prescribed transformations; unified accounts (Yu et al., 2022; Dao & Gu, 2024; Gavranović et al., 2024) describe many families by one template or algebra. Here task patterns and their composition determine the structural target; group quotients enter as a subclass of folds. Linear-region counts (Montúfar et al., 2014) describe piecewise-linear expressivity. Our count concerns a declared response-source grammar spanning linear mixing, selection, and joint composition. Zero-cost proxies such as NASWOT, SynFlow, Zen, TE-NAS, and NASGraph (Mellor et al., 2021; Tanaka et al., 2020; Lin et al., 2021; Chen et al., 2021b; Huang et al., 2024) score candidates from initialized networks or graph surrogates, and supernet search trains a shared network over the whole space (Chen et al., 2021a). Our framework describes these families in one class and, from the architecture alone, supports both diagnosis and selection without training any candidate (Section A.11).

2 Two Empirical Principles

Let X be the task domain of complete inputs and $y : X \to \mathcal{Y}$ the target. An input or intermediate field is $A : U \to \mathbb{R}^c$ on a finite set of sites U. A pattern consists of a support $S \subseteq U$ and a response rule $\psi : (\mathbb{R}^c)^S \to \mathbb{R}^d$. Supports can be disconnected or span the whole field. A declared grammar $\mathcal{P}$ specifies the primitives, support choices, wiring, and readouts independently of the target labels. Supports and coefficients can subsequently be learned.

Empirical Principle 1 (Information is pattern)**.** *The task is determined on X by a finite joint tuple of admissible responses:*

$$p_i(x) = \psi_i(x|_{S_i}), \qquad p(x) = (p_1(x), \dots, p_m(x)), \qquad y(x) = r(p(x)).$$

The response rules and readout belong to the label-independent grammar $\mathcal{P}$. The readout family consists of affine maps followed by a threshold or argmax, specified before labels are inspected.

Thus the relevant information is the collection of responses needed to decide the task. The tuple need not reconstruct the input. Its responses must be available in one network at once; having each in a different network is not enough. A large-support response can be formed directly or through intermediate responses on smaller supports; the principle does not prescribe that decomposition.

At a given stage, composing all preceding operations yields effective patterns on the original input domain. A pattern is *exposed* at that stage when it lies in the linear span of these effective patterns and the constant function 1.

Joint coverage. An architecture's formal combination space records which response rules, supports, and shared branches can be assembled in one construction. We say that it *covers a task-generating tuple* when one admissible configuration exposes all patterns of the tuple at one stage, with the declared readout of EP1. Coverage respects the declared grammar, including which parts can coexist and which weights are shared; having enough output slots of the right shape is not enough.

For an architecture E, fix a protocol Π and a declared finite set $\mathcal{I}_E$ of seeds, each fixing all randomness of a run, independently of solution weights. Let $D_{\mathrm{geo}}(E)$ count physical layers on the longest path, and let $S_\Pi(E)$ and $M_\Pi(E)$ be independently specified or calibrated sample and training-computation requirements. The budgets are (D_0, S_0, M_0).

Empirical Principle 2 (Failure is cost)**.** *If the formal combination space of E covers a task-generating joint tuple and*

$$D_{\mathrm{geo}}(E) \le D_0, \qquad S_\Pi(E) \le S_0, \qquad M_\Pi(E) \le M_0,$$

then training under Π reaches an exact realization of y on X from some seed in $\mathcal{I}_E$.

EP1 fixes what a task needs; EP2 says when training can reach it. Budgets cannot be defined retrospectively by whether a run succeeds. Read contrapositively: under joint coverage, if no admissible initialization reaches an exact realization, some declared budget is too small; failure is then a matter of cost (EP2). At a fixed finite budget, achieved accuracy depends jointly on the architecture and the training protocol; natural-task experiments report this performance relative to declared references. The role of a reference model is to calibrate a prediction for subsequent models, which may fail that test.

Empirical tests. EP1 fails in a declared regime if no admissible finite tuple and readout generate y. For EP2, certified joint coverage and sufficient predeclared budgets together with failure from every admissible initialization constitute a counterexample. With a finite declared set this is audited exhaustively. For a distribution $\mathcal{D}_I$ over initializations, a practical version fixes a success floor p_0 and n draws: if the success rate is at least p_0, all n fail with probability at most $(1-p_0)^n$. An architecture without coverage is a structural negative control.

3 Architectures as Pattern Programs

Definition 3.1 (Align–fold program)**.** *An align Q exposes pattern instances in shared coordinates. A fold F combines responses and identifies selected distinctions on its declared reachable domain, possibly none. A QF program is a finite, well-typed directed acyclic composition of stages*

$$A_{\ell+1}(x) = F_\ell(Q_\ell A_\ell(x))$$

and declared wiring operations, including branching, concatenation, and addition.

An align acts on a bank $QA = (q_v(A))_v$, where v need not be a spatial translation. Support extraction uses $q_v(A) = A|_{S_v}$. For a convolution, this is patch extraction and placement; the filter bank then produces responses, with the layer's own weights, by linear folding across the extracted coordinates. For a patch operator P_v, weights W, bias b, activation ρ, and aggregation operator Agg over the window V_u,

$$\begin{aligned} e_A(v) &= P_v A, & s_A(v) &= W e_A(v) + b, \\ r_A(v) &= \rho(s_A(v)), & A'(u,j) &= \mathrm{Agg}_{v\in V_u}\, r_A(v,j). \end{aligned} \tag{1}$$

Thus a 3×3 convolution followed by ReLU is $F_\rho \circ F_W \circ Q_{3\times 3}$, with Agg the identity, and a residual block with an identity shortcut, inference-time batch normalization absorbed into each F_{W_i}, is $x \mapsto F_\rho\big(F_{\mathrm{add}}(x,\, F_{W_2} \circ Q_{3\times 3} \circ F_\rho \circ F_{W_1} \circ Q_{3\times 3}(x))\big)$. An align can discard inputs, and a fold can be injective on the reachable domain. The names specify computational roles; task safety concerns all information retained by the complete program.

Response provenance and span. Each symbolic response retains its original-input support S_{orig}. Composition propagates the union of incoming supports, including control dependencies. With a fixed input metric, a response is *full-span* when $\mathrm{diam}(S_{\mathrm{orig}}) = \mathrm{diam}(U)$. This records range, rather than coverage of every required task distinction. Widths, support choices, and the readout are declared with the candidate family. Local implementations must accumulate enough communication radius to reach a required distant input (Proposition A.3).

The dictionary is open. Adding a primitive can reduce the serial cost of a construction; finite affine–ReLU expressions also approximate continuous responses uniformly on compact domains. The exact/approximate distinction, dictionary-relative depth, and symmetry folds are developed in Sections A.1.10, A.2 and A.3. They explain how the class extends without changing the two empirical principles.

4 COMBINATORIAL FREEDOM OF JOINT PATTERN STRUCTURES

4.1 WHAT IS BEING COUNTED?

For a fixed realization, the patterns exposed at a stage span a response space; across realizations, an architecture forms different joint responses and hence different spaces. Our object is a finite, grammar-based description of their *construction possibilities*.

A configuration records response rules, original-input supports, shared sources, and construction provenance. Formal propagation follows $\mathcal{T}_{\ell+1} = F_{\#}(Q_{\#}(\mathcal{T}_\ell))$, where $\mathcal{T}_\ell$ is the set of configurations at stage ℓ and $Q_{\#}, F_{\#}$ apply the stage's align and fold to each of them. We count a declared set of local composition labels independently, retaining the history of the reduced description; one formal label can correspond to continuously many numerical realizations. Compatibility can restrict which resulting traces are feasible, so the count is a formal upper count of feasible traces.

4.2 RESPONSE-SOURCE COUNTING RULES

The channel responses are treated as a basis of the exposed pattern space. Motivated by the general-position intuition of intersection theory, the ledger treats nonlinear folds as opportunities to transform the exposed basis and change its span. It counts formal transformation choices independently; compatibility constraints and coincident realizations can lower the realized count.

For a fold F and scalar output position q, let $\mathcal{S}_F(q)$ be the already formed response maps or branches read there, and let $m_F(q) = |\mathcal{S}_F(q)|$. Spatial positions within a response map are instances of the same pattern-response rule on different supports; they are not counted as distinct response sources. They add no sources to a combination, while the combination event at each output position is still counted. Thus a convolution has $m_F(q) = c_{\text{in}}$ regardless of kernel size, and a depthwise convolution has one source.

The grammar distinguishes *selection*, with $n_F(q)$ alternatives, from *joint composition*, with binary contribution labels for each of $m_F(q)$ sources. Their local counts and logarithms are

$$N_F(q) = \begin{cases} n_F(q) & \text{selection,} \\ 2^{m_F(q)} & \text{joint composition,} \end{cases} \qquad \ell_F(q) = \begin{cases} \log_2 n_F(q) & \text{selection,} \\ m_F(q) & \text{joint composition.} \end{cases}$$

These are formal source-combination labels, fixed before training. For example, a linear mix of c_{in} maps, such as a convolution or a dense layer, is a selection with $n = c_{\text{in}}$ and contributes $\log_2 c_{\text{in}}$ bits per output, four bits for $c_{\text{in}} = 16$. A ReLU selects between passing and zeroing its input, $n = 2$, one bit; a max of m branches selects one of them, $\log_2 m$ bits. A residual addition of m branches is a joint composition, m bits: the joint count relaxes the fixed sum of all m sources to sums over arbitrary subsets, so an ordinary two-branch residual has four labels (neither, first only, second only, both) and two bits. Joint labels are not measurements of activation masks.

Within-map spatial reweighting, ordinary spatial pooling, and concatenation add no source-combination factor. This preserves one response type while allowing its support and numerical realization to change. A branch that only linearly re-expresses another is not an independent merge source. Consecutive folds, shared-weight repetitions, and the other structural reductions follow the rules in Section A.6.

Definition 4.1 (Formal joint-pattern capacity)**.** *After the declared reductions, let $\mathcal{T}_{\text{formal},\le\ell}(E)$ be the Cartesian product of the local composition-label sets over the set $\mathcal{F}_{\le\ell}$ of folds up to stage ℓ and their scalar outputs $q \in \text{out}(F)$. Define*

$$C_{\le\ell}(E) = \prod_{F\in\mathcal{F}_{\le\ell}} \prod_{q\in\text{out}(F)} N_F(q), \qquad L_{\le\ell}(E) = \log_2 C_{\le\ell}(E) = \sum_{F\in\mathcal{F}_{\le\ell}} \sum_{q\in\text{out}(F)} \ell_F(q).$$

For feasible traces expressed in these same labels, $|\mathcal{T}_{\text{feasible},\le\ell}(E)| \le C_{\le\ell}(E) = |\mathcal{T}_{\text{formal},\le\ell}(E)|$. Write $L(E) = L_{\le\ell_{\text{final}}}(E)$.

L measures, in bits, the formal possibilities for transforming the basis of jointly exposed patterns, i.e., the combinatorial freedom in which patterns combine, not in their coefficients. Later folds contribute because each transforms the basis that earlier folds formed. The count accumulates along

the construction after the declared reductions. The count is computed without weights or activation samples. It does not assert that every counted history is realizable or that all earlier distinctions survive at the output.

In attention, pairwise scores combine the d_h response channels of one head and contribute $\log_2 d_h$ per score; value aggregation reweights positions within each map. GELU and SiLU, smoothed versions of ReLU, are counted like it, one bit per scalar; their measured retention is reported separately as L_φ (Section A.4) and does not enter the screen. Figure 1 illustrates the accounting.

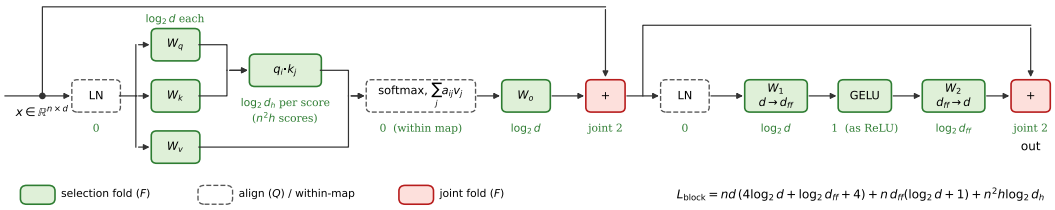

Figure 1: A generic pre-norm transformer block as a QF program. Projections and pairwise scores use selection counts; residual merges use joint counts. Softmax and value aggregation reweight within a map. GELU counts like ReLU, one bit per scalar; its measured retention is reported separately.

4.3 Capacity and cost are complementary

For a dense $k \times k$ convolution on an $H \times W$ output grid, before any consecutive-linear reduction,

$$L_{\text{conv}} = HW c_{\text{out}} \log_2 c_{\text{in}}, \qquad \text{Par} = k^2 c_{\text{in}} c_{\text{out}}, \qquad \text{MAC} = HW \,\text{Par}.$$

Holding channels and kernel fixed, spatial size scales L and MACs together. Holding input channels, kernel, and grid fixed, output width scales all three together. Varying kernel size can change cost without changing the local source count, while adding a counted activation can change L without changing parameters or MACs under our convention. The convention excludes elementwise work from MACs, so equal MACs do not assert equal latency or total operation count.

The measured block family in Figure 2a exhibits both directions: equal costs with different L, and equal L with different costs. Thus $(L, \text{Par}, \text{MAC})$ resolves distinctions absent from either capacity or cost alone. This is the reason to constrain L and optimize cost, rather than maximize L or use it as a replacement for cost.

5 Familiar Architectures as Special Cases

The grammar gives a common view of architectures usually treated as separate families. For each, the same questions apply: which supports are exposed, how their responses are combined, which distinctions can be discarded, and what the implementation costs. Naming a backbone restricts these choices to a sub-family of QF programs.

CNNs: shared local responses composed into larger patterns. In Equation (1), Q extracts translated patches, F_W forms shared responses, and activation and aggregation determine which distinctions remain. Stacking stages builds larger-support patterns from local ones. Weight sharing and local connectivity reduce the resources required to expose repeated structures. They do not restrict the task's patterns to local supports: the composed support is the union of its dependencies. This matches the view of convolution as shared local processing (LeCun et al., 1998) and the advantage of deep hierarchies on compositional functions (Poggio et al., 2017). Changing a spatial kernel can alter span and cost without increasing the number of response-map sources. Changing the cross-map combination or activation structure can instead alter L at the same parameter and MAC counts. Global average pooling additionally shows why a prescribed invariance is not a complete account of folding: distinct inputs outside the same symmetry orbit can share a mean (Proposition A.1).

Transformers: response-controlled combination across token supports. In a vision transformer (Vaswani et al., 2017; Dosovitskiy et al., 2021), patch extraction first aligns input groups; the

embedding produces their responses. Query and key projections form controller responses, and the pairwise score combines their channels. Softmax produces weights for aggregating values over token positions. Under the response-source grammar, the score contributes a cross-channel selection count, whereas aggregation mixes positions within each value map. Head concatenation, output projection, residual addition, and the feed-forward branch then continue the joint response construction (Figure 1). Dense attention exposes global dependencies in one block, with quadratic token-pair work. The distinction from a CNN is therefore a choice of support exposure and response-controlled composition, together with its cost, rather than a different kind of task information.

State-space mixers: extending the support mechanism. The Mamba mixer (Gu & Dao, 2024) adds a prefix align Q_{ssm}: position t receives responses on $\{0, \ldots, t\}$ through the declared selective scan. Local convolution, controller responses, product gating, and output projection use the same remaining roles. Its scan parameterization and computation are charged to cost. This example shows how a new support mechanism extends the dictionary while retaining the same construction and comparison questions.

What the common view enables. A dense MLP exposes global supports immediately; a CNN can realize some of the same responses with local sharing at lower cost. Attention and state-space mechanisms offer other support/composition choices. The framework separates this structural choice from whether it covers the task and whether the learning budget suffices. Detailed constructions are in Section A.7.

6 Pre-training Diagnosis and Model Construction

6.1 Diagnosing coverage and failure

The two principles suggest a sequence: identify a task-generating joint response, assess whether the architecture covers it, and evaluate the learning budget. For known patterns, an explicit realization f_E, a decoding map δ with $p = \delta \circ f_E$, and an admissible readout $r \circ \delta$ certify coverage. Under EP2, sufficient declared budgets then imply a successful admissible initialization.

Different joint pattern realizations can induce different partitions of the input domain. The fiber criterion characterizes which of these partitions preserve the distinctions required by the task. It provides a semantic link between pattern exposure and task realizability, while construction and screening operate on the formal ledger.

Fibers partition *complete inputs* in X, such as images, by equality of their cumulative representations; they do not partition pixels within an image. Let π_f be the fiber partition of f, let π_L be the label partition, and let $\preceq$ mean refinement. Include all branches still available downstream in f.

Proposition 6.1 (Exact decision-losslessness criterion)**.** *A representation f admits a deterministic readout g on* $\operatorname{im} f$ *with $y = g \circ f$ if and only if $\pi_f \preceq \pi_L$.*

Indeed, $g(f(x)) = y(x)$ is well defined exactly when every fiber carries one label. Membership of g in a restricted readout family is an additional condition; failure of a linear readout alone does not imply a collision.

Theorem 6.2 (Irreversibility)**.** *For $f : X \to Z$ and $h : Z \to W$, $\pi_f \preceq \pi_{h \circ f}$. Once two differently labelled inputs have the same cumulative representation, no downstream deterministic computation can distinguish them.*

A collision proved for every allowed parameter rules out the architecture; a collision for one trained parameter choice rules out only that realization. These statements require neither empirical principle. The linear subclass admits an exact optimum $\ker P = N_y$, where N_y is the task's translation-invariant subspace (Theorem A.4).

6.2 Calibrating a capacity floor

For natural tasks the required joint tuple is unknown, and so is the number of combinations it would need. The threshold belongs to the task, under a declared dictionary and protocol, not to any model.

A reference that meets its declared performance shows that its capacity suffices, so the L read off it is an upper bound on what the task requires at that performance; a reference with larger L gives a looser bound, and the reference with the smallest L meeting the same declared performance the tightest. Fix a reference regime, response-width requirement k, measurement stage, and threshold L_{target} before evaluating candidates.

Capacity is used as a floor. A candidate above the floor need not improve as L rises. Passing the calibrated L threshold predicts joint coverage within the declared family. Conditional on coverage, EP2 posits that exact realization is reachable from some admissible initialization within sufficient declared budgets. Experiments on screened candidates jointly test the coverage prediction and the resource calibration.

6.3 Constructing within a finite family

Let $\mathcal{A}_0$ be a finite family with a declared dictionary, wiring, widths, readouts, and resource bounds. Attach a readout only at a completed full-span stage with width at least k and cumulative $L_{\le \ell} \ge L_{\mathrm{target}}$. Let $\mathcal{A}$ contain the candidates passing this structural screen and all cost bounds. For $\widehat{K}(E) = (D_{\mathrm{geo}}(E), \mathrm{Par}(E), \mathrm{MAC}(E))$ and a cost objective J fixed before construction, choose

$$\min_{E \in \mathcal{A}} J(\widehat{K}(E)), \tag{2}$$

or return the nondominated cost vectors. Depth is a cost in its own right: deeper networks are harder to optimize (He et al., 2016a), and longer serial paths are slower.

Proposition 6.3 (Budgeted selection)**.** *Finite enumeration finds a minimum-cost admissible candidate whenever one exists. Pruning a prefix whose monotone budget is exceeded, or whose valid final-objective lower bound exceeds an incumbent cost, preserves an optimum.*

The proof and procedure are in Sections A.1.5 and A.1.6; L is computed by arithmetic, and training then tests the selected candidate's performance.

7 Experiments

Protocols and registries are in Sections A.9 and A.14.

7.1 The class is exact

Independent QF and reference implementations agree in double precision over five parameter draws and four input batches per draw. ResNet blocks with and without downsampling agree to maximum forward error 1.8×10^{-14} and relative parameter-gradient error 7.0×10^{-14}; the pre-norm ViT block agrees exactly; the mixer of an official Mamba-130M agrees to 3.6×10^{-11} and 1.4×10^{-10} and needs one additional align, the prefix scan.

7.2 Coverage, decided before training

EP1 on a complete domain. On all 512 binary 3×3 images, six pattern grammars were declared first and seven tasks defined independently, one held out until the grammars were fixed. For each pair, exhaustive search decides whether responses collide across labels (Proposition 6.1), and an exact linear program decides whether the declared affine readout realizes the task. Every verdict carries an integer certificate (Section A.12).

Single pixels keep all information, yet six of seven tasks are unreadable; 2×2 counts lose it on every task; all pairs fail only parity, and by readout. The certificates distinguish information loss from insufficient readout expressivity.

Certificates on toy and real inputs. On an 8-bit task, two architectures identical in encoder, parameters, MACs, and depth differ only in which input pairs a zero-parameter gather exposes. For the local gather an explicit pair of inputs with different labels collides for every weight (Proposition A.7); for the far gather a witness realizes the task on all 2^8 inputs. Trained on half the domain, the far

Table 1: EP1 on all 512 binary 3×3 images. C: responses collide across labels; I: no collision, but no affine readout; F: feasible.

grammar	bar	corners	majority	parity	mirror	corner XOR	held out
single pixels	I	I	F	I	I	I	I
adjacent pairs	I	I	C	C	I	I	I
all pairs	F	F	F	I	F	F	F
2×2 window counts	C	C	C	C	C	C	C
rows and columns	F	C	F	I	C	C	C
whole image	F	F	F	F	F	F	F

architecture reaches full-domain accuracy 1.000 in every seed; the local one stops at 0.745. On natural images, CIFAR-10 (Krizhevsky & Hinton, 2009) cut into 2×2 tiles and permuted, with the permutation as label: a ViT without position embedding is invariant to tile permutations at every weight (checked at initialization to 4×10^{-7}), so every readout is at chance, and three seeds give 3.9–4.7% against 4.17%. With position embedding the task is learnable (26–33% in two of three seeds). Sixteen patterns declared before labels were inspected—seam continuity, seam gradients, tile orientation, content at slot—reach 34.7%, eight times chance, with the declared affine readout alone, above all $3{,}072$ raw pixels (28.0%). Published results without position embedding fall on the same lines (Section A.12). The certificate also names the repair: on a second enumerable task, restoring the response a fold had removed raises accuracy from 88.9% to 98.4%, while widening the same fold leaves it at 89.1%.

Failure is cost. A budget rule calibrated at 4, 6, and 8 inputs and frozen predicted every held-out Pair-QF case and underestimated the training budget required by the unrestricted MLP at 10 and 12 inputs. An explicit witness certifies that the MLP covers the task (Section A.9). We then attempted to falsify EP2 on a predeclared set of 30 seeds at four times each optimizer's extrapolated budget: failure for every seed would be a counterexample. Under SGD, $28/30$ and $26/30$ seeds reach an exact realization at 10 and 12 inputs; the same seeds under Adam reach $30/30$ at both. Increasing the training budget alone let the same architecture reach an exact realization.

7.3 Selection before training

The vision-transformer family. The ViT-Lite family on CIFAR-10 has 24 members (depth, width, MLP ratio; patch 4); its threshold is the capacity of the smallest reported reference, ViT-Lite-6/4 (Hassani et al., 2021). Pricing all 24 took seconds; five were trained with the reference recipe (Figure 2b; Table 11). The cheapest member meeting the threshold is ViT-Lite-6/4 itself, the hand-designed model at its own capacity. At 0.998 of the threshold, a configuration with 12% fewer parameters than ViT-Lite-6/4 and 25% fewer than ViT-Lite-7/4 matches it (92.80%); the two at 0.95 score 92.54% and 92.40%. The threshold is a conservative bound, and accuracy is lower further below it. In the same family the published patch-8 and patch-16 models have more parameters than patch 4 and lower accuracy; their L falls with accuracy.

What L sees that cost does not. A matched-cost comparison trains 14 of 140 enumerated residual-block schemas on CIFAR-100 with three seeds. Their parameters differ only by kernel order, and their MACs and depth are identical within each order: cost cannot separate them. The ledger admits seven and excludes seven, and the admitted seven are the seven most accurate (64.1–66.4% against 62.0–63.2%), with no interleaving. The criterion is one structural fact, whether a nonlinearity separates the block's two convolutions; without it the two compose into one linear fold and L falls below the threshold (Figure 2a, Table 7). The separation holds run by run: the lowest of the 21 admitted runs (63.9%) is above the highest of the 21 excluded (63.3%).

Above the threshold L is not a score: the two-ReLU block has more capacity and less accuracy, its second ReLU sitting before the residual addition (He et al., 2016b). The one-ReLU block has the standard block's L with 53% of its parameters and a 2.4-point accuracy drop; narrowing the standard block to the same parameters loses 3.9 (interpolated between widths 11 and 12). Post-training filter pruning of ResNet-20 to 37–43% fewer parameters loses 1.0–2.7 points after 300 epochs

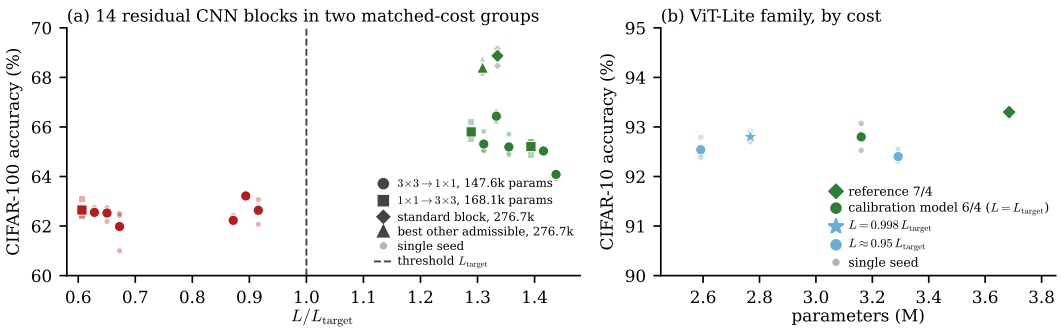

Figure 2: Accuracy after training against L before training. Large markers are means over three seeds, small points single seeds; green has $L \geq L_{\text{target}}$, red and blue have $L < L_{\text{target}}$. (a) Fourteen residual blocks on CIFAR-100, cost matched within each kernel order, the standard block (diamond), and the best of the other 32 admissible blocks (triangle; the standard block's cost); the threshold was read from a structural variant fixed before they were trained. (b) ViT-Lite on CIFAR-10 against parameters; the threshold is the capacity of the smallest reported reference, 6/4, and the star marks the configuration at 0.998 of it.

Table 2: The ledger's block, chosen on CIFAR-100, across tasks (accuracy, %); two seeds, three for CIFAR-100; same recipe, flips off for digits.

	MNIST	Fashion	SVHN	CIFAR-10	C-100 super	CIFAR-100
ledger block	99.73	94.85	96.50	91.21	77.59	66.43
standard	99.68	94.89	96.78	92.35	79.02	68.87

of fine-tuning (Zhang et al., 2023); the ledger's block achieves a comparable parameter–accuracy trade-off with no prior training. At equal parameters and MACs, an additive merge carries one more bit per merge site than a max merge and scores 1.4 points higher. Across all 33 admissible blocks (Section A.9), those three convolutions deep ($D_{\text{geo}} = 73$) average 63.8–64.2%, below the two-convolution blocks with fewer parameters (65.2–65.3%), consistent with depth as a cost (EP2); of the 28 cheaper than the standard block, the ledger's block scores highest.

Across tasks. At about half the standard block's parameters, the same block is within 0.3 points of it on MNIST, Fashion-MNIST, and SVHN, and 1.1–2.4 points below on the natural-image tasks (Table 2).

Inside a searched space. AutoFormer searched ViT architectures on ImageNet (Deng et al., 2009) by training a supernet (Chen et al., 2021a). In its tiny space, with released inherited-weight accuracies for 200 subnets (Wei et al., 2024), the ledger's single front member uses 98% of AutoFormer-T's parameters and 92% of its depth and scores 75.35% against the released AutoFormer-T's 75.3%. In its base space, over $1{,}001$ subnets (Ho et al., 2024), every front member scores 81.8–82.2% at 0.90–0.985 of AutoFormer-B's parameters. A threshold read from the hand-designed DeiT-B would have excluded an additional 799 working subnets, the best one among them: a threshold read from a larger reference is a looser bound (Section 6.2). The language-model case is in Section A.12.

8 Conclusion

Architectures are programs of aligns and folds; convolution, attention, and state-space mixing are members of one class, exactly. From the architecture alone, the ledger counts, as a formal upper bound, the ways a structure can transform the basis of the patterns it exposes, and fibers tie that exposure to what a task needs. The framework supports diagnosis before training and, with L as a threshold and cost as the objective, constructs cheaper architectures within declared families. We look forward to extending the dictionary of primitives and exploring the architectures it makes possible.

AI USE STATEMENT

The core ideas, including the conceptual framework and the two empirical principles, originated with the authors. Every definition, counting rule, and experimental design was decided by the authors. Generative AI tools assisted with wording, literature search, feedback on drafts, and implementing training and audit scripts to author-specified protocols; draft feedback included Google's Paper Assistant Tool (PAT), offered to authors through the ICLR 2027 program. The authors checked all such output and take full responsibility for the text, proofs, code, and results.

REPRODUCIBILITY STATEMENT

Proofs, construction rules, and translations are in Sections A.1, A.7 and A.8. Experimental settings and reported registry categories are in Sections A.9 and A.14. The execution-cost convention used throughout is stated in Section A.5. Unreported selected-candidate outcomes are not inferred from the completed subset. Training scripts, enumerators, and exactness checks are intended for release as anonymous supplementary material.

ETHICS STATEMENT

The experiments use public benchmark datasets and a public model checkpoint, involve no human subjects, and raise no concerns beyond those common to architecture research.

REFERENCES

Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL https://arxiv.org/abs/2104.13478.

Joan Bruna and Stephane Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, 2013. doi: 10.1109/TPAMI.2012.230.

Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *ICCV*, pp. 12250–12260, 2021a. URL https://doi.org/10.1109/ICCV48922.2021.01205.

Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four GPU hours: A theoretically inspired perspective. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=Cnon5ezMHtu.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/cohenc16.html.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale.

In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023. URL https://arxiv.org/abs/2305.07759.

Bruno Gavranović, Paul Lessard, Andrew Joseph Dudzik, Tamara Von Glehn, João Guilherme Madeira Araújo, and Petar Veličković. Position: Categorical deep learning is an algebraic theory of all architectures. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 15209–15241. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/gavranovic24a.html.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers, 2021. URL https://arxiv.org/abs/2104.05704.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. Transformer language models without positional encodings still learn positional information. In *EMNLP (Findings)*, pp. 1382–1390, 2022. URL https://aclanthology.org/2022.findings-emnlp.99.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016a. doi: 10.1109/CVPR.2016.90.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision – ECCV 2016*, pp. 630–645, Cham, 2016b. Springer International Publishing. ISBN 978-3-319-46493-0.

Sy-Tuyen Ho, Tuan Van Vo, Somayeh Ebrahimkhani, and Ngai man Cheung. Vision transformer neural architecture search for out-of-distribution generalization: Benchmark and insights. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=2AIwiIkE0s.

Zhenhan Huang, Tejaswini Pedapati, Pin-Yu Chen, Chunheng Jiang, and Jianxi Gao. Graph is all you need? lightweight data-agnostic neural architecture search without training. In *AutoML Conference 2024 (Workshop Track)*, 2024. URL https://openreview.net/forum?id=0DWTTOvx73.

Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. The impact of positional encoding on length generalization in transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Drrl2gcjzl.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Ming Lin, Pichao Wang, Zhenhong Sun, Hesen Chen, Xiuyu Sun, Qi Qian, Hao Li, and Rong Jin. Zen-nas: A zero-shot nas for high-performance image recognition. In *ICCV*, pp. 337–346, 2021. URL https://doi.org/10.1109/ICCV48922.2021.00040.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. In *NeurIPS*, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/baa2da9ae4bfed26520bb61d259a3653-Abstract-Conference.html.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yuxing Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 9992–10002, 2021. URL https://doi.org/10.1109/ICCV48922.2021.00986.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. MobileLLM: Optimizing sub-billion parameter language models for on-device use cases. In *Forty-first International Conference on Machine Learning*, 2024b. URL https://openreview.net/forum?id=EIGbXbxcUQ.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pp. 11966–11976, 2022. URL https://doi.org/10.1109/CVPR52688.2022.01167.

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65 (10):1331–1398, 2012. doi: 10.1002/cpa.21413. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21413.

Joe Mellor, Jack Turner, Amos J. Storkey, and Elliot J. Crowley. Neural architecture search without training. In *ICML*, pp. 7588–7598, 2021. URL http://proceedings.mlr.press/v139/mellor21a.html.

Guido F. Montúfar, Razvan Pascanu, KyungHyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *NIPS*, pp. 2924–2932, 2014. URL http://papers.nips.cc/paper/5422-on-the-number-of-linear-regions-of-deep-neural-networks.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Sergei Ovchinnikov. Max-min representation of piecewise linear functions. *Beiträge zur Algebra und Geometrie*, 43(1):297–302, 2002. URL https://eudml.org/doc/225460.

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14(5):503–519, 2017. doi: 10.1007/s11633-017-1054-2. URL https://doi.org/10.1007/s11633-017-1054-2.

Sophia Sanborn, Christian A Shewmake, Bruno Olshausen, and Christopher J. Hillar. Bispectral neural networks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=xnsg4pfKb7.

Stefano Soatto and Alessandro Chiuso. Modeling Visual Representations: Defining Properties and Deep Approximations. In *International Conference on Learning Representations*, 2016. URL https://mlanthology.org/iclr/2016/soatto2016iclr-modeling/.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/46a4378f835dc8040c8057beb6a2da52-Abstract.html.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pp. 10347–10357, 2021. URL http://proceedings.mlr.press/v139/touvron21a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pp. 5998–6008, 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.

Zimian Wei, Peijie Dong, Zheng Hui, Anggeng Li, Lujun Li, Menglong Lu, Hengyue Pan, and Dongsheng Li. Auto-prox: Training-free vision transformer architecture search via automatic proxy discovery. In *AAAI*, pp. 15814–15822, 2024. URL https://doi.org/10.1609/aaai.v38i14.29511.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4484–4496, 2025. doi: 10.1109/CVPR52734.2025.00423.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10809–10819, 2022. doi: 10.1109/CVPR52688.2022.01055.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R. Salakhutdinov, and Alexander Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3391–3401. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

Chaoyan Zhang, Cheng Li, Baolong Guo, and Nannan Liao. Neural network compression via low frequency preference. *Remote Sensing*, 15(12):3144, 2023. doi: 10.3390/rs15123144. URL https://www.mdpi.com/2072-4292/15/12/3144.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=YbHCqn4qF4.

A Appendix

A.1 Proofs

A.1.1 Convolution and invariant averaging

Proposition A.1 (Translation commutants and invariant averages)**.** *Let U be a finite cyclic grid. A linear map L between feature fields on U commutes with every translation if and only if it is a block-circulant convolution. On a declared reachable domain $\mathcal{R}$, its response map introduces no collision on the reachable domain precisely when*

$$\ker L \cap (\mathcal{R} - \mathcal{R}) = \{0\};$$

otherwise the nonzero intersection produces reachable collisions.

Let a finite group G act orthogonally on a feature space through T_g, and set

$$P_G = \frac{1}{|G|} \sum_{g \in G} T_g.$$

Then P_G is the orthogonal projection onto the invariant subspace. On a G-stable task domain X, let R_G be the orbit relation, $x \, R_G \, x'$ iff $x' = T_g x$ for some $g \in G$, and let R_{P_G} be equality under P_G. Then $R_G \subseteq R_{P_G}$, with equality exactly when the induced map $X/G \to \operatorname{im} P_G$ is injective. Hence averaging is decision-lossless exactly when $\pi_{P_G|_X} \preceq \pi_L$; invariance alone does not prevent over-folding.

For the convolution claim, commutation forces the spatial column at every site to be the corresponding translate of the column at the origin, which is exactly the block-circulant form; the converse follows by reindexing the convolution sum. For averaging, reindexing gives $P_G T_h = P_G$ and $P_G^2 = P_G$, while orthogonality gives $P_G^* = P_G$. Thus P_G is the stated projection. Every orbit lies in one fiber, and equality of the two relations is precisely injectivity of the induced orbit map. The final claim is Proposition 6.1.

A.1.2 IRREVERSIBILITY

If $f(x) = f(x')$, then $h(f(x)) = h(f(x'))$, so every f-fiber is contained in an $(h \circ f)$-fiber. If an identified pair has different labels, it remains identified under every downstream composition and Proposition 6.1 rules out an exact readout.

A.1.3 CANONICAL FIBER FACTORIZATION

For a deterministic map $g : Z \to Z'$, define $z \sim_g z'$ iff $g(z) = g(z')$. Then $g = \iota_g \circ F_g$, where $F_g : Z \to Z/\sim_g$ is the quotient and $\iota_g([z]) = g(z)$ is injective. This is a set-theoretic background fact, not a guarantee that either factor has a finite expression in the dictionary. The quotient map F_g is surjective by construction. The rule $\iota_g([z]) = g(z)$ is well-defined because equivalent points have equal g-value, and it is injective because $\iota_g([z]) = \iota_g([z'])$ implies $g(z) = g(z')$, hence $[z] = [z']$. Thus $g = \iota_g \circ F_g$.

A.1.4 PRESERVATION UNDER COMPOSITION

For joint outputs $H(x) = (F(x), B(x))$, equality of H is exactly simultaneous equality of F and B. Hence H discards only distinctions both outputs discard, regardless of implementation wiring. The map B need not itself be injective. A sufficient condition for an additive residual map $x + b(x)$ to be injective is that $b : \mathbb{R}^d \to \mathbb{R}^d$ be L-Lipschitz with $L < 1$: equal outputs imply $\|x - x'\| \leq L\|x - x'\|$, forcing $x = x'$.

A.1.5 BUDGETED SELECTION

Finite operator, wiring and width choices, together with a finite bound on internal operations, give a finite construction tree. If a prefix exceeds a budget nondecreasing under extension, every completion also exceeds it. Removing that prefix therefore removes no within-budget candidate. Checking all remaining completions against the declared structural requirements and comparing total costs proves the proposition. If later graph simplification can reduce a cost, early pruning requires a valid lower bound on final cost; otherwise that cost is checked only on completed networks. Likewise, if a prefix has a lower bound on final objective exceeding the cost of an already completed admissible candidate, none of its completions can improve that candidate. Updating this incumbent bound therefore preserves an optimum.

A.1.6 CONSTRUCTION PROCEDURE

Inputs are a finite grammar and wiring family, finite width choices, a finite bound on internal operations, a declared response-width requirement k, a cumulative capacity threshold L_{target}, resource bounds, and a cost objective J. A candidate state retains original-input supports, response-source provenance, the reduced expression, capacity, and costs. The reduction rules of the dictionary are applied before evaluating a completed expression. The first full-span stage is a useful calibration location; further full-span stages can be explored within the declared family when the screen has not yet been met.

```
frontier := {input field}; candidates := empty set; best :=
infinity
while frontier is not empty:
  remove one partial construction N
  if valid final-cost lower bound(N) > best:  continue
  for each legal next stage allowed by the finite family:
    extend N; propagate supports and response-source
provenance
    apply declared reductions; compute cumulative L and
costs
    if a monotone resource bound is exceeded:  continue
    if full span, width >= k, and L >= L_target:
      attach each allowed compatible readout
      if all final budgets hold:  record E; update best
    if further extension belongs to the family:  add to
frontier
```

```
    return minimum-J completed candidate, or none in the family
    freeze the selected structure; then train and evaluate
```

Readouts contribute to implementation cost and are excluded from L under the declared convention. No candidate weights are trained during this procedure. Prefix pruning uses monotone budgets or valid lower bounds; if simplification can reduce a cost, the unreduced accumulated value alone is not a valid pruning bound. A finite operation bound together with finite primitive, wiring, and width choices makes the tree finite, including when routing operations have zero arithmetic cost. The result optimizes the structural screen and cost objective, not an unknown trained accuracy.

A.1.7 Greedy folding need not minimize atomic transition count

Proposition A.2 (A finite greedy counterexample)**.** *Maximizing the immediate number of newly identified within-label pairs at each admissible fold need not reach π_L in the minimum number of atomic transitions in a declared transition graph.*

Let $X = \{1, \dots, 8\}$ and

$$\pi_L = \big\{\{1, \dots, 7\}, \{8\}\big\}.$$

Starting from the discrete partition $\hat{0} = \{\{1\}, \dots, \{8\}\}$, define

$$\alpha = \big\{\{1, \dots, 5\}, \{6\}, \{7\}, \{8\}\big\},$$
$$\beta = \big\{\{1, \dots, 4\}, \{5, 6, 7\}, \{8\}\big\},$$
$$\gamma = \big\{\{1, \dots, 6\}, \{7\}, \{8\}\big\}.$$

Declare the admissible atomic fold transitions for this example to be exactly

$$\hat{0} \to \alpha, \quad \hat{0} \to \beta, \quad \beta \to \pi_L, \quad \alpha \to \gamma, \quad \gamma \to \pi_L.$$

All five folds are decision-lossless. From $\hat{0}$, the immediate unordered-pair gains are

$$\hat{0} \to \alpha : \binom{5}{2} = 10,$$
$$\hat{0} \to \beta : \binom{4}{2} + \binom{3}{2} = 9.$$

The greedy rule therefore chooses α and is forced to take three atomic transitions $\hat{0} \to \alpha \to \gamma \to \pi_L$. The alternative $\hat{0} \to \beta \to \pi_L$ takes two, which is minimal because the graph has no direct edge from $\hat{0}$ to π_L. Both paths are consecutive fold runs and each counts once under Definition A.6. Thus the example separates greedy and optimal atomic transition counts, not their algebraic depths. Reaching π_L is the prescribed endpoint here; decision-losslessness alone already holds at every intermediate partition and does not require this endpoint.

A.1.8 Locality bound

Dependency span is the largest diameter of input supports available jointly to a response or readout. It is traced by taking unions of incoming supports, including control dependencies. Widths depend on a proposed decomposition's intermediate responses and may be chosen empirically; a final pattern count is not a universal width bound. Greedy folding can waste atomic transitions even though consecutive folds count as one stage (Proposition A.2).

Measure all dependencies on one fixed graph; a pattern may depend on sites anywhere in U, and a required pattern may reach across the whole field. Suppose each physical layer, including resolution changes and the readout at site o, communicates within radius $r > 0$.

Proposition A.3 (Locality lower bound)**.** *Measure the communication radius of each physical layer on the original input graph, so that a layer of nominal radius κ_ℓ after accumulated stride $\prod_{j<\ell} s_j$ has $r_\ell = \kappa_\ell \prod_{j<\ell} s_j$ (Equation (3)). If two admissible inputs differing only at a site distance R from o have different targets, every exact implementation satisfies*

$$\sum_{\ell \le D_{\mathrm{geo}}(E)} r_\ell \;\ge\; R;$$

for unstrided layers of common radius r this is $D_{\mathrm{geo}}(E) \ge \lceil R/r \rceil$.

This lower bound constrains the earliest possible full-span stage; the actual stage is found by support propagation.

A library restricted to local communication must propagate a distant dependence through successive layers; a global primitive can expose it at once. The radius assumption restricts this implementation class, not admissible pattern supports.

For the locality bound, after L physical layers of input-graph radii $r_1, \dots, r_L$, including the readout at o, that output can depend only on sites within graph distance $\sum_{\ell \le L} r_\ell$ of o; genuine dependence at distance R therefore forces $\sum_{\ell \le L} r_\ell \ge R$. Downsampling enlarges later radii on the input graph, which is why the bound is stated with accumulated stride.

A.1.9 Linear fold optimum and compositions

On $\mathbb{R}^n$ let the decision-null subspace be

$$N_y = \{v : y(x + tv) = y(x) \text{ for all } x, t\}.$$

Theorem A.4 (Linear optimal fold)**.** *With unrestricted deterministic readouts on the full domain $\mathbb{R}^n$, a linear representation P is decision-lossless exactly when $\ker P \subseteq N_y$. Consequently $\operatorname{rank}(P) \ge n - \dim N_y$, with equality exactly when $\ker P = N_y$, so $\mathbb{R}^n \to \mathbb{R}^n/N_y$ is the maximally compressed decision-lossless linear fold.*

Here the domain is all of $\mathbb{R}^n$, and readouts are unrestricted; the same formula need not apply to a restricted reachable subset. First, N_y is a subspace. Scaling is immediate; if $v, w \in N_y$, then for all a, b, t,

$$\begin{aligned} y\big(x + t(av + bw)\big) &= y\big((x + tav) + tbw\big) \\ &= y(x + tav) = y(x). \end{aligned}$$

For linear P, equal representations mean that the input difference lies in $\ker P$. Decision losslessness is therefore equivalent to $\ker P \subseteq N_y$. Rank–nullity gives

$$\operatorname{rank}(P) = n - \dim \ker P \ge n - \dim N_y,$$

with equality exactly when $\ker P = N_y$.

If $P_i = L_i \circ P_{i-1}$ are cumulative maps in a linear chain, then $\ker P_{i-1} \subseteq \ker P_i$. Since every map in the declared chain is decision-lossless, each kernel lies in N_y. Conversely, given nested subspaces $\{0\} = K_0 \subseteq \cdots \subseteq K_r \subseteq N_y$, set $P_i = I - \Pi_{K_i}$, the identity minus the orthogonal projection onto K_i. Nestedness gives $P_i P_{i-1} = P_i$, so choosing $L_i = P_i$ realizes these as cumulative maps with $\ker P_i = K_i$. Because $K_i \subseteq N_y$, every P_i is decision-lossless. Rank–nullity applied to the restriction $P|_{N_y}$ gives

$$\dim P(N_y) = \dim N_y - \dim(N_y \cap \ker P).$$

For decision-lossless P, $\ker P \subseteq N_y$, so this dimension is zero exactly when $N_y \subseteq \ker P$, hence exactly when $\ker P = N_y$. Finally, if $N_i \perp N_j$, then the folds $I - \Pi_{N_i}$ and $I - \Pi_{N_j}$ commute because both cross-products of the orthogonal projections vanish; such independent folds need no algebraic serial order.

A.1.10 Completion of the finite-stage class

Finite compositions need not realize every operation exactly, but their limits can be richer. For compact $K \subset \mathbb{R}^n$, let $\mathcal{R}_{\text{fin}}(K, \mathbb{R}^m)$ denote restrictions of finite affine–ReLU programs to K.

Theorem A.5 (Completion of finite affine–ReLU programs)**.** *Under the uniform norm, the metric completion of $\mathcal{R}_{\text{fin}}(K, \mathbb{R}^m)$ is $C(K, \mathbb{R}^m)$.*

For continuous g on compact $K \subset \mathbb{R}^n$, extend each coordinate continuously to a containing box and triangulate finely enough to obtain a continuous piecewise-affine interpolant with arbitrarily small uniform error on K. Every scalar continuous piecewise-affine function is a finite max–min expression of affine functions (Ovchinnikov, 2002). The identities

$$\begin{aligned} \max(a, b) &= \operatorname{ReLU}(a - b) + b, \\ \min(a, b) &= a - \operatorname{ReLU}(a - b) \end{aligned}$$

give a finite affine–ReLU realization. Thus $\mathcal{R}_{\text{fin}}(K, \mathbb{R}^m)$ is dense in the complete space $C(K, \mathbb{R}^m)$, which identifies its metric completion. For GELU $\sigma(x) = x\Phi(x)$, with Φ the standard normal distribution function, on $[-R, R]$ with $R > 0$, take the uniform k-interval interpolant E_k, $k \geq 1$. The mesh width is $h = 2R/k$, so

$$\|E_k - \sigma\|_\infty \leq \frac{(2R)^2}{8k^2} \sup_{|x| \leq R} |\sigma''(x)|.$$

The interpolant is continuous piecewise affine and therefore finite affine–ReLU. Every finite affine–ReLU program is piecewise affine with finitely many regions, whereas $\sigma''(x) = \phi(x)(2 - x^2)$, with ϕ the standard normal density, vanishes on no open interval; hence GELU cannot agree with any such finite program on a nontrivial interval.

A.2 Fold mechanisms and symmetry

Extraction and placement align. The linear fold F_W folds whatever kernel directions it collapses; ReLU folds negative magnitudes; nontrivial aggregation may fold spatial configurations, and may be the identity, in which case later convolutions act on the retained field. Global average pooling shows why symmetry is only one fold mechanism: averaging over a group G projects onto the invariant subspace, but the relation it induces contains the orbit relation strictly in general, since distinct orbits may share a mean; invariance does not prevent discarding more than the group does (Proposition A.1).

Three examples cover the familiar cases. *Value folds* act on the response axis, as ReLU does; *spatial folds* identify site configurations, as pooling does; *interaction folds* remove directions in a product of branches, so addition $(a, b) \mapsto a + b$ removes every anti-diagonal $(t, -t)$ while concatenation removes nothing.

Folds quotienting a declared group action form a well-developed subclass, where a fold is *complete* when invariant and also separating, so $\pi_F = \pi_G$, the partition of X into G-orbits, rather than $\pi_G \prec \pi_F$. Averaging gives only invariance, the over-folding recorded in Proposition A.1. Sorting realizes the permutation quotient; bispectral completeness requires its stated nondegeneracy conditions (Zaheer et al., 2017; Sanborn et al., 2023). The dictionary admits such primitives; it does not have to invent them.

Task safety asks whether the cumulative representation preserves the required label distinctions (Proposition 6.1); no prescribed symmetry group is needed. Folds may act differently across regions or channels, with other outputs retaining distinctions still needed by the task. Their composition and order are therefore design choices.

Bypass preservation can be realized by linear mixing; one of the simplest examples is this invertible map retaining x_2 alongside the sum:

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ x_1 + x_2 \end{pmatrix}.$$

A.3 Dictionary-relative depth

GELU also exposes a choice that the dictionary makes visible. It is one layer in a library that contains it, while a more elementary affine–ReLU dictionary admits only finite approximations whose complexity grows as the requested accuracy is refined. Depth therefore needs two notions.

Fix a dictionary $\mathcal{D}$, wiring rules, readout family and allowed counting reductions. Write $\ell_{\text{QF}}(A)$ for the longest serial stage count of an expression A, after merging consecutive aligns or folds. An expression is exact for y when it computes y on the full task domain with its declared readout.

Definition A.6 (Algebraic and geometric depth)**.** *The algebraic depth of a target is*

$$D^{\mathcal{D}}_{\text{alg}}(y) = \min_{A \text{ exact for } y \text{ over } \mathcal{D}} \ell_{\text{QF}}(A),$$

with value $+\infty$ if no exact expression exists. The geometric depth $D_{\text{geo}}(E)$ counts physical layers on the longest path of an implementation E.

The readout and stage-counting conventions are fixed with the dictionary. Every exact expression A gives $D^{\mathcal{D}}_{\mathrm{alg}}(y) \le \ell_{\mathrm{QF}}(A)$; expressions need not be unique or shortest. A translated network gives this task bound only if it realizes the task exactly. Combining a run for stage counting does not promote it to an atomic dictionary entry or reduce its physical cost. A small minimum, even one stage, is allowed.

If a construction uses three stages and a new primitive realizes the same computation in one, the certified upper bound improves from three to one. This does not retroactively prove that the old dictionary had a smaller minimum. Enlarging a dictionary cannot increase $D^{\mathcal{D}}_{\mathrm{alg}}(y)$ when the old counting rules are retained. This expresses the design trade between primitive richness and serial composition; physical layer count additionally depends on implementation.

A.4 Supplementary continuous-gate audit

A continuous gate such as GELU, SiLU, or sigmoid is recorded with the three-state vocabulary $\{0, \epsilon, 1\}$, $0 < \epsilon < 1$, where 0 deletes a response, 1 retains it, and ϵ denotes weak retention. For the g weak-retention events of a declared audit with retained scales $\epsilon_1, \dots, \epsilon_g$,

$$C_\varphi := \prod_{i=1}^{g} \big(2 + \varphi(\epsilon_i)\big), \qquad L_\varphi := \log_2 C_\varphi,$$

with $\varphi(0) = 0$ and φ nondecreasing on $[0, 1]$; the binary convention is $\varphi \equiv 0$, and an audit may take $\varphi(\epsilon) = \epsilon$. ϵ_i is not fixed globally but measured at each declared gate position as the median retention ratio on a stated model and input sample. This is a separate, sample-dependent gate audit. The binary convention gives one bit per gate, which is what L already counts; a measured ϵ refines it. Calibration and construction use L alone.

For the Mamba checkpoint, the two reported SiLU medians are 0.4924 and 0.4569, from $4{,}718{,}592$ observations at each gate position. These measurements describe numerical retention on the stated sample, separately from the pre-training count. Over the audited sample (eight inputs of 128 tokens in all, 24 mixers, two gate positions, $9{,}437{,}184$ sites) this gives $L_\varphi \simeq 1.23 \times 10^7$ bits; at a context of $2{,}048$ tokens the same medians give 1.97×10^8. Neither enters any screen.

A.5 Cost Formulae

This section derives the entries of Table 3 and the cost statements of Section 5 under their stated convention. Biases, normalization, embeddings, output heads, nonlinearities, and data movement are excluded unless explicitly mentioned; a multiply followed by its accumulation counts as one MAC. These are excluded because they are small beside the weight matrices, vary with implementation (inference-time batch normalization with fixed statistics folds into the preceding weights, and biases are optional), or scale with the task rather than the architecture (embeddings and output heads grow with vocabulary and class count), as in the non-embedding counts of scaling studies.

A.5.1 Convolution, pooling, and structural span

For a grouped convolution at stage ℓ, each of the $H_\ell W_\ell c_{\mathrm{out},\ell}$ output activations uses $k_{h,\ell} k_{w,\ell} c_{\mathrm{in},\ell} / g_\ell$ weights and the same number of MACs, for a $k_{h,\ell} \times k_{w,\ell}$ kernel with g_ℓ groups. Hence

$$\mathrm{Par}_\ell = \frac{k_{h,\ell} k_{w,\ell} c_{\mathrm{in},\ell} c_{\mathrm{out},\ell}}{g_\ell},$$
$$\mathrm{MAC}_\ell = \frac{k_{h,\ell} k_{w,\ell} c_{\mathrm{in},\ell} c_{\mathrm{out},\ell} H_\ell W_\ell}{g_\ell}.$$

Summing these expressions over ℓ gives the convolutional costs of Section A.7.2.

On a one-dimensional underlying grid, adjacent sites at stage ℓ are separated in input coordinates by the accumulated stride $\prod_{j<\ell} s_j$. A kernel of width k_ℓ and dilation d_ℓ therefore increases dependency diameter by

$$d_\ell (k_\ell - 1) \prod_{j<\ell} s_j.$$

Adding these increments and capping by the declared domain diameter gives the upper bound equation 3; it is attained when the extreme dependencies are reachable. The same argument applies coordinatewise in higher dimensions. Arbitrary nonlocal supports instead use the union rule of Section A.1.8. With one nontrivial pattern-forming convolution or pooling operation per stage, the macro-stage order is $N_{\mathrm{conv}} + N_{\mathrm{pool}}$, the numbers of such convolution and pooling stages; physical depth also counts separately implemented activations and other layers.

For a same-resolution layer, an unrestricted dense map from HWc_{in} inputs to HWc_{out} outputs has $(HW)^2 c_{\mathrm{in}} c_{\mathrm{out}}$ parameters and MACs. It contains a shared local convolution as a constrained parameter choice. Implementing that choice as a convolution uses $k_h k_w c_{\mathrm{in}} c_{\mathrm{out}}$ parameters and $HWk_h k_w c_{\mathrm{in}} c_{\mathrm{out}}$ MACs instead. Thus support sparsity and sharing can reduce the cost of the same response field; these counts alone do not determine sample cost.

For a pooling window containing $q = k_h k_w$ sites, each output and channel visits q values: max pooling uses $q - 1$ comparisons, while mean pooling uses $q - 1$ additions and one scaling. Thus its graph work is linear in $H_{\mathrm{out}} W_{\mathrm{out}} cq$.

A.5.2 Dense attention and feed-forward blocks

Here q, k, v denote attention's query, key, and value; Q denotes an align, as throughout. For n tokens of width d, h heads of width d_h, and feed-forward width d_{ff}, assume $hd_h = d$ and that the concatenated value width is also d. The q, k, v, and output projections each have a $d \times d$ weight matrix, so multi-head attention contributes $4d^2$ parameters and $4nd^2$ projection MACs. Across all heads, forming $qk^\top$ costs $n^2 d$ MACs and multiplying the attention weights by v costs another $n^2 d$. Softmax and other elementwise work are omitted. A two-matrix FFN contributes $2dd_{\mathrm{ff}}$ parameters and $2ndd_{\mathrm{ff}}$ MACs. Consequently, for N_{blk} dense transformer blocks,

$$\mathrm{Par} = N_{\mathrm{blk}}(4d^2 + 2dd_{\mathrm{ff}}),$$
$$\mathrm{MAC} = N_{\mathrm{blk}}(2n^2 d + 4nd^2 + 2ndd_{\mathrm{ff}}).$$

For head r, with query and key projections $W_{q,r}, W_{k,r}$, the pre-softmax logit has bilinear coefficient

$$B_r = W_{q,r} W_{k,r}^\top / \sqrt{d_h}, \qquad \mathrm{rank}(B_r) \le \min(d, d_h).$$

The declared direct-bilinear capacity of the tuple of heads is therefore at most $\sum_{r=1}^{h} \min(d, d_h)$. This calculation concerns the available pre-softmax bilinear forms, not the matrix rank of a realized post-softmax attention matrix. Counting attention and its FFN separately gives $2N_{\mathrm{blk}}$ macro stages for this expression, not its minimum algebraic depth. Physical depth D_{geo} requires the chosen implementation to be expanded into the declared physical layers; dense attention already exposes the full input span.

A.5.3 MLPs and the remaining dictionary entries

For an MLP with widths $w_0, \ldots, w_L$, layer ℓ has $w_\ell w_{\ell+1}$ weights and the same number of per-example MACs. Summation gives

$$\mathrm{Par} = \mathrm{MAC} = \sum_{\ell=0}^{L-1} w_\ell w_{\ell+1}.$$

The first dense layer can couple every declared input site, so its dependency span is $\mathrm{diam}(U)$; the displayed macro-layer count is L. Physical depth is counted separately, including the implementation's activation layers. The standard affine–ReLU CNN and MLP families contain no *direct bilinear primitive*; this does not prevent their nonlinear programs from approximating multiplication. When such a primitive $a^\top B b$ is included, its architecture-only contribution is the maximum rank attainable by an admissible B.

The remaining elementary rows follow directly from their response maps. ReLU uses one sign comparison per activation and identifies all negative preactivations with zero. Concatenation $(a, b) \mapsto (a, b)$ is injective and requires no arithmetic beyond writing its output, whereas addition $(a, b) \mapsto a + b$ uses one addition per coordinate and has kernel $\{(t, -t) : t\}$. A residual merge has the same linear addition cost, but the boundary map $x \mapsto x + H(x)$ is only conditionally injective; a sufficient condition is given in Section A.1.4.

A.6 Detailed Operation Dictionary

Table 3: A QF and cost dictionary for common layers. Costs omit bias, normalization, embeddings, heads, and elementwise terms. The table describes available structure before training, not which folds actually fire after weights are chosen.

module	align / exposed pattern	possible fold	structural contribution	leading graph cost
convolutional stage	Q_{local}: translated patches; F_W produces shared responses	F_W: kernel directions collapsed on the reachable domain	Cross-map selection: $HWc_o \log_2(c_i/g)$; within-map positions add no source count	$\mathrm{Par} = k_h k_w c_i c_o/g$; MAC $= \mathrm{Par}\, HW$
ReLU	preceding affine map places a crease	binary selection from $\{0, x\}$	one bit per scalar output; no support-span increase	one comparison per activation
GELU / SiLU / sigmoid	preceding scalar responses	smoothed binary selection; retention ϵ measured separately	one bit per scalar output, as ReLU; no support-span increase	elementwise scalar nonlinearity
mean/max pool	aligns a spatial window	locations and within-window configurations	span increase; a declared max selector contributes $\log_2 m$ for m already formed response sources	linear in window visits
self-attention	pairwise controller and values on token supports	token configurations with equal weighted aggregates	span of combined input supports; pre-softmax rank $\leq \sum_r \min(d, d_h)$	$4nd^2 + 2n^2 d$ MACs
FFN / MLP layer	pointwise FFN or dense mixing of input coordinates	affine kernel and activation	FFN inherits span; dense input mixing may span the field; order $+1$	$2ndd_{\mathrm{ff}}$ for transformer FFN
normaliza-tion	shared centred coordinates	LN core: additive-shift fibers for $\epsilon > 0$; affine gains may add collisions. Softmax: additive-shift fibers	see domain conditions below	linear in vector size
concat. / add	shared output coordinates	concat. is injective; add removes $(t, -t)$	concat. alone selects or joins nothing; joint add across m already formed response sources contributes m per scalar	linear in output size
residual block	skip and branch share coordinates	conditional; injective under Section A.1.4	a two-endpoint add is a joint Fold and contributes two per scalar; a two-endpoint max is a selection Fold and contributes one per scalar	branch cost plus merge

Table 4 gives, for each primitive, its role, its contribution to L per output scalar, its Par and MAC under the cost convention, and its effect on span and D_{geo}. Decomposing a network into the class is a table lookup followed by the disambiguation rules below. In the table, n = sites or tokens at the output, c_i, c_o = input/output maps, g = groups, k = kernel width, $H'W'$ = retained output grid, s = accumulated stride, m = independent branches merged, d, d_{ff} = transformer widths, k_p, s_p = pool window side and stride, and n_{cls} = number of classes; "span" in the last column is the support span, the set of input sites a response depends on, not a linear span.

Table 4: Ledger lookup table.

primitive	role	L per output scalar	Par	MAC	span / D_{geo}
input site; gather; patch extraction; concat	align	0	0	0	span + support; D_{geo} +0 (routing)
conv $k \times k$, $c_i \to c_o$, groups g	selection	$\log_2(c_i/g)$	$k^2 c_i c_o/g$	Par $\cdot H'W'$	span $+(k-1)s$; D_{geo} +1
1×1 conv	selection	$\log_2(c_i/g)$	$c_i c_o/g$	Par $\cdot H'W'$	span +0; D_{geo} +1
depthwise conv	—	0 ($g = c_i$)	$k^2 c_o$	Par $\cdot H'W'$	span $+(k-1)s$; D_{geo} +1
dense $d_i \to d_o$	selection	$\log_2 d_i$	$d_i d_o$	$n d_i d_o$	span = field; D_{geo} +1
ReLU	selection	1, counted at its input field	0	0	D_{geo} +1
max / min over m branches	selection	$\log_2 m$	0	0	D_{geo} +1
ordinary spatial max pool, window $k_p \times k_p$, stride s_p	—	0 (within one map; no winner index)	0	0	span $+(k_p-1)s$; grid $/s_p^2$
average pool, window $k_p \times k_p$, stride s_p	—	0	0	0 (pool work recorded separately)	span $+(k_p-1)s$; grid $/s_p^2$
spatial skip / stride s'	fold	0	0	0; downstream MACs charged on retained grid only	grid $/s'^2$; $s \leftarrow ss'$
GELU / SiLU / sigmoid	selection	1, as ReLU	0	0	D_{geo} +1
pairwise scores $q_i \cdot k_j$	selection	$\log_2 d_h$ per score, $n^2 h$ scores ($n(n+1)h/2$ if causal)	0	$n^2 d$	span = token field; D_{geo} +1
softmax; value aggregation	—	0 (reweights within each map)	0	$n^2 d$	—
q, k, v, out projections	selection	$\log_2 d$ each	d^2 each	nd^2 each	D_{geo} +1 each
FFN $d \to d_{\text{ff}} \to d$	selection	$\log_2 d$, then $\log_2 d_{\text{ff}}$	$2dd_{\text{ff}}$	$2ndd_{\text{ff}}$	D_{geo} +2
residual add, m independent branches	joint	m	0	0	D_{geo} +1
node sum of k incoming edges	joint	k if $k \geq 2$; 0 if $k \leq 1$	0	0	D_{geo} +1
product merge $a \odot b$ (gating)	joint	2	0	nd	D_{geo} +1
Q_{ssm} (prefix scan)	align	0	per scan parameterization	per scan (App. audit)	span = prefix
BatchNorm / LayerNorm / RMSNorm	—	0	0 (excluded)	0	D_{geo} +0
cls token; positional embedding; rotary position	align	0	0 (excluded)	0	—
readout head $d \to n_{\text{cls}}$	readout	excluded	$d\, n_{\text{cls}}$	$d\, n_{\text{cls}}$ per readout site	—

Disambiguation rules. Applied after the lookup, in this order. (i) A source is an already formed response map or branch; spatial and kernel positions inside one map are never separate sources. (ii) A merged branch that is only a linear re-expression of the other source is not independent and does not raise m. (iii) A node fed by a single surviving edge selects nothing and joins nothing: 0 bits. (iv)

An operation with no surviving path from the input exposes nothing and contributes 0 to L, Par, and MAC. (v) An edge declared `none` is removed before counting. (vi) ReLU is counted at the field it acts on, one bit per scalar; a pre-activation ReLU and a post-merge ReLU are two events. (vii) Stem and readout are counted by the table; only the readout is excluded from L. (viii) A block applied r times with shared weights is counted once in L and in Par, since its applications are not independent; MAC and D_{geo} count each application. Blocks with separate weights are counted layer by layer. (ix) When a sequence of consecutive folds admits both a layer-by-layer description and a single-primitive description that contains all its feasible joint structures, use the smaller of the two formal counts. Both descriptions upper-bound the same set of feasible joint structures; the single-primitive description reads the first fold's sources and produces the last fold's outputs. Examples include linear folds separated only by inference-mode batch normalization with fixed statistics or other fixed affine maps, provided the composite is representable by a primitive in the table, and consecutive ReLUs. For a linear $d \to 1 \to m$ chain, the separate description gives $L = \log_2 d$, whereas the unrestricted composite description gives $L = m \log_2 d$; the former retains the constraint that all outputs share one intermediate source. (x) A function of parameters alone, or a monotone reparameterization of a coefficient (softplus on a step size), is not an operation on the response field and is not counted. (xi) A primitive not in the table is added by declaring its role and its L rule before any candidate is evaluated, never afterwards.

In the table, "excluded" marks a quantity that exists but is not counted: Par under the cost convention of Section A.5, and L for the readout, which acts on exposed patterns under the declared readout family of EP1 rather than exposing new ones.

Dictionary domain conditions. For LayerNorm with $\epsilon > 0$, the normalization core identifies exactly common additive shifts. The full affine LayerNorm has the same fibers if all gains are nonzero; zero gains can introduce further collisions (all-zero gains give a constant map). At $\epsilon = 0$, restrict to nonzero variance and include positive rescaling in the identified relation. For the standard bispectrum on a finite commutative group, all Fourier coefficients being nonzero is a sufficient condition for orbit completeness; no uniform completeness claim is made on the degenerate domain. The normalized variant $B/\|B\|$ additionally requires $B \neq 0$ and, under the same completeness conditions, identifies positive input rescalings as well as group translations (Sanborn et al., 2023, Appendices D–E). Other group domains and variants require their own completeness conditions.

A.7 Canonical constructions

We exhibit CNNs, transformers and MLPs as programs in the class by expressing their internal operations in the declared primitives and wiring, with exact deterministic input–output correspondence. These constructions need not be unique or shortest.

A.7.1 Transformer blocks

Take a standard pre-norm block (Dosovitskiy et al., 2021) on n tokens of width d with h heads of dimension $d_h = d/h$ and feed-forward width d_{ff}. Work at deterministic inference, with normalization, softmax and the chosen activation admitted in the dictionary. In the attention formulas below, X is the normalized token field; displayed projections omit biases, which are included by affine maps when used.

Patch embedding first extracts and flattens every patch: retaining their positions makes this a reversible align. The shared affine projection then folds any distinctions in its kernel on the reachable domain; it need not lose information. Position embeddings and class-token insertion use affine maps and concatenation.

Layer normalization removes common additive shifts. With $\epsilon > 0$, its normalization core has no positive-scale quotient; nonzero affine gains preserve these fibers, while degenerate gains may add collisions. For $\epsilon = 0$, positive-scale invariance holds on the nonzero-variance domain.

Score formation $S_r = X B_r X^\top$ with $B_r = W_{q,r} W_{k,r}^\top / \sqrt{d_h}$ of rank at most d_h exposes a pairwise relation among tokens. It supplies the controller for the fold that follows rather than lying on the path the values take.

Table 5: Leading costs of common blocks. Biases, normalization, heads and elementwise work are excluded. One MAC counts one multiply–accumulate. The full operation dictionary is in Table 3.

Module	Parameters	MACs
Grouped convolution	$k_h k_w c_i c_o / g$	$HW k_h k_w c_i c_o / g$
Dense self-attention	$4d^2$	$4nd^2 + 2n^2 d$
Transformer FFN	$2dd_{\text{ff}}$	$2ndd_{\text{ff}}$
Dense layer $u \to v$	uv	uv per vector

Softmax identifies logit vectors differing by a common shift, $\mathrm{softmax}(z + c\mathbf{1}) = \mathrm{softmax}(z)$, exactly, and is again a fold on one nuisance direction.

Value aggregation $Z_r = \mathrm{softmax}(S_r)XW_{V,r}$ folds configurations with equal outputs across all query positions. The align retains both scores and values until aggregation; head concatenation and the output projection complete the attention branch. Each token represents an input support; attention combines their supports, which may together span the whole image. Patch grouping is an intermediate realization of such a pattern, not a bound on its range.

Residual addition recombines the branch with the retained input. The *feed-forward sublayer* uses two pointwise affine maps with an activation between them. Both pre-norm branches use these same primitives and wiring, so the internal graph itself is a QF program, without requiring one QF stage per transformer block.

Exact realization. The composition realizes a pre-norm transformer block exactly, with the same parameter and MAC counts under the stated cost convention.

The factorized coefficients $B_r = W_{q,r}W_{k,r}^{\top}/\sqrt{d_h}$ and $C_r = W_{V,r}W_{O,r}$ retain the standard implementation, giving $4d^2 + 2dd_{\text{ff}}$ parameters and $4nd^2 + 2n^2 d + 2ndd_{\text{ff}}$ MACs per block. These are construction costs, not lower bounds over all equivalent programs. Derivations are in Section A.5.

A.7.2 Convolutional stacks

A convolutional stage is the worked example of Equation (1): patch extraction with declared placements is Q_{local}; the subsequent F_W performs kernel weighting and cross-channel linear aggregation, folding whatever kernel directions it collapses; the activation folds response magnitudes; and pooling folds site configurations. Local windows are chosen supports; their composition can span the input. For unit-spaced one-dimensional stacks, including any local readout in the count, connectivity gives the bound

$$\mathrm{span}(E) \le \min\Big\{\mathrm{diam}(U),\ \sum_{\ell} d_\ell(k_\ell - 1)\prod_{j<\ell} s_j\Big\}. \tag{3}$$

Costs again agree with the usual counting: a stage with kernel $k_h \times k_w$, c_{in} input and c_{out} output channels in g groups contributes $k_h k_w c_{\text{in}} c_{\text{out}}/g$ parameters and $H'W'$ times that many multiply–accumulates, where $H' \times W'$ is the output grid actually retained: a stride-s stage is charged only for the $1/s^2$ of positions its spatial-skip Fold keeps, so the ledger equals execution cost.

A.7.3 The remaining families

An unrestricted dense layer on flattened input exposes the full input span at once, paying for all its connections. A deep MLP can still compose responses; it does not prescribe local supports or spatial weight sharing. Residual wiring adds a bypass whose boundary map may stay injective even when its branch folds (Section A.1.4). Group-equivariant and invariant layers enter as symmetry folds, with complete invariants realizing the exact orbit quotient where an ordinary invariant would identify more. Table 5 summarizes leading costs; the full exposure, folding and cost dictionary is in Table 3.

A.8 Counting rules by primitive

Thus ReLU is binary selection from $\{0, x\}$ and contributes one bit per scalar site. Elementwise max of two branch responses is also binary selection and contributes one bit per scalar site; no separately

returned winner index is required or additionally counted. Elementwise addition of the same two branches is joint composition and contributes two bits per scalar site. More generally, selection from m already formed response maps contributes $\log_2 m$ per output position, whereas joint composition of m such maps contributes m. A convolutional or linear Fold adds $\log_2 c_{\text{in}}$ per scalar output when it combines c_{in} already formed response maps; within-map spatial mixing adds zero. Concatenation alone is wiring and adds zero. At a merge, a branch obtained solely by a linear re-expression of the other source, with no intervening nonlinearity or cross-response-map composition, is not counted as an independent source.

A.9 Experimental details

Controlled studies test joint coverage, collision diagnosis, repair, and training-budget predictions. Natural-image studies test the structural distinctions of L, empirical screening, and bounded construction. Frozen protocols and certificates are reported for the controlled and selection studies (Section A.14); the structural intervention study below is exploratory. An explicit witness certifies joint coverage. A weight-independent cross-label collision certifies impossibility. For unknown patterns, $C = 2^L$ remains the independent-choice formal count, and its calibrated threshold is tested against independently measured performance. Structural and learning costs are reported separately.

Cost-exact controlled separation. For the binary T_{far} task, we freeze two architectures with the same explicit four-pair gather, shared $2 \to 8$ affine encoder, ReLU, coordinatewise max, and $8 \to 1$ readout. Architecture A gathers $(0,4),(1,5),(2,6),(3,7)$, whereas B gathers $(0,1),(2,3),(4,5),(6,7)$. Thus the declared implementation has the same $D_{\text{geo}} = 4$, 24 non-bias parameters (33 trainable scalars including affine biases), 72 MACs, width, and aggregation rule on both sides; only the exposed supports differ. The fixed gather is declared zero-parameter routing and is not counted as a physical layer.

Proposition A.7 (Cost-exact T_{far} separation)**.** *Let $y(x)$ be the OR of the four far-pair ANDs. Under the frozen implementation above, A realizes y, while B cannot realize y for any shared encoder parameters and any deterministic downstream readout.*

Proof. The inputs $x = (0,0,0,0,0,1,0,1)$ and $x' = (0,0,0,1,0,0,0,1)$ have labels 0 and 1, but both induce the local-pair multiset $\{(0,0),(0,0),(0,1),(0,1)\}$. For every shared encoder $\phi_\theta : \{0,1\}^2 \to \mathbb{R}^8$, coordinatewise max is a function only of that multiset. Hence $\pi_{f_{B,\theta}} \not\preceq \pi_L$, so Proposition 6.1 and Theorem 6.2 rule out every deterministic downstream readout. For A, set one encoder coordinate to $\mathrm{ReLU}(a+b-1)$, set the other coordinates to zero, take the maximum over the four far pairs, and read out that coordinate. On binary inputs this equals y. Enumeration of all 2^8 inputs is an independent computational check of this analytic witness. □

This certificate evaluates a declared oracle-support repair; it does not claim that construction automatically discovers the task supports. Training is therefore an empirical confirmation, rather than the source of the impossibility claim.

Training confirmation. Both architectures, and an unrestricted Tiny-MLP reading all eight inputs, are trained on 128 of the 256 inputs for 4,000 Adam steps with three seeds and evaluated on the full domain. Far-Pair-QF reaches full-domain accuracy 1.000 in every seed; Local-QF reaches a mean of 0.745 (range 0.703–0.781) and never reaches full accuracy, as the certificate requires; Tiny-MLP reaches a mean of 0.987. The two models fall short of perfect accuracy for different reasons: Local-QF carries a structural collision that no weights can remove, whereas Tiny-MLP admits a task realization; its observed gap is a learning outcome rather than a structural impossibility certificate. An expanded matrix of five seeds, training sizes $\{64, 128, 192\}$, and five task–model pairs (75 trajectories), together with a sixteen-initialization audit of Pair-QF, gives the same separation at every scale. On the companion T_{local} task the pair-gather program coincides with the local-pair model, so their agreement there is an identity rather than a test and serves only as a negative control.

Pre-registered diagnosis and repair. Two further controlled families extend the same protocol; structural verdicts were committed before training. On an 81-input task, exhaustive enumeration confirms a mixed-label collision for the all-ReLU variant, while the Keep and Selective variants carry explicit witnesses that reach 100%. On a collision diagnosed in a Drop variant, restoring the discarded

coordinate x_0 recovers $98.44\% \pm 1.56\%$ over seeds, against about 88.9% and 89.1% for Drop and Wide-Drop. The repaired model removes the diagnosed obstruction; the remaining empirical gap is not explained by that collision certificate.

Blind budget prediction. To test a budget prediction motivated by EP2, a budget rule for training steps as a function of input size was calibrated on 4-, 6-, and 8-input instances, and predictions for held-out 10- and 12-input instances were frozen before those runs. For Pair-QF, all ten held-out runs reach full-domain accuracy within the predicted budget. For Tiny-MLP the extrapolated rule fails: three of five seeds at ten inputs and five of five at twelve. We report this failure as a result. It is not a counterexample to EP2, whose budget for the unrestricted family was not independently calibrated; it is a prediction error of the extrapolation rule, and it shows that a rule calibrated on the task-matched architecture does not transfer to one that must expose the same patterns at greater width. The MLP ($n \to 16 \to 16 \to 1$, ReLU) covers the task: one first-layer unit $\mathrm{ReLU}(x_i + x_{i+n/2} - 1)$ per far pair, their sum in the second layer, and the readout threshold $1/2$ realize the label on all 2^{10} and 2^{12} inputs, with margin $1/2$.

Exact-realization and cost-formula audit. We audit the translations used to place standard computation graphs in the align–fold class without training either graph. A separately frozen QF implementation is compared with the reference graph in double-precision evaluation over five nontrivial parameter draws and four continuous random input batches per draw. We use both a same-shape CIFAR ResNet-20 basic block and a downsampling $16 \to 32$ block with its 1×1 projection shortcut. The QF graph explicitly applies dense local patch alignment, kernel weighting and cross-channel linear folding, batch normalization, ReLU, the residual addition, and the final ReLU. Across all comparisons, the largest forward absolute error is 1.8×10^{-14} and the largest relative parameter-gradient error is 7.0×10^{-14}. A forced zero-preactivation tie test also agrees exactly at the ReLU boundary.

We make the same audit for a canonical pre-norm ViT block with $n = 65$ tokens, $d = 192$, three heads and $d_{\mathrm{ff}} = 768$. Explicit token normalization, $q/k/v$ projections, score formation, softmax, value aggregation, output projection, both residual additions, and the exact erf GELU feed-forward path are compared with the reference graph. All forward, input-gradient, and parameter-gradient errors are zero in the audit.

The corresponding shape formulas also agree with the implementations. For the same-shape and downsampling ResNet blocks, total trainable parameter counts (including affine BatchNorm parameters) are respectively 4,672 and 14,528; the convolution-only cost-convention counts are 4,608 and 14,336, with 294,912 and 917,504 MACs. For the ViT block, the total trainable count is 444,864 and the cost-convention count is $4d^2 + 2dd_{\mathrm{ff}} = 442{,}368$; its MAC count is $4nd^2 + 2n^2d + 2ndd_{\mathrm{ff}} = 30{,}376{,}320$. The MAC convention excludes normalization, activation, softmax, residual addition, and pooling. These audits validate the stated translations and accounting conventions; they do not claim that the architectures are functionally equivalent to one another.

Extending the dictionary to a state-space mixer. The mixer of the official 768-dimensional, 24-layer Mamba-130M is translated operation by operation: a local causal align, a continuous SiLU fold, a controller branch, one additional align Q_{ssm} under which position t receives responses on the prefix $\{0, \ldots, t\}$, a per-position product merge, and a linear fold. No other primitive is needed. The translation is checked against the official `Mamba.forward` path in double precision over five parameter draws and four input batches, using the repository's CPU-compatible bridge and a source-faithful scan where CUDA kernels are unavailable: the largest forward absolute error is 3.64×10^{-11} and the largest relative parameter-gradient error 1.40×10^{-10}, with parameter and MAC counts matching exactly. At sequence length $T = 2048$ one mixer carries $3{,}770{,}880$ parameters and 7.817G reference MACs. This places an SSM mixer in the class and in the same ledger as convolution and attention; it makes no claim about its task performance.

SiLU counts one bit per scalar in L, like ReLU; its ϵ is measured directly on the trained checkpoint over eight natural-language inputs, giving $4{,}718{,}592$ scalar observations per gate position. The post-convolution SiLU has median retention ratio $\epsilon = 0.4924$ with a concentrated distribution; the post-scan gate has $\epsilon = 0.4569$ with a longer low-retention tail. Two positions of the same primitive in the same network differ, which is why ϵ is assigned per position rather than globally. These values

Table 6: ResNet-20 structural variants on CIFAR-100, mean of three seeds, 100 epochs. Parameters are total trainable; L on the retained grid.

Variant	Par	L	Acc. (%)
Standard ResNet-20	278,324	1,213,808	69.06
Moved-ReLU	278,324	1,197,424	68.88
Early-thin, add, keep S2	268,412	1,147,576	68.79
Early-thin, max, keep S2	268,412	1,061,560	67.36
Early-thin, add, delete S2	231,292	918,200	66.94
Early-thin, max, delete S2	231,292	848,568	66.66
Half-width, max	144,356	808,304	64.05

enter L_φ for the mixer through the retention-distribution artifact; the measurement does not change L.

Reference capacity audit and calibration. Before training, we audit the declared selection/joint Fold grammar on reference architectures. For CIFAR-ResNet18 with a 3×3, stride-one stem and no input max-pool, the first completed full-span stage is the first block at $8 \times 8 \times 256$. Counting convolutional selection across already formed input response maps, scalar ReLU selection, and two-source residual joint Folds gives

$$L_{\text{first-full}} = 3{,}932{,}160 + 65{,}536 \log_2 3 \simeq 4{,}036{,}032.102.$$

The complete network has

$$L_{\text{final}} = 4{,}677{,}632 + 65{,}536 \log_2 3 \simeq 4{,}781{,}504.102.$$

Kernel positions inside one response map do not multiply either value.

Standard Patch4-ViT has $n = 65$ tokens including the class token, embedding width $d = 192$, $h = 3$ heads of width $d_h = 64$, and FFN width $d_{\text{ff}} = 768$. Each block contributes the four projections $4nd \log_2 d$, the pairwise scores $n^2 h \log_2 d_h$, the two FFN cross-response-map terms $nd_{\text{ff}} \log_2 d + nd \log_2 d_{\text{ff}}$, the GELU sites nd_{ff} counted like ReLU, and two residual joint Folds $4nd$. Hence

$$L_{\text{block}} = nd\big(4 \log_2 d + \log_2 d_{\text{ff}} + 4\big) + nd_{\text{ff}}(\log_2 d + 1) + n^2 h \log_2 d_h \simeq 1{,}052{,}792.988,$$

which is already full-span in block 1, and $L_{\text{final}} \simeq 12{,}633{,}515.857$ after 12 blocks. These reference values calibrate capacity thresholds; a threshold is relative to a declared reference model and is not a cross-architecture score of total functional complexity. For the ResNet-20 block family below, the threshold is read from the early-thin, add, delete-S2 variant of Table 6, with terminal response width $k = 64$.

Structural interventions on CIFAR-100. Seven predeclared variants of ResNet-20 are trained on CIFAR-100 for 100 epochs with three seeds (Table 6). Deleting or narrowing blocks reduces accuracy in every case. At equal parameters and MACs, the additive merge carries one more bit per merge site than the max merge (86,016 with S2 kept, 69,632 with S2 deleted) and scores 1.43 and 0.28 points higher. Moved-ReLU, whose ledger differs from the standard model's by one ReLU field (16,384 bits, where its shortcut ReLU follows the stem's), lands within noise of it.

Bounded selection within a frozen family. Two frozen enumerations exercise Proposition 6.3 with no candidate training. A network-level enumeration of ResNet-20 stage widths and block counts (72 candidates) returns two front points on $(\text{Par}, \text{MAC}, D_{\text{geo}})$: width and depth move parameters and MACs together. The block-level enumeration varies the layer multiset within ± 1 of the standard block, 3×3 versus 1×1 kernels, and layer order, with stage widths fixed. Of 140 legal schemas, 33 pass $L \geq 918{,}200$ with parameter and MAC bounds and terminal width $k = 64$; 10 are cost-nondominated; they occupy four numeric ledgers. Six representatives spanning both kernel orders and zero to two ReLUs before the merge, the standard block, and eight further schemas were trained with three seeds (Table 7). The seven admitted schemas are the seven most accurate, and the seven excluded are exactly those with no nonlinearity between their two convolutions. The standard block's width narrowed to the same parameters gives 64.3% and 65.5% at widths 11 and 12, both with $L < L_{\text{target}}$.

Table 7: The six ledger representatives on CIFAR-100, three seeds. $\mathrm{MAC}_{\mathrm{exec}}$ charges retained positions only; L on the retained grid with consecutive linear folds composed. Threshold $918{,}200$.

kernel order	ReLU between convs	Par	$\mathrm{MAC}_{\mathrm{exec}}$	L	accuracy (%)
$3\times3\to1\times1$	no	147,632	21,944,576	587,120	62.53 ± 0.30
$3\times3\to1\times1$	yes	147,632	21,944,576	1,213,808	66.43 ± 0.19
$3\times3\to1\times1$	yes, and before merge	147,632	21,944,576	1,299,824	65.03 ± 0.09
$1\times1\to3\times3$	no	168,112	24,041,728	587,120	62.65 ± 0.38
$1\times1\to3\times3$	yes	168,112	24,041,728	1,213,808	65.80 ± 0.37
$1\times1\to3\times3$	yes, and before merge	168,112	24,041,728	1,299,824	65.21 ± 0.30

All admissible blocks. The remaining 25 admissible schemas were trained with the same recipe and seeds, so every one of the 33 has three seeds on CIFAR-100 (Table 8). All clear the threshold, and their L takes only two values, $1{,}213{,}808$ and $1{,}299{,}824$; accuracy separates by depth and parameters instead. Blocks three convolutions deep average 1.0–1.5 points below two-convolution blocks with fewer parameters, although the group ranges overlap, and the four lowest (62.7–63.2%) reach the accuracy range of the excluded blocks: above the threshold, consistent with depth as a cost (EP2). Within a group, placement of the ReLU and batch normalization moves accuracy by up to 2.4 points; six pairs differing only in batch-normalization position, identical at inference in function class, Par, MAC, D_{geo}, and L, differ by 0.01–0.82 points. The best admissible block other than the standard one, a double 3×3 at the standard block's cost, scores 68.40% against the standard block's 68.87%. Among the 28 admissible blocks cheaper than the standard one, the ledger's block (147,632 parameters, one ReLU) has the highest mean, 66.43%.

Table 8: All 33 admissible block schemas on CIFAR-100, three seeds each, grouped by depth and parameters.

D_{geo}	Par	structure	schemas	range (%)	mean (%)
55	276,656	two 3×3	5	67.59–68.87	68.17
55	168,112	1×1 and 3×3	5	64.90–65.80	65.34
55	147,632	3×3 and 1×1	5	64.08–66.43	65.21
73	163,760	3×3 and two 1×1	6	63.89–64.74	64.17
73	184,240	two 1×1 and a 3×3	12	62.72–65.15	63.83

Small-space construction audit. On the pair-gather family, all 32 declared candidates are enumerated with the budgeted procedure (49 prefixes expanded, 112 generated, 32 completed), returning a four-point Pareto front with no candidate training and no incorrect pruning.

A.10 Post-training pruning on the same model

Table 9: ResNet-20 on CIFAR-100: post-training filter pruning (Table 5 of Zhang et al., 2023; each method's own baseline 68.47, 278.3k parameters, 41.20M FLOPs, 300 epochs of fine-tuning) against the ledger's pre-training pick (our baseline 68.9 ± 0.4, 276,656 parameters, 40.8M MACs, 100 epochs from scratch, 3 seeds).

method	parameters ↓	FLOPs ↓	Δ top-1
L1-norm	36.7%	49.5%	−1.88
FPGM-0.4	34.0%	50.0%	−1.79
LFP	36.7%	50.0%	−1.04
IENP	39.4%	51.5%	−2.71
LFP	43.4%	52.3%	−2.65
ledger pick, $3\times3\to1\times1$, one ReLU (before training)	46.6%	46.2%	−2.44

A.11 Related work in detail

Unified accounts of architectures. Geometric deep learning (Bronstein et al., 2021) organizes architectures by the symmetries they respect; categorical deep learning (Gavranović et al., 2024) gives an algebraic language, monads valued in parametric maps, that recovers both such constraints and implementations. MetaFormer (Yu et al., 2022) abstracts transformers and MLP-like models into one template with an interchangeable token mixer, and structured state space duality (Dao & Gu, 2024) shows that a class of state-space models and a class of masked attention compute the same maps. The align–fold class shares their aim of one description for many families. It differs in what the description is for: it carries a ledger of pattern constructions and cost, computed before training and used to diagnose coverage and to select within a declared family.

Training-free selection. Zero-cost proxies score initialized networks: NASWOT by activation patterns (Mellor et al., 2021), SynFlow by synaptic saliency (Tanaka et al., 2020), Zen by an expressivity score (Lin et al., 2021), and TE-NAS by the conditioning of the neural tangent kernel together with the number of linear regions (Chen et al., 2021b). NASGraph (Huang et al., 2024) converts an architecture to a graph and ranks it by graph measures such as average degree, without data and on a CPU. These methods rank candidates. Our framework places the candidates in one class, certifies structural obstructions, and reads its ledger directly from the architecture, without instantiating or training a network, using it as an admissibility threshold together with a cost-Pareto front rather than as a score. The closest quantities are linear-region counts (Montúfar et al., 2014) and the regions measured by TE-NAS; they count realized regions of a particular network, whereas the ledger counts formal basis transformations of a structure.

A.12 Additional results

EP1 on a complete domain. Grammars (declared before tasks): single pixels; adjacent-pair AND, OR, XOR; all-pair AND, OR, XOR; 2×2 window counts, AND, OR; row and column counts, AND, OR; the whole image as one pattern. Tasks: a full row; equal opposite corners; majority; parity; left–right symmetry; XOR of two corners; and, held out until the grammars were fixed, all four corners equal. Collisions are found by exact integer grouping. Feasible cells carry an LP solution scaled to integers and checked on all 512 inputs; infeasible cells carry an integer Farkas witness. Adjacent pairs lose exactly one pair of $130{,}816$, the two complementary checkerboards. Parity under all pairs has no witness of size ≤ 3; weighting each Hamming level by $(84, 28, 7, 1)$, the third finite difference, gives two multisets of 336 images with equal sums of every pairwise feature, since each such sum is quadratic in the number of ones and parity is not.

Table 10: Published results without position embedding, read through the fiber criterion. The criterion concerns what the architecture exposes; the reported numbers are the authors'.

setting	what exposes order	result
ViT-B/16, ImageNet 5-shot (Dosovitskiy et al., 2021)	position embedding	61.4 without vs 64.2 with
causal LM (Haviv et al., 2022)	causal mask: position t sees $\{1, \ldots, t\}$	perplexity 13.10 without vs 13.05 with
bidirectional MLM (Haviv et al., 2022)	nothing	perplexity 147.2 without vs 4.1 with
jigsaw, this paper	nothing (ViT, no PE)	chance, certified before training

Without position embedding. Classification keeps much of its accuracy without position embedding because the retained responses still carry most of what the labels need; this is a reading of the published result, not a fiber certificate. The causal mask as an order-exposing mechanism is analysed in Kazemnejad et al. (2023).

ViT-Lite. Table 11 gives the means behind Figure 2b.

Table 11: ViT-Lite on CIFAR-10, three seeds; L counts GELU like ReLU.

configuration	L	Par	accuracy (%)
ViT-Lite-7/4, reference	8.07 M	3.68 M	93.30 ± 0.09
ViT-Lite-6/4, threshold and cost minimum	6.92 M	3.16 M	92.80 ± 0.27
depth 7, ratio 1 (0.998 of threshold)	6.91 M	**2.77 M**	$\mathbf{92.80 \pm 0.11}$
below threshold (depth 7, width 192)	6.60 M	2.59 M	92.54
below threshold (depth 5, ratio 3)	6.57 M	3.29 M	92.40

Language model: depth as a cost. Decoder-only models on TinyStories (Eldan & Li, 2023), a 21 M reference. The family contains 891 candidates under the same ledger; the reported configurations were trained with two seeds under the same token budget. Under rule (viii) shared weights count once,

configuration	Par / ref	D_{geo} / ref	val. loss − ref (nats)
16 layers, width 416	0.78	1.33	+0.05
24 layers, width 256	0.59	2	+0.19
48 layers, width 160	0.35	4	+0.41
64 layers, width 128	0.30	5.3	+0.50

so sharing adds compute and depth but no capacity; at matched MACs and depth a shared-weight model trails its unshared twin by 0.12 nats. The prediction registered for the 24-layer point (+0.05) did not hold. Capped at the reference's depth, the cheapest admissible configuration is the reference itself.

Tasks. The cross-task study uses MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017), SVHN (Netzer et al., 2011), and CIFAR-10, CIFAR-100, and its twenty superclasses (Krizhevsky & Hinton, 2009), all at 32×32 with the CIFAR-100 recipe; horizontal flips are off for the two digit tasks. On CIFAR-10 the three $3 \times 3 \to 1 \times 1$ blocks keep their CIFAR-100 order: one ReLU 91.21, two ReLUs 90.47, no ReLU (excluded) 88.58, against 92.35 for the standard block.

MobileLLM depth–width grids. Liu et al. (2024b) train nine 125 M and ten 350 M configurations of equal size that trade depth for width. With the threshold read from each group's baseline, the ledger's gate agrees with their training in the 125 M group: every excluded configuration scores below every admitted one. In the 350 M group one excluded configuration scores above the baseline; the baseline is not that group's smallest success, and a threshold read from it is a looser bound.

AutoFormer search spaces. Accuracies are ImageNet top-1 with weights inherited from the released supernets, as AutoFormer evaluates its own subnets. Tiny space (ViT-Bench-101, 200 subnets): the ledger's front is one subnet, width 192, depth 12, Par 0.98, MAC 1.00, D_{geo} 0.92 of AutoFormer-T's, 75.35% against 75.3% listed for the released AutoFormer-T checkpoint in the authors' repository. Base space (OoD-ViT-NAS, 1,001 subnets): all lie within 0.56 points; mean 82.08 below L(AutoFormer-B) (163 subnets), 82.07 between L(AutoFormer-B) and L(DeiT-B) (799), 82.05 above (39). Front members: width 528, depth 15, 0.90 of AutoFormer-B's Par, 81.81%; and one at 0.985 of its Par, 82.24%. The cheapest buys its L with depth; in this space accuracy follows width more than depth, the same trade as the language model at a smaller price.

A.13 External static accounting

Published architecture pairs are re-expressed in the same ledger as a consistency check of the accounting. These are static re-computations of others' reported results; they test whether the ledger is well defined across families and carry no causal claim. For Swin-T (Liu et al., 2021), shifted and unshifted windows have identical parameters, MACs, elementwise work, peak field, and D_{geo} under the logical ledger, while the original paper reports gains of ImageNet 80.2 → 81.3, COCO box 47.7 → 50.5 and mask 41.5 → 43.7, and ADE20K 43.3 → 46.1; the cyclic-shift latency reported there lies outside the logical ledger by construction. Vision Mamba (Zhu et al., 2024), MambaOut

(Yu & Wang, 2025), DeiT (Touvron et al., 2021), VMamba (Liu et al., 2024a), and ConvNeXt (Liu et al., 2022) pairs from the literature show that equal reported accuracy can be reached with visibly different support mechanisms and resource ledgers, and in particular that a global SSM support does not by itself yield higher accuracy. This is consistent with the position that capacity is an admissibility condition rather than a quality score.

A.14 Registered artifacts

The reported controlled and construction studies identify the following registry or certificate categories: the T_{far} iso-cost certificate and training records; the expanded 75-run matrix and initialization audit; the 81-input and Drop/Wide-Drop registries and results; the blind-prediction file and held-out results; the ResNet block and ViT block exactness certificates; the Mamba-130M official-path cross-check and retention-distribution audit; the reference capacity audit; the cross-layer ledger audit for the CIFAR-100 variants; the 72-candidate and 140-schema enumeration ledgers with their frozen protocols and training selections; the pair-gather construction audit; the ViT-Lite protocol; the enumeration of EP1 with its certificates; the jigsaw certificate and declared-pattern readout; the language-model selections; and the AutoFormer ledger audits. The block-schema registry records MACs as frozen, on the input grid; the execution ledger of Section A.5 is used elsewhere in this paper.